

Alibaba Cloud

Apsara Stack

Enterprise

User Guide - Cloud Essentials
and Security

Product Version: 2006, Internal: V3.12.0

Document Version: 20201020

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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.ASCM console	88
1.1. What is the ASCM console?	88
1.2. User roles and permissions	88
1.3. Log on to the ASCM console	89
1.4. Web page introduction	90
1.5. Initial configuration	91
1.5.1. Configuration description	91
1.5.2. Configuration process	92
1.6. Monitoring	93
1.6.1. View the workbench	93
1.6.2. CloudMonitor	94
1.6.2.1. Cloud Monitor overview	94
1.6.2.2. Metrics	94
1.6.2.3. View monitoring charts	107
1.6.3. Alerts	107
1.6.3.1. View alarm overview	107
1.6.3.2. Enable or disable alert notification	107
1.6.3.3. View alert logs	107
1.6.3.4. Alarm rules	108
1.6.3.4.1. Query alert rules	108
1.6.3.4.2. Create an alarm rule	108
1.6.3.4.3. Disable an alarm rule	110
1.6.3.4.4. Enable an alarm rule	110
1.6.3.4.5. Delete an alarm rule	110
1.7. Enterprise	111
1.7.1. Organizations	111

1.7.1.1. Create an organization	111
1.7.1.2. Query an organization	111
1.7.1.3. View organization information	111
1.7.1.4. Modify the name of an organization	111
1.7.1.5. Change organization ownership	112
1.7.1.6. Obtain the AccessKey pair of an organization	112
1.7.1.7. Delete an organization	113
1.7.2. Resource sets	113
1.7.2.1. Create a resource set	113
1.7.2.2. View the details of a resource set	113
1.7.2.3. Modify the name of a resource set	114
1.7.2.4. Add a member to a resource set	114
1.7.2.5. Add or remove a user group of a resource set	114
1.7.2.6. Delete a resource set	115
1.7.3. Roles	115
1.7.3.1. Create a custom role	115
1.7.3.2. View the details of a role	116
1.7.3.3. Modify custom role information	117
1.7.3.4. Copy a role	117
1.7.3.5. Disable a role	118
1.7.3.6. Enable a role	118
1.7.3.7. Delete a custom role	118
1.7.4. Users	119
1.7.4.1. System users	119
1.7.4.1.1. Create a user	119
1.7.4.1.2. Query a user	120
1.7.4.1.3. Modify user information	120
1.7.4.1.4. Change user roles	121

1.7.4.1.5. Modify the information of a user group	121
1.7.4.1.6. Modify a user logon policy	122
1.7.4.1.7. View the initial password of a user	122
1.7.4.1.8. Reset the password of a user	122
1.7.4.1.9. Disable and enable a user	123
1.7.4.1.10. Delete a user	123
1.7.4.2. Historical users	124
1.7.4.2.1. Query historical users	124
1.7.4.2.2. Restore historical users	124
1.7.5. Logon policies	124
1.7.5.1. Create a logon policy	124
1.7.5.2. Query a logon policy	126
1.7.5.3. Modify a logon policy	126
1.7.5.4. Disable a logon policy	126
1.7.5.5. Enable a logon policy	126
1.7.5.6. Delete a logon policy	127
1.7.6. User groups	127
1.7.6.1. Create a user group	127
1.7.6.2. Add users to a user group	128
1.7.6.3. Delete users from a user group	128
1.7.6.4. Add a role	129
1.7.6.5. Delete a role	129
1.7.6.6. Modify the name of a user group	130
1.7.6.7. Delete a user group	130
1.7.7. Resource pools	130
1.7.7.1. Update associations	130
1.7.8. Change ownership	130
1.8. Configurations	131

1.8.1. Password policies	131
1.8.2. Menus	131
1.8.2.1. Create a menu	131
1.8.2.2. Modify a menu	133
1.8.2.3. Delete a menu	133
1.8.2.4. Display or hide menus	134
1.8.3. Specifications	134
1.8.3.1. Specification parameters	134
1.8.3.2. Create specifications	138
1.8.3.3. View specifications	138
1.8.3.4. Disable specifications	138
1.8.3.5. Export specifications	139
1.8.3.6. View specifications of each resource type in previ...	139
1.8.4. Message center	139
1.8.4.1. View internal messages	139
1.8.4.2. Mark messages as read	140
1.8.5. Resource pool management	140
1.9. Operations	141
1.9.1. Quotas	141
1.9.1.1. Quota parameters	141
1.9.1.2. Set quotas for a cloud service	143
1.9.1.3. Modify quotas	144
1.9.1.4. Reset quotas	144
1.9.2. Usage statistics	145
1.9.2.1. View the usage statistics of cloud resources	145
1.10. Security	145
1.10.1. View operation logs	145
1.11. RAM	146

1.11.1. RAM introduction	146
1.11.2. Permission policy structure and syntax	146
1.11.3. RAM roles	149
1.11.3.1. View basic information about a RAM role	149
1.11.3.2. Create a RAM role	149
1.11.3.3. Add a permission policy	149
1.11.3.4. Modify the content of a RAM permission policy	150
1.11.3.5. Modify the name of a RAM permission policy	150
1.11.3.6. Add a RAM role to a user group	151
1.11.3.7. Grant permissions to a RAM role	151
1.11.3.8. Remove permissions from a RAM role	151
1.11.3.9. Modify a RAM role name	152
1.11.3.10. Delete a RAM role	152
1.11.4. RAM authorization policies	152
1.11.4.1. Create a RAM role	152
1.11.4.2. View the details of a RAM role	153
1.11.4.3. View RAM authorization policies	153
1.12. Personal information management	153
1.12.1. Modify personal information	153
1.12.2. Change your logon password	154
1.12.3. Switch the current role	154
1.12.4. View the AccessKey pair of your Apsara Stack tena...	155
2.Elastic Compute Service (ECS)	156
2.1. What is ECS?	156
2.1.1. Overview	156
2.1.2. Instance types	156
2.1.3. Instance lifecycle	172
2.2. Instructions	173

2.2.1. Restrictions	173
2.2.2. Suggestions	174
2.2.3. Limits	174
2.2.4. Notice for Windows users	174
2.2.5. Notice for Linux users	175
2.2.6. Notice on defense against DDoS attacks	175
2.3. Quick start	175
2.3.1. Overview	175
2.3.2. Log on to the ECS console	176
2.3.3. Create a security group	176
2.3.4. Create an instance	177
2.3.5. Connect to an instance	182
2.3.5.1. Instance connecting overview	182
2.3.5.2. Connect to a Linux instance by using SSH comm...	182
2.3.5.3. Connect to a Linux-based instance by using rem...	182
2.3.5.4. Connect to a Windows instance by using RDP	183
2.3.5.5. Connect to an ECS instance by using the VNC	184
2.4. Instances	185
2.4.1. Create an instance	185
2.4.2. Connect to an instance	189
2.4.2.1. Instance connecting overview	189
2.4.2.2. Connect to a Linux-based instance by using SSH ...	189
2.4.2.3. Connect to a Linux-based instance by using rem...	189
2.4.2.4. Connect to a Windows instance by using RDC	190
2.4.2.5. Install the certificate for VNC in Windows	191
2.4.2.6. Connect to an ECS instance by using the VNC	192
2.4.3. View instances	193
2.4.4. Modify an instance	194

2.4.5. Stop an instance	194
2.4.6. Start an instance	195
2.4.7. Restart an instance	195
2.4.8. Delete an instance	196
2.4.9. Change the instance type of an instance	196
2.4.10. Change an instance logon password	196
2.4.11. Change the VNC password	197
2.4.12. Add an ECS instance to a security group	197
2.4.13. Customize instance data	197
2.4.14. Modify a private IP address	200
2.4.15. Install the CUDA and GPU drivers for a Linux insta...	200
2.4.16. Install the CUDA and GPU drivers for a Windows in...	203
2.5. Disks	204
2.5.1. Create a disk	204
2.5.2. View disks	206
2.5.3. Roll back a disk	207
2.5.4. Modify the disk properties	208
2.5.5. Modify the disk description	208
2.5.6. Attach a disk	209
2.5.7. Partition and format disks	209
2.5.7.1. Format a data disk for a Linux instance	209
2.5.7.2. Format a data disk of a Windows instance	214
2.5.8. Expand a disk	215
2.5.9. Reinitialize a disk	216
2.5.10. Detach a data disk	216
2.5.11. Release a data disk	217
2.6. Images	217
2.6.1. Create a custom image	217

2.6.2. View images	218
2.6.3. View instances related to an image	218
2.6.4. Modify the description of a custom image	219
2.6.5. Share custom images	219
2.6.6. Import custom images	219
2.6.6.1. Limits on importing custom images	219
2.6.6.2. Convert the image file format	223
2.6.6.3. Import a custom image	224
2.6.7. Export a custom image	225
2.6.8. Delete a custom image	226
2.7. Snapshots	226
2.7.1. Create a snapshot	226
2.7.2. View snapshots	227
2.7.3. Delete a snapshot	228
2.8. Automatic snapshot policies	228
2.8.1. Create an automatic snapshot policy	228
2.8.2. View automatic snapshot policies	229
2.8.3. Modify an automatic snapshot policy	230
2.8.4. Configure an automatic snapshot policy	230
2.8.5. Configure an automatic snapshot policy for multiple...	230
2.8.6. Delete an automatic snapshot policy	231
2.9. Security groups	231
2.9.1. Create a security group	231
2.9.2. View security groups	233
2.9.3. Modify a security group	233
2.9.4. Add a security group rule	233
2.9.5. Clone a security group rule	235
2.9.6. Modify a security group rule	235

2.9.7. Export security group rules	236
2.9.8. Import security group rules	236
2.9.9. Add an instance	236
2.9.10. Remove instances from a security group	237
2.9.11. Delete a security group	237
2.10. Elastic Network Interfaces	237
2.10.1. Create an ENI	237
2.10.2. View ENIs	240
2.10.3. Modify a secondary ENI	240
2.10.4. Bind a secondary ENI to an instance	241
2.10.5. Unbind a secondary ENI from an instance	241
2.10.6. Delete a secondary ENI	242
2.11. Deployment sets	242
2.11.1. Create a deployment set	242
2.11.2. View deployment sets	243
2.11.3. Modify a deployment set	244
2.11.4. Delete a deployment set	244
2.12. Install FTP software	244
2.12.1. Overview	244
2.12.2. Install and configure vsftpd in CentOS	244
2.12.3. Install vsftpd in Ubuntu or Debian	245
2.12.4. Build an FTP site in Windows Server 2008	246
2.12.5. Build an FTP site in Windows Server 2012	247
3. Container Service for Kubernetes	248
3.1. What is Container Service?	248
3.2. Planning and preparation	248
3.3. Quick start	248
3.3.1. Flowchart	248

3.3.2. Log on to the Container Service for Kubernetes con...	249
3.3.3. Log on to the Container Registry console	249
3.3.4. Create a Kubernetes cluster	250
3.3.5. Create an application from an orchestration templat...	253
3.4. Kubernetes clusters	255
3.4.1. Authorizations	255
3.4.1.1. Assign RBAC permissions to a RAM user	255
3.4.2. Clusters	257
3.4.2.1. Create a Kubernetes cluster	257
3.4.2.2. View cluster logs	260
3.4.2.3. Connect to a cluster through kubectl	260
3.4.2.4. Connect to a master node by using SSH	261
3.4.2.5. Expand a cluster	262
3.4.2.6. Renew a certificate	262
3.4.2.7. Delete a cluster	263
3.4.2.8. View cluster overview	263
3.4.3. Nodes	264
3.4.3.1. Add an existing node	264
3.4.3.2. View nodes	266
3.4.3.3. Manage node labels	267
3.4.3.4. Set node schedulability	269
3.4.3.5. Remove a node	269
3.4.3.6. View node resource usage	271
3.4.3.7. Upgrade the NVIDIA driver on a GPU node	272
3.4.3.8. Create a Kubernetes cluster for GPU computing	275
3.4.3.9. Use labels to schedule pods to GPU nodes	280
3.4.3.10. Manually upgrade the kernel of a GPU node in...	284
3.4.4. Storage	287

3.4.4.1. Overview	287
3.4.4.2. Use Apsara Stack disks	288
3.4.4.3. Use NAS volumes	295
3.4.4.4. Use OSS volumes	303
3.4.4.5. Create a PVC	307
3.4.4.6. Use a PVC	309
3.4.5. Network management	310
3.4.5.1. Set access control for pods	310
3.4.5.2. Set bandwidth limits for pods	312
3.4.6. Namespaces	313
3.4.6.1. Create a namespace	313
3.4.6.2. Set resource quotas and limits	315
3.4.6.3. Edit a namespace	317
3.4.6.4. Delete a namespace	318
3.4.7. Applications	319
3.4.7.1. Create an application from an image	319
3.4.7.2. Create an application from an orchestration tem...	327
3.4.7.3. Create an application through Kubernetes Dashb...	330
3.4.7.4. Use commands to manage applications	332
3.4.7.5. Create a service	333
3.4.7.6. Scale a service	336
3.4.7.7. View a service	337
3.4.7.8. Update a service	338
3.4.7.9. Delete a service	340
3.4.7.10. Create a trigger on an application	340
3.4.7.11. View pods	342
3.4.7.12. Schedule pods to nodes	343
3.4.7.13. Simplify Kubernetes application deployment by u...	345

3.4.8. SLB and Ingress	350
3.4.8.1. Overview	350
3.4.8.2. Access a service through SLB	350
3.4.8.3. Configure ingress monitoring	354
3.4.8.4. Ingress support	357
3.4.8.5. Ingress configurations	363
3.4.8.6. Create an Ingress in the console	364
3.4.8.7. Update an ingress	371
3.4.8.8. Delete an ingress	371
3.4.9. Config maps and secrets	372
3.4.9.1. Create a ConfigMap	372
3.4.9.2. Use a ConfigMap in a Pod	373
3.4.9.3. Update a ConfigMap	378
3.4.9.4. Delete a ConfigMap	379
3.4.9.5. Create a secret	379
3.4.9.6. Edit a secret	380
3.4.9.7. Delete a secret	381
3.4.10. Templates	382
3.4.10.1. Create an orchestration template	382
3.4.10.2. Update an orchestration template	383
3.4.10.3. Save an orchestration template as a new one	384
3.4.10.4. Download an orchestration template	384
3.4.10.5. Delete an orchestration template	385
3.4.11. Images	385
3.4.11.1. Create an image repository	385
3.4.11.2. Create a namespace	387
3.4.11.3. Synchronize an image	388
3.4.11.4. Sign and verify an image	389

3.4.11.5. Synchronize images between instances	394
3.4.12. Create a batch release	394
3.4.13. Use Log Service to collect Kubernetes logs	397
4.Auto Scaling (ESS)	409
4.1. What is ESS?	409
4.2. Notes	410
4.2.1. Precautions	410
4.2.2. Manual intervention	411
4.2.3. Limits	412
4.2.4. Scaling group status	412
4.2.5. Scaling activity process	413
4.2.6. Remove unhealthy ECS instances	414
4.2.7. Instance rollback after a failed scaling activity	414
4.2.8. Instance lifecycle management	414
4.3. Quick start	415
4.3.1. Overview	415
4.3.2. Log on to the Auto Scaling console	415
4.3.3. Create a scaling group	416
4.3.4. Create a scaling configuration	418
4.3.5. Enable a scaling group	420
4.3.6. Create a scaling rule	421
4.3.7. Create a scheduled task	421
4.3.8. Create an event-triggered task	422
4.4. Scaling groups	424
4.4.1. Create a scaling group	424
4.4.2. Enable a scaling group	426
4.4.3. View scaling groups	426
4.4.4. Modify a scaling group	427

4.4.5. Disable a scaling group	427
4.4.6. Delete a scaling group	428
4.4.7. View ECS instances	428
4.4.8. Put an ECS instance into the Standby state	428
4.4.9. Remove an ECS instance from the Standby state	429
4.4.10. Put an ECS instance into the Protected state	429
4.4.11. Remove an ECS instance from the Protected state	430
4.5. Scaling configurations	430
4.5.1. Create a scaling configuration	430
4.5.2. View scaling configurations	432
4.5.3. Modify a scaling configuration	433
4.5.4. Apply a scaling configuration	433
4.5.5. Delete a scaling configuration	433
4.6. Scaling rules	434
4.6.1. Create a scaling rule	434
4.6.2. View scaling rules	434
4.6.3. Modify a scaling rule	435
4.6.4. Delete a scaling rule	435
4.7. Scaling tasks	435
4.7.1. Manually execute a scaling rule	435
4.7.2. Manually add an ECS instance	436
4.7.3. Manually remove an ECS instance	437
4.8. Scheduled tasks	437
4.8.1. Create a scheduled task	437
4.8.2. View scheduled tasks	438
4.8.3. Modify a scheduled task	439
4.8.4. Disable a scheduled task	439
4.8.5. Enable a scheduled task	439

4.8.6. Delete a scheduled task	440
4.9. Event-triggered tasks	440
4.9.1. Create an event-triggered task	440
4.9.2. View event-triggered tasks	441
4.9.3. Modify an event-triggered task	442
4.9.4. Disable an event-triggered task	442
4.9.5. Enable an event-triggered task	442
4.9.6. Delete an event-triggered task	443
5.Resource Orchestration Service (ROS)	444
5.1. What is ROS?	444
5.2. Log on to the ROS console	444
5.3. Create a stack	445
5.4. Template syntax	445
5.4.1. Template structure	445
5.4.2. Parameters	447
5.4.3. Resources	450
5.4.4. Outputs	454
5.4.5. Functions	457
5.4.6. Mappings	480
5.4.7. Conditions	482
5.5. Resource types	484
5.5.1. ECS	484
5.5.1.1. ALIYUN::ECS::AutoSnapshotPolicy	484
5.5.1.2. ALIYUN::ECS::BandwidthPackage	489
5.5.1.3. ALIYUN::ECS::Command	491
5.5.1.4. ALIYUN::ECS::CustomImage	494
5.5.1.5. ALIYUN::ECS::DedicatedHost	499
5.5.1.6. ALIYUN::ECS::Disk	506

5.5.1.7. ALIYUN::ECS::DiskAttachment	510
5.5.1.8. ALIYUN::ECS::ForwardEntry	513
5.5.1.9. ALIYUN::ECS::Instance	514
5.5.1.10. ALIYUN::ECS::InstanceClone	522
5.5.1.11. ALIYUN::ECS::InstanceGroup	529
5.5.1.12. ALIYUN::ECS::InstanceGroupClone	540
5.5.1.13. ALIYUN::ECS::Invocation	549
5.5.1.14. ALIYUN::ECS::JoinSecurityGroup	552
5.5.1.15. ALIYUN::ECS::LaunchTemplate	553
5.5.1.16. ALIYUN::ECS::NatGateway	564
5.5.1.17. ALIYUN::ECS::NetworkInterface	566
5.5.1.18. ALIYUN::ECS::NetworkInterfaceAttachment	569
5.5.1.19. ALIYUN::ECS::NetworkInterfacePermission	571
5.5.1.20. ALIYUN::ECS::Route	574
5.5.1.21. ALIYUN::ECS::SNatEntry	576
5.5.1.22. ALIYUN::ECS::SecurityGroup	577
5.5.1.23. ALIYUN::ECS::SecurityGroupClone	591
5.5.1.24. ALIYUN::ECS::SecurityGroupEgress	594
5.5.1.25. ALIYUN::ECS::SecurityGroupIngress	599
5.5.1.26. ALIYUN::ECS::Snapshot	605
5.5.1.27. ALIYUN::ECS::SSHKeyPair	607
5.5.1.28. ALIYUN::ECS::SSHKeyPairAttachment	609
5.5.1.29. ALIYUN::ECS::VPC	611
5.5.1.30. ALIYUN::ECS::VSwitch	613
5.5.2. ESS	617
5.5.2.1. ALIYUN::ESS::AlarmTask	617
5.5.2.2. ALIYUN::ESS::AlarmTaskEnable	623
5.5.2.3. ALIYUN::ESS::LifecycleHook	624

5.5.2.4. ALIYUN::ESS::ScalingConfiguration	630
5.5.2.5. ALIYUN::ESS::ScalingGroup	636
5.5.2.6. ALIYUN::ESS::ScalingGroupEnable	644
5.5.2.7. ALIYUN::ESS::ScalingRule	646
5.5.2.8. ALIYUN::ESS::ScheduledTask	650
5.5.3. OSS	654
5.5.3.1. ALIYUN::OSS::Bucket	654
5.5.4. RDS	663
5.5.4.1. ALIYUN::RDS::Account	663
5.5.4.2. ALIYUN::RDS::AccountPrivilege	666
5.5.4.3. ALIYUN::RDS::DBInstance	669
5.5.4.4. ALIYUN::RDS::DBInstanceParameterGroup	677
5.5.4.5. ALIYUN::RDS::DBInstanceSecurityIps	679
5.5.4.6. ALIYUN::RDS::PrepayDBInstance	682
5.5.5. ROS	696
5.5.5.1. ALIYUN::ROS::WaitCondition	696
5.5.5.2. ALIYUN::ROS::WaitConditionHandle	698
5.5.5.3. ALIYUN::ROS::Stack	701
5.5.6. SLB	710
5.5.6.1. ALIYUN::SLB::AccessControl	710
5.5.6.2. ALIYUN::SLB::BackendServerAttachment	714
5.5.6.3. ALIYUN::SLB::BackendServerToVServerGroupAdditio...	716
5.5.6.4. ALIYUN::SLB::Certificate	718
5.5.6.5. ALIYUN::SLB::DomainExtension	721
5.5.6.6. ALIYUN::SLB::Listener	723
5.5.6.7. ALIYUN::SLB::LoadBalancer	736
5.5.6.8. ALIYUN::SLB::LoadBalancerClone	741
5.5.6.9. ALIYUN::SLB::MasterSlaveServerGroup	745

5.5.6.10. ALIYUN::SLB::Rule	748
5.5.6.11. ALIYUN::SLB::VServerGroup	752
5.5.7. VPC	754
5.5.7.1. ALIYUN::VPC::EIP	754
5.5.7.2. ALIYUN::VPC::EIPAssociation	757
5.5.7.3. ALIYUN::VPC::PeeringRouterInterfaceBinding	760
5.5.7.4. ALIYUN::VPC::PeeringRouterInterfaceConnection	761
5.5.7.5. ALIYUN::VPC::RouterInterface	762
6.Object Storage Service (OSS)	769
6.1. What is OSS?	769
6.2. Usage notes	769
6.3. Quick start	769
6.3.1. Log on to the OSS console	769
6.3.2. Create buckets	770
6.3.3. Upload objects	772
6.3.4. Obtain object URLs	773
6.4. Buckets	773
6.4.1. View bucket information	773
6.4.2. Delete buckets	773
6.4.3. Modify bucket ACLs	774
6.4.4. Configure static website hosting	774
6.4.5. Configure hotlink protection	775
6.4.6. Configure logging	775
6.4.7. Configure CORS	776
6.4.8. Manage lifecycle rules	777
6.4.9. Configure storage quota	778
6.4.10. Configure cluster-disaster recovery	779
6.4.11. Configure CRR	779

6.4.12. Access OSS through custom domain names	781
6.5. Objects	782
6.5.1. Search for objects	782
6.5.2. Configure object ACL	782
6.5.3. Create folders	783
6.5.4. Delete objects	783
6.5.5. Manage parts	784
7.Apsara File Storage NAS	785
7.1. What is Apsara File Storage NAS?	785
7.2. Precautions	785
7.3. Quick start	786
7.3.1. Log on to the Apsara File Storage NAS console	786
7.3.2. Create a file system	787
7.3.3. Create a permission group and add rules	789
7.3.4. Add a mount target	791
7.3.5. Mount an NFS file system	793
7.3.6. Mount an SMB file system	795
7.4. File systems	798
7.4.1. View the details of a file system	798
7.4.2. Delete a file system	799
7.5. Mount targets	799
7.5.1. View mount targets	799
7.5.2. Enable or disable a mount target	799
7.5.3. Delete a mount target	800
7.5.4. Modify the permission group of a mount target	800
7.6. Permission groups	801
7.6.1. View permission groups	801
7.6.2. Delete a permission group	801

7.6.3. Manage permission group rules	801
7.7. Manage quotas	802
7.8. Create and manage a unified namespace	805
7.9. Manage the file lifecycle	808
7.10. Directory-level ACLs that grant the read and write acce...	810
7.10.1. Overview	810
7.10.2. Features	811
7.10.3. Use POSIX ACLs to control access	821
7.10.4. Use NFSv4 ACLs to control access	824
8. Tablestore	828
8.1. What is Tablestore?	828
8.2. Precautions	828
8.3. Quick start	829
8.3.1. Log on to the Tablestore console	829
8.3.2. Create instances	830
8.3.3. Create tables	830
8.3.4. Read and write data in the console	833
8.3.5. Bind a VPC to a Tablestore instance	835
9. ApsaraDB RDS for MySQL	836
9.1. What is ApsaraDB for RDS?	836
9.2. Log on to the ApsaraDB for RDS console	836
9.3. Quick start	836
9.3.1. Limits	836
9.3.2. Procedure	837
9.3.3. Create an instance	838
9.3.4. Initialization settings	840
9.3.4.1. Configure a whitelist	840
9.3.4.2. Create an account	842

9.3.4.3. Create a database	845
9.3.5. Connect to an ApsaraDB RDS for MySQL instance	846
9.4. Instances	847
9.4.1. Create an instance	847
9.4.2. Create an ApsaraDB RDS for MySQL instance with st...	849
9.4.3. View basic information of an instance	850
9.4.4. Restart an instance	850
9.4.5. Change the specifications of an instance	851
9.4.6. Set a maintenance window	851
9.4.7. Change the data replication mode	852
9.4.8. Release an instance	853
9.4.9. Upgrade the kernel version of an instance	853
9.4.10. Modify parameters of an instance	854
9.4.11. Read-only instances	855
9.4.11.1. Overview of read-only instances	855
9.4.11.2. Create a read-only instance	856
9.4.11.3. View details of read-only instances	857
9.5. Accounts	858
9.5.1. Create an account	858
9.5.2. Reset the password	862
9.5.3. Modify account permissions	863
9.5.4. Delete an account	863
9.6. Databases	864
9.6.1. Create a database	864
9.6.2. Delete a database	864
9.7. Database connection	865
9.7.1. Change the endpoint and port number of an instanc...	865
9.7.2. Log on to an ApsaraDB for RDS instance by using D...	865

9.7.3. Switch the access mode	866
9.7.4. Hybrid access from both the classic network and VP...	867
9.7.5. Change the network type of an instance	869
9.8. Database proxy	869
9.8.1. Database proxy	870
9.8.2. Dedicated proxy	870
9.8.3. Short-lived connection optimization	874
9.8.4. Transaction splitting	874
9.8.5. Read/write splitting	875
9.8.5.1. Enable read/write splitting	876
9.8.5.2. Set the read/write splitting parameters	878
9.8.5.3. Disable read/write splitting	879
9.9. Monitoring and alerts	880
9.9.1. View resource and engine monitoring data	880
9.9.2. Set a monitoring frequency	881
9.10. Data security	882
9.10.1. Configure a whitelist	882
9.10.2. Configure SSL encryption	884
9.10.3. Configure TDE	887
9.10.4. SQL audit	889
9.11. Service availability	890
9.11.1. Automatically or manually switch over services betw...	891
9.11.2. Change the data replication mode	891
9.12. Database backup and restoration	892
9.12.1. Automatic backup	892
9.12.2. Manual backup	893
9.12.3. Download data and log backup files	894
9.12.4. Upload binlogs	895

9.12.5. Restore data to a new instance (formerly known a...	896
9.13. CloudDBA	898
9.13.1. Introduction to CloudDBA	898
9.13.2. Diagnostics	899
9.13.3. Instance sessions	899
9.13.4. Real-time monitoring	899
9.13.5. Storage analysis	900
9.13.6. Deadlock analysis	900
9.13.7. Dashboard	900
9.13.8. Slow query logs	901
9.13.9. Diagnostic reports	901
9.14. Logs	901
9.15. Use mysqldump to migrate MySQL data	902
10. ApsaraDB RDS for SQL Server	905
10.1. What is ApsaraDB for RDS?	905
10.2. Log on to the ApsaraDB for RDS console	905
10.3. Quick Start	905
10.3.1. Procedure	905
10.3.2. Create an instance	906
10.3.3. Configure a whitelist	907
10.3.4. Connect to an instance	909
10.3.5. Create an account	910
10.3.5.1. Create an account for an ApsaraDB for RDS inst...	910
10.3.5.2. Create an account for an ApsaraDB for RDS ins...	911
10.3.6. Create a database	912
10.4. Instances	913
10.4.1. Create an instance	913
10.4.2. View basic information of an instance	914

10.4.3. Restart an instance	915
10.4.4. Change the specifications of an instance	915
10.4.5. Set the maintenance window of an instance	915
10.4.6. Switch over services between a primary RDS instan...	916
10.4.7. Release an instance	917
10.4.8. Upgrade the SQL Server version of an instance fro...	917
10.5. Accounts	918
10.5.1. Create an account for an ApsaraDB for RDS instanc...	918
10.5.2. Create an account for an ApsaraDB for RDS instan...	920
10.5.3. Reset the password	921
10.6. Databases	922
10.6.1. Create a database	922
10.6.2. Delete a database	922
10.6.3. Change the character set collation and the time zo...	923
10.7. Database connection	926
10.7.1. Change the endpoint and port number of an instan...	926
10.7.2. Connect to an instance	926
10.8. Monitoring and alerting	927
10.8.1. Set a monitoring frequency	927
10.8.2. View resource and engine monitoring data	927
10.9. Data security	928
10.9.1. Configure a whitelist	929
10.9.2. Configure SSL encryption for an instance	930
10.9.3. Configure TDE	932
10.10. Logs	933
10.11. Database backup and restoration	934
10.11.1. Configure an automatic backup policy	934
10.11.2. Manually back up an instance	934

10.11.3. Shrink transaction logs	935
10.12. Migrate full backup data to ApsaraDB RDS for SQL Se... ..	935
11.PolarDB	939
11.1. What is ApsaraDB for RDS?	939
11.2. Limits on PolarDB	939
11.3. Log on to the ApsaraDB for RDS console	939
11.4. Quick Start	940
11.4.1. Procedure	940
11.4.2. Create an instance	941
11.4.3. Configure an IP whitelist	943
11.4.4. Create a database and an account	943
11.4.5. Connect to a PolarDB instance	946
11.5. Instances	947
11.5.1. Create an instance	947
11.5.2. Restart an instance	948
11.5.3. Set the maintenance window	948
11.5.4. Configure primary/secondary switchover	949
11.5.5. Change the network type	950
11.5.6. Release an instance	952
11.5.7. Change the specifications of an instance	952
11.5.8. Modify parameters of an instance	952
11.6. Database connection	953
11.6.1. Connect to a PolarDB instance	953
11.6.2. Hybrid access from both the classic network and V... ..	955
11.6.3. Log on to an ApsaraDB for RDS instance by using	956
11.6.4. View and modify the internal endpoint and port n... ..	957
11.7. Accounts	958
11.7.1. Create an account	958

11.7.2. Reset the password	961
11.8. Databases	961
11.8.1. Create a database	961
11.8.2. Delete a database	963
11.9. Network, VPC, and VSwitch	965
11.9.1. Change the network type	965
11.9.2. Hybrid access from both the classic network and V...	967
11.10. Monitoring	968
11.10.1. View monitoring data	969
11.10.2. Set the monitoring frequency	969
11.11. Data security	969
11.11.1. Switch to the enhanced whitelist mode	969
11.11.2. Configure an IP whitelist	970
11.12. Backup	971
11.12.1. Back up data	971
11.12.2. Download backup files	973
11.13. Logs	974
11.14. Plug-ins supported	974
11.15. PolarDB development driver	975
11.16. Compatibility for Oracle	976
11.17. Management functions	988
12.ApsaraDB RDS for PostgreSQL	991
12.1. What is ApsaraDB for RDS?	991
12.2. Limits	991
12.3. Log on to the ApsaraDB for RDS console	991
12.4. Quick Start	992
12.4.1. Procedure	992
12.4.2. Create an instance	993

12.4.3. Configure an IP address whitelist	995
12.4.4. Create a database and an account	995
12.4.5. Connect to an ApsaraDB RDS for PostgreSQL instan... ..	998
12.5. Instances	999
12.5.1. Create an instance	999
12.5.2. View basic information of an instance	1001
12.5.3. Restart an instance	1001
12.5.4. Change the specifications of an instance	1001
12.5.5. Set a maintenance window	1001
12.5.6. Configure primary/secondary switchover	1002
12.5.7. Release an instance	1003
12.5.8. Modify parameters of an instance	1003
12.5.9. Read-only instances	1004
12.5.9.1. Overview of read-only ApsaraDB RDS for Postgre... ..	1004
12.5.9.2. Create a read-only ApsaraDB RDS for PostgreSQ... ..	1006
12.5.9.3. View a read-only ApsaraDB RDS for PostgreSQL	1007
12.6. Database connection	1008
12.6.1. Connect to an ApsaraDB RDS for PostgreSQL instan... ..	1008
12.6.2. Log on to an ApsaraDB for RDS instance by using	1009
12.6.3. View and modify the internal endpoint and port n... ..	1010
12.7. Accounts	1011
12.7.1. Create an account	1011
12.7.2. Reset the password	1014
12.8. Databases	1014
12.8.1. Create a database	1014
12.8.2. Delete a database	1015
12.9. Network, VPC, and VSwitch	1016
12.9.1. Change the network type of an ApsaraDB RDS for	1016

12.9.2. Hybrid access from both the classic network and V...	1018
12.10. Monitoring	1020
12.10.1. View monitored resources	1020
12.10.2. Set the monitoring frequency	1020
12.11. Data security and encryption	1021
12.11.1. Switch to the enhanced whitelist mode	1021
12.11.2. Configure an IP address whitelist	1022
12.12. Log and audit	1022
12.12.1. SQL audit (database audit)	1023
12.12.2. Manage logs	1023
12.13. Backup	1024
12.13.1. Back up an ApsaraDB RDS for PostgreSQL instance	1024
12.13.2. Download data and log backup files	1025
12.13.3. Create a logical backup for an ApsaraDB RDS for ...	1026
12.13.4. Create a full backup of an ApsaraDB RDS for Post...	1030
12.14. Restoration	1031
12.14.1. Restore data of an ApsaraDB RDS for PostgreSQL i...	1032
12.14.2. Restore data from a logical backup file	1033
12.15. Plug-ins	1036
12.15.1. Plug-ins supported	1036
12.15.2. Use mysql_fdw to read and write data from and t...	1040
12.15.3. Read and write foreign data files by using oss_fdw	1043
12.16. Use Pgpool for read/write splitting in ApsaraDB RDS f...	1047
12.17. Use ShardingSphere to develop ApsaraDB RDS for Pos...	1083
13.Cloud Native Distributed Database PolarDB-X	1091
13.1. What is PolarDB-X?	1091
13.2. Quick start	1091
13.3. Log on to the PolarDB-X console	1092

13.4. Instance management	1092
13.4.1. Create a PolarDB-X instance	1092
13.4.2. Change specifications	1093
13.4.3. Read-only PolarDB-X instances	1094
13.4.3.1. Overview	1094
13.4.3.3. Create a read-only PolarDB-X instance	1094
13.4.3.4. Manage a read-only PolarDB-X instance	1095
13.4.3.5. Release a read-only PolarDB-X instance	1096
13.4.4. Restart a PolarDB-X instance	1096
13.4.5. Release a PolarDB-X instance	1096
13.4.6. Recover data	1097
13.4.6.1. Backup and recovery	1097
13.4.6.2. Configure an automatic backup policy	1098
13.4.6.3. Configure local logs	1099
13.4.6.4. Manual backup	1099
13.4.6.5. Recover data	1099
13.4.6.6. SQL flashback	1100
13.4.6.6.1. Overview	1100
13.4.6.6.2. Generate a recovery file	1101
13.4.6.6.3. Rollback SQL statements and original SQL st... ..	1102
13.4.6.6.4. Exact match and fuzzy match	1103
13.4.6.7. Table recycle bin	1103
13.4.6.7.1. Overview	1103
13.4.6.7.2. Enable the table recycle bin	1104
13.4.6.7.3. Recover tables	1104
13.4.6.7.4. Delete tables from the recycle bin	1104
13.4.6.7.5. Disable the table recycle bin	1105
13.4.7. Set parameters	1105

13.4.8. SQL audit and analysis	1107
13.4.8.1. Description	1107
13.4.8.2. Enable SQL audit and analysis	1109
13.4.8.3. Log fields	1110
13.4.8.4. Log analysis	1111
13.4.8.5. Log reports	1116
13.4.9. Monitor PolarDB-X instances	1122
13.4.9.1. View monitoring information	1122
13.4.9.2. Monitoring metrics	1122
13.4.9.3. How metrics work	1124
13.4.9.4. Prevent performance problems	1124
13.4.9.4.1. Example 1: PolarDB-X CPU utilization	1124
13.4.9.4.2. Example 2: Logical RT and physical RT	1125
13.4.9.4.3. Example 3: Logical QPS and physical QPS	1127
13.4.9.4.4. Example 4: High memory usage	1128
13.4.10. View the instance version	1129
13.5. Account management	1129
13.5.1. Basic concepts	1129
13.5.2. Create an account	1131
13.5.3. Reset password	1132
13.5.4. Modify account permissions	1133
13.5.5. Delete an account	1136
13.6. Database management	1136
13.6.1. Create a database	1136
13.6.2. View a database	1137
13.6.3. Perform smooth scale-out	1138
13.6.4. View database monitoring information	1140
13.6.5. Set the IP address whitelist	1140

13.6.6. Delete a database	1141
13.6.7. Fix database shard connections	1141
13.7. Custom control commands	1142
13.7.1. Overview	1142
13.7.2. Help statements	1142
13.7.3. Statements for viewing rules and node topologies	1143
13.7.4. SQL tuning statements	1149
13.7.5. Statistics query statements	1157
13.7.6. SHOW PROCESSLIST and KILL commands	1160
13.7.7. SHOW PROCESSLIST and KILL commands in earlier ve...	1163
13.8. Custom hints	1165
13.8.1. Introduction to hints	1165
13.8.2. Read/write splitting	1167
13.8.3. Specify a timeout period for an SQL statement	1168
13.8.4. Specify a database shard to run an SQL statement	1169
13.8.5. Scan all or some of database shards and table sh...	1172
13.8.6. INDEX HINT	1174
13.9. PolarDB-X 5.2 hints	1176
13.9.1. Introduction to hints	1176
13.9.2. Read/write splitting	1176
13.9.3. Prevent the delay from a read-only ApsaraDB RDS ...	1177
13.9.4. Specify a timeout period for an SQL statement	1178
13.9.5. Specify a database shard to run an SQL statement	1179
13.9.6. Scan all database shards and table shards	1183
13.10. Distributed transactions	1184
13.10.1. Distributed transactions based on MySQL 5.7	1184
13.10.2. Distributed transactions based on MySQL 5.6	1185
13.11. DDL operations	1186

13.11.1. DDL statements	1186
13.11.2. CREATE TABLE statement	1187
13.11.2.1. Overview	1187
13.11.2.2. Create a single-database non-partition table	1188
13.11.2.3. Create a non-partition table in database shards	1188
13.11.2.4. Create table shards in database shards	1189
13.11.2.5. Use the primary key as the shard key	1201
13.11.2.6. Create a broadcast table	1201
13.11.2.7. Other attributes of the MySQL CREATE TABLE sta...	1202
13.11.3. ALTER TABLE statement	1202
13.11.4. DROP TABLE statement	1203
13.11.5. FAQ about DDL statements	1203
13.11.6. DDL functions for sharding	1204
13.11.6.1. Overview	1204
13.11.6.2. HASH	1206
13.11.6.3. UNI_HASH	1207
13.11.6.4. RIGHT_SHIFT	1209
13.11.6.5. RANGE_HASH	1210
13.11.6.6. MM	1210
13.11.6.7. DD	1211
13.11.6.8. WEEK	1212
13.11.6.9. MMDD	1212
13.11.6.10. YYYYMM	1213
13.11.6.11. YYYYWEEK	1214
13.11.6.12. YYYYDD	1215
13.11.6.13. YYYYMM_OPT	1216
13.11.6.14. YYYYWEEK_OPT	1218
13.11.6.15. YYYYDD_OPT	1218

13.12. Automatic protection of high-risk SQL statements	1219
13.13. PolarDB-X sequence	1220
13.13.1. Overview	1220
13.13.2. Explicit sequence usage	1223
13.13.3. Implicit sequence usage	1227
13.13.4. Limits and precautions for sequences	1228
13.14. Best practices	1229
13.14.1. Select a shard key	1229
13.14.2. Select the number of shards	1231
13.14.3. Basic concepts of SQL optimization	1232
13.14.4. SQL optimization methods	1236
13.14.4.1. Overview	1236
13.14.4.2. Single-table SQL optimization	1237
13.14.4.3. JOIN query optimization	1242
13.14.4.4. Subquery optimization	1246
13.14.5. Select connection pools for an application	1246
13.14.6. Connections to PolarDB-X instances	1247
13.14.7. Perform instance upgrade	1249
13.14.8. Perform scale-out	1250
13.14.9. Troubleshoot slow SQL statements in PolarDB-X	1252
13.14.9.1. Details about a low SQL statement	1252
13.14.9.2. Locate slow SQL statements	1255
13.14.9.3. Locate nodes with performance loss	1256
13.14.9.4. Troubleshoot the performance loss	1259
13.14.10. Handle DDL exceptions	1259
13.14.11. Efficiently scan PolarDB-X data	1263
13.15. Appendix: PolarDB-X terms	1264
14. AnalyticDB for MySQL	1272

14.1. What is AnalyticDB for MySQL?	1272
14.2. Limits	1272
14.3. Quick start	1273
14.3.1. Log on to the AnalyticDB for MySQL console	1273
14.3.2. Create a database cluster	1274
14.3.3. Configure a whitelist	1275
14.3.4. Create a database account	1275
14.3.5. Create a database	1277
14.3.6. Connect to a database cluster	1277
14.3.7. Apply for a public endpoint	1278
14.3.8. Synchronize data	1278
14.4. Connect to a database cluster	1279
14.4.1. Use the MySQL command-line tool to connect to An...	1279
14.4.2. Use the code in a business system to connect to A...	1279
14.4.2.1. C#	1279
14.4.2.2. PHP	1280
14.4.2.3. Python	1282
14.4.2.4. Druid connection pool	1283
14.4.2.5. Java	1284
14.4.3. Enable PreparedStatement for a client in different p...	1288
14.4.4. Use a client to connect to AnalyticDB for MySQL	1288
14.4.4.1. SQL Workbench/J	1288
14.4.4.2. DbVisualizer	1289
14.4.4.3. DBeaver	1291
14.4.4.4. Navicat	1293
14.5. Manage database clusters	1294
14.5.1. View monitoring information	1294
14.5.2. Change specifications	1295

14.5.3. Delete a cluster	1295
14.6. Backup and restoration	1295
14.6.1. Back up data	1295
14.6.2. Restore data	1296
14.7. Diagnostics and optimization	1296
14.7.1. Use functions related to slow SQL queries	1296
14.8. Account and permission management	1296
14.8.1. Permission model	1296
14.8.2. Manage database accounts and permissions	1298
14.9. Data visualization	1298
14.9.1. Tableau	1298
14.9.2. QlikView	1299
14.9.3. FineReport	1300
14.10. Data migration and synchronization	1301
14.10.1. Use Kettle to synchronize local data to AnalyticDB...	1301
14.11. SQL manual	1303
14.11.1. Data types	1303
14.11.2. Data definition statements	1305
14.11.2.1. CREATE DATABASE	1305
14.11.2.2. CREATE TABLE	1306
14.11.2.3. ALTER TABLE	1310
14.11.2.4. CREATE VIEW	1312
14.11.2.5. DROP DATABASE	1313
14.11.2.6. DROP TABLE	1313
14.11.2.7. DROP VIEW	1314
14.11.3. Data manipulation statements	1314
14.11.3.1. INSERT INTO	1314
14.11.3.2. REPLACE INTO	1316

14.11.3.3. INSERT SELECT FROM	1316
14.11.3.4. REPLACE SELECT FROM	1317
14.11.3.5. INSERT OVERWRITE INTO SELECT	1317
14.11.3.6. UPDATE	1318
14.11.3.7. DELETE	1318
14.11.3.8. TRUNCATE TABLE	1319
14.11.3.9. KILL PROCESS	1319
14.11.3.10. SHOW PROCESSLIST	1319
14.11.4. SELECT	1320
14.11.4.1. Syntax	1320
14.11.4.2. WITH	1321
14.11.4.3. GROUP BY	1322
14.11.4.4. HAVING	1324
14.11.4.5. JOIN	1325
14.11.4.6. LIMIT	1326
14.11.4.7. ORDER BY	1326
14.11.4.8. Subqueries	1327
14.11.4.9. UNION, INTERSECT, and EXCEPT	1327
14.11.5. CREATE USER	1328
14.11.6. GRANT	1329
14.11.7. REVOKE	1330
14.11.8. Query users	1330
14.11.9. RENAME USER	1331
14.11.10. DROP USER	1331
14.11.11. SHOW	1331
14.12. System functions	1333
14.12.1. Aggregate functions	1333
14.12.1.1. AVG	1333

14.12.1.2. BIT_AND	1333
14.12.1.3. BIT_OR	1334
14.12.1.4. BIT_XOR	1334
14.12.1.5. COUNT	1334
14.12.1.6. MAX	1335
14.12.1.7. MIN	1335
14.12.1.8. STD/STDDEV	1336
14.12.1.9. STDDEV_POP	1336
14.12.1.10. STDDEV_SAMP	1336
14.12.1.11. SUM	1337
14.12.1.12. VAR_POP	1337
14.12.1.13. VAR_SAMP	1337
14.12.1.14. VARIANCE	1338
14.12.2. Date and time functions	1338
14.12.2.1. ADDDATE/DATE_ADD	1338
14.12.2.2. ADDTIME	1339
14.12.2.3. CONVERT_TZ	1340
14.12.2.4. CURDATE	1340
14.12.2.5. CURTIME	1341
14.12.2.6. DATE	1341
14.12.2.7. DATE_FORMAT	1341
14.12.2.8. SUBDATE/DATE_SUB	1343
14.12.2.9. DATEDIFF	1344
14.12.2.10. DAY/DAYOFMONTH	1344
14.12.2.11. DAYNAME	1345
14.12.2.12. DAYOFWEEK	1345
14.12.2.13. DAYOFYEAR	1346
14.12.2.14. EXTRACT	1346

14.12.2.15. FROM_DAYS	1346
14.12.2.16. FROM_UNIXTIME	1347
14.12.2.17. HOUR	1347
14.12.2.18. LAST_DAY	1348
14.12.2.19. LOCALTIME/LOCALTIMESTAMP/NOW	1348
14.12.2.20. MAKEDATE	1349
14.12.2.21. MAKETIME	1349
14.12.2.22. MINUTE	1349
14.12.2.23. MONTH	1350
14.12.2.24. MONTHNAME	1350
14.12.2.25. PERIOD_ADD	1351
14.12.2.26. PERIOD_DIFF	1351
14.12.2.27. QUARTER	1352
14.12.2.28. SEC_TO_TIME	1352
14.12.2.29. SECOND	1352
14.12.2.30. STR_TO_DATE	1353
14.12.2.31. SUBTIME	1353
14.12.2.32. SYSDATE	1354
14.12.2.33. TIME	1354
14.12.2.34. TIME_FORMAT	1355
14.12.2.35. TIME_TO_SEC	1355
14.12.2.36. TIMEDIFF	1355
14.12.2.37. TIMESTAMP	1356
14.12.2.38. TIMESTAMPADD	1356
14.12.2.39. TIMESTAMPDIFF	1357
14.12.2.40. TO_DAYS	1357
14.12.2.41. TO_SECONDS	1358
14.12.2.42. UNIX_TIMESTAMP	1358

14.12.2.43. UTC_DATE	1359
14.12.2.44. UTC_TIME	1359
14.12.2.45. UTC_TIMESTAMP	1359
14.12.2.46. WEEK	1360
14.12.2.47. WEEKDAY	1361
14.12.2.48. WEEKOFYEAR	1361
14.12.2.49. YEAR	1362
14.12.2.50. YEARWEEK	1362
14.12.3. String functions	1363
14.12.3.1. ASCII	1363
14.12.3.2. BIN	1363
14.12.3.3. BIT_LENGTH	1363
14.12.3.4. CHAR	1364
14.12.3.5. CHAR_LENGTH/CHARACTER_LENGTH	1364
14.12.3.6. CONCAT	1364
14.12.3.7. CONCAT_WS	1365
14.12.3.8. ELT	1365
14.12.3.9. EXPORT_SET	1365
14.12.3.10. FIELD	1366
14.12.3.11. FIND_IN_SET	1366
14.12.3.12. FORMAT	1366
14.12.3.13. HEX	1367
14.12.3.14. INSTR	1367
14.12.3.15. LEFT	1367
14.12.3.16. LENGTH/OCTET_LENGTH	1368
14.12.3.17. LIKE	1368
14.12.3.18. LOCATE	1368
14.12.3.19. LOWER/LCASE	1369

14.12.3.20. LPAD	1369
14.12.3.21. LTRIM	1369
14.12.3.22. MAKE_SET	1370
14.12.3.23. MID	1370
14.12.3.24. OCT	1370
14.12.3.25. POSITION	1371
14.12.3.26. REPEAT	1371
14.12.3.27. REPLACE	1371
14.12.3.28. REVERSE	1372
14.12.3.29. RIGHT	1372
14.12.3.30. RLIKE/REGEXP	1372
14.12.3.31. RPAD	1373
14.12.3.32. RTRIM	1373
14.12.3.33. SPACE	1373
14.12.3.34. STRCMP	1374
14.12.3.35. SUBSTR/SUBSTRING	1374
14.12.3.36. SUBSTRING_INDEX	1374
14.12.3.37. TRIM	1375
14.12.3.38. UPPER/UCASE	1375
14.12.4. Numeric functions	1375
14.12.4.1. ABS	1376
14.12.4.2. ROUND	1376
14.12.4.3. SQRT	1377
14.12.4.4. LN	1377
14.12.4.5. LOG	1377
14.12.4.6. LOG2	1377
14.12.4.7. PI	1378
14.12.4.8. LOG10	1378

14.12.4.9. POWER/POW	1378
14.12.4.10. RADIANS	1379
14.12.4.11. DEGREES	1379
14.12.4.12. SIGN	1379
14.12.4.13. CEILING/CEIL	1380
14.12.4.14. FLOOR	1380
14.12.4.15. EXP	1380
14.12.4.16. COS	1381
14.12.4.17. ACOS	1381
14.12.4.18. TAN	1381
14.12.4.19. ATAN	1382
14.12.4.20. ATAN2	1382
14.12.4.21. COT	1382
14.12.4.22. ASIN	1383
14.12.4.23. SIN	1383
14.12.5. Arithmetic operators	1383
14.12.6. Bit functions and operators	1385
14.12.7. Control flow functions	1387
15. AnalyticDB for PostgreSQL	1390
15.1. What is AnalyticDB for PostgreSQL?	1390
15.2. Quick start	1390
15.2.1. Overview	1390
15.2.2. Log on to the AnalyticDB for PostgreSQL console	1390
15.2.3. Create an instance	1391
15.2.4. Configure a whitelist	1392
15.2.5. Create an initial account	1393
15.2.6. Obtain the client tool	1393
15.2.7. Connect to a database	1394

15.3. Instances	1398
15.3.1. Reset the password	1398
15.3.2. View monitoring information	1399
15.3.3. Switch the network type of an instance	1399
15.3.4. Restart an instance	1399
15.3.5. Import data	1400
15.3.5.1. Import or export data from or to OSS in parallel	1400
15.3.5.2. Import data from MySQL	1408
15.3.5.3. Import data from PostgreSQL	1410
15.3.5.4. Import data by using the \COPY statement	1411
15.4. Databases	1412
15.4.1. Overview	1412
15.4.2. Create a database	1412
15.4.3. Create a partition key	1413
15.4.4. Construct data	1413
15.4.5. Query data	1414
15.4.6. Manage extensions	1414
15.4.7. Manage users and permissions	1415
15.4.8. Manage JSON data	1416
15.4.9. Use HyperLogLog	1424
15.4.10. Use the CREATE LIBRARY statement	1426
15.4.11. Create and use the PL/Java UDF	1427
15.5. Table	1428
15.5.1. Create a table	1429
15.5.2. Principles and scenarios of row store, column stor...	1435
15.5.3. Enable the column store and compression features	1436
15.5.4. Add a field to a column store table and set the d...	1437
15.5.5. Configure the table partition	1439

15.5.6. Configure the sort key	1440
15.6. Best practices	1442
15.6.1. Configure memory and load parameters	1442
16.KVStore for Redis	1450
16.1. What is KVStore for Redis?	1450
16.2. Quick Start	1450
16.2.1. Get started with KVStore for Redis	1450
16.2.2. Log on to the KVStore for Redis console	1451
16.2.3. Create an instance	1451
16.2.4. Configure a whitelist	1453
16.2.5. Connect to an instance	1455
16.2.5.1. Use a Redis client	1455
16.2.5.2. Use redis-cli	1468
16.3. Instance management	1469
16.3.1. Change the password	1469
16.3.2. Configure a whitelist	1469
16.3.3. Change configurations	1471
16.3.4. Set a maintenance window	1472
16.3.5. Upgrade the minor version	1473
16.3.6. Configure SSL encryption	1473
16.3.7. Clear data	1474
16.3.8. Release an instance	1474
16.3.9. Manage database accounts	1474
16.3.10. Use a Lua script	1475
16.3.11. Restart an instance	1476
16.3.12. Export the list of instances	1476
16.4. Connection management	1476
16.4.1. View connection strings	1476

16.4.2. Apply for a public endpoint	1477
16.4.3. Change the connection string of an instance	1477
16.5. Parameter configuration	1478
16.6. Backup and recovery	1483
16.6.1. Back up data automatically	1483
16.6.2. Back up data manually	1483
16.6.3. Download backup files	1483
16.6.4. Restore data	1484
16.6.5. Clone an instance	1484
16.7. Performance monitoring	1484
16.7.1. View monitoring data	1484
16.7.2. Customize metrics	1485
16.7.3. Modify monitoring frequency	1486
16.7.4. Monitoring metrics	1486
17. ApsaraDB for MongoDB	1490
17.1. Before you start	1490
17.2. Log on to the ApsaraDB for MongoDB console	1490
17.3. Quick start	1491
17.3.1. Use ApsaraDB for MongoDB	1491
17.3.2. Create an ApsaraDB for MongoDB instance	1491
17.3.3. Configure a whitelist for an ApsaraDB for MongoDB...	1493
17.3.4. Overview of replica set instance connections	1494
17.3.5. Use the mongo shell to connect to a replica set in...	1495
17.4. Instances	1496
17.4.1. Create an ApsaraDB for MongoDB instance	1496
17.4.2. View the details of an ApsaraDB for MongoDB insta...	1498
17.4.3. Restart an ApsaraDB for MongoDB instance	1498
17.4.4. Change the specifications of an ApsaraDB for Mong...	1499

17.4.5. Change the name of an ApsaraDB for MongoDB in...	1499
17.4.6. Reset the password for an ApsaraDB for MongoDB ...	1500
17.4.7. Release an ApsaraDB for MongoDB instance	1500
17.4.8. Back up an ApsaraDB for MongoDB instance	1501
17.4.8.1. Configure automatic backup for an ApsaraDB for...	1501
17.4.8.2. Manually back up an ApsaraDB for MongoDB in...	1501
17.4.9. Monitoring	1502
17.5. Database connection	1504
17.5.1. Use the mongo shell to connect to a replica set ins...	1504
17.5.2. Use DMS to log on to a replica set instance of Aps...	1505
17.5.3. Overview of replica set instance connections	1506
17.6. Security and audit	1507
17.6.1. Configure a whitelist for an ApsaraDB for MongoDB...	1507
17.6.2. Add or delete a whitelist	1508
17.6.3. Audit logs	1509
17.6.4. Configure SSL encryption	1510
17.6.5. Configure TDE	1511
17.6.6. Use the mongo shell to connect to an ApsaraDB fo...	1513
17.7. CloudDBA	1514
17.7.1. Authorize DAS to manage ApsaraDB for MongoDB in...	1514
17.7.2. Performance trends	1514
17.7.3. Real-time performance	1514
17.7.4. Instance sessions	1516
17.7.5. Capacity analysis	1517
17.7.6. Slow query logs	1519
18. ApsaraDB for OceanBase	1521
18.1. What is ApsaraDB for OceanBase?	1521
18.2. Quick start	1521

18.2.1. Overview	1521
18.2.2. Log on to the Apsara Stack Operations console for...	1522
18.2.3. Log on to the ApsaraDB for OceanBase console	1523
18.2.4. Add ApsaraDB for OceanBase RPM packages	1524
18.2.5. Add OBProxy RPM packages	1524
18.2.6. Install the OBProxy	1524
18.2.7. Create clusters	1525
18.2.8. Create instances	1525
18.3. Clusters	1526
18.3.1. Overview	1527
18.3.2. Scale out clusters online	1527
18.3.3. Restart clusters	1527
18.3.4. Delete clusters	1527
18.3.5. Upgrade clusters	1527
18.3.6. View the monitoring information about clusters	1528
18.3.7. View the real-time monitoring information about cl...	1528
18.3.8. View performance metrics	1528
18.4. Instances	1529
18.4.1. Overview	1529
18.4.2. Change instance passwords	1529
18.4.3. View instance details	1529
18.4.4. Change instance specifications	1530
18.4.5. Delete instances	1530
18.4.6. View the performance metrics of instances	1530
18.5. SQL syntax reference	1530
18.5.1. Introduction to ApsaraDB for OceanBase SQL	1530
18.5.2. Overview of ApsaraDB for OceanBase SQL	1531
18.5.2.1. Identifiers	1531

18.5.2.2. Supported SQL statements	1533
18.5.2.3. SQL limits	1533
18.5.3. Partitions	1534
18.5.3.1. Overview	1534
18.5.3.2. Range partitioning	1535
18.5.3.3. Hash partitioning	1539
18.5.3.4. Key partitioning	1540
18.5.3.5. Subpartitioning	1540
18.5.3.6. Handle NULL values	1544
18.5.4. Data types	1545
18.5.4.1. Overview	1545
18.5.4.2. Numeric types	1546
18.5.4.3. String types	1551
18.5.4.4. Date and time data types	1552
18.5.5. Character sets	1557
18.5.6. Auto-increment fields	1558
18.5.6.1. Overview	1558
18.5.6.2. System variables for auto-increment columns	1560
18.5.6.3. Change start values of auto-increment columns	1563
18.5.6.4. LAST_INSERT_ID() function	1564
18.5.6.5. Limits	1565
18.5.7. Functions	1565
18.5.7.1. Overview	1565
18.5.7.2. Date and time functions	1567
18.5.7.3. String functions	1584
18.5.7.4. Type conversion functions	1597
18.5.7.5. Aggregate functions	1598
18.5.7.6. Mathematical functions	1603

18.5.7.7. Comparison functions	1610
18.5.7.8. Flow control functions	1612
18.5.7.9. Information functions	1614
18.5.7.10. Other functions	1616
18.5.7.11. Full-text search functions	1618
18.5.8. Operators and precedences	1619
18.5.8.1. Overview	1619
18.5.8.2. Logical operators	1619
18.5.8.3. Arithmetic operators	1621
18.5.8.4. Comparison operators	1622
18.5.8.5. Vector comparison operators	1626
18.5.8.6. Bitwise operators	1628
18.5.8.7. Operator precedences	1629
18.5.9. Escape characters	1630
18.5.10. DDL statements	1631
18.5.10.1. Overview	1631
18.5.10.2. CREATE DATABASE	1631
18.5.10.3. ALTER_DATABASE	1632
18.5.10.4. DROP DATABASE	1633
18.5.10.5. CREATE TABLE	1633
18.5.10.6. ALTER TABLE	1638
18.5.10.7. DROP TABLE	1644
18.5.10.8. CREATE INDEX	1644
18.5.10.9. DROP INDEX	1647
18.5.10.10. CREATE VIEW	1648
18.5.10.11. DROP VIEW	1649
18.5.10.12. ALTER VIEW	1649
18.5.10.13. TRUNCATE TABLE	1649

18.5.10.14. RENAME TABLE	1650
18.5.10.15. CREATE SYNONYM	1650
18.5.10.16. DROP SYNONYM	1651
18.5.11. DML statements	1651
18.5.11.1. Overview	1651
18.5.11.2. INSERT	1652
18.5.11.3. REPLACE	1658
18.5.11.4. UPDATE	1660
18.5.11.5. DELETE	1663
18.5.11.6. SELECT	1664
18.5.12. Transaction language	1669
18.5.13. Database management language	1670
18.5.13.1. CREATE RESOURCE UNIT	1670
18.5.13.2. ALTER RESOURCE UNIT	1671
18.5.13.3. DROP RESOURCE UNIT	1672
18.5.13.4. CREATE RESOURCE POOL	1672
18.5.13.5. ALTER RESOURCE POOL	1672
18.5.13.6. DROP RESOURCE POOL	1673
18.5.13.7. CREATE TENANT	1673
18.5.13.8. ALTER TENANT	1674
18.5.13.9. Lock or unlock a tenant	1675
18.5.13.10. DROP TENANT	1675
18.5.13.11. CREATE TABLEGROUP	1676
18.5.13.12. DROP TABLEGROUP	1676
18.5.13.13. ALTER TABLEGROUP	1676
18.5.13.14. Users and permissions	1677
18.5.13.15. Modify system variables	1683
18.5.13.16. Modify user variables	1685

18.5.13.17. ALTER SYSTEM database management stateme...	1686
18.5.13.17.1. Overview	1686
18.5.13.17.2. System-level management statements	1686
18.5.13.17.3. Tenant-level management statements	1694
18.5.14. READ ONLY	1694
18.5.14.1. Overview	1694
18.5.14.2. Tenant-level READ ONLY property	1695
18.5.14.3. Database-level READ ONLY property	1696
18.5.14.4. Table-level READ ONLY property	1696
18.5.15. Other SQL statements	1697
18.5.15.1. SHOW statements	1697
18.5.15.2. KILL statement	1703
18.5.15.3. DESCRIBE statement	1704
18.5.15.4. USE statement	1704
18.5.15.5. Hints	1705
18.5.15.6. HELP statements	1711
18.5.16. OUTLINE	1713
18.5.16.1. Overview	1713
18.5.16.2. CREATE OUTLINE	1713
18.5.16.3. ALTER OUTLINE	1715
18.5.16.4. DROP OUTLINE	1716
18.5.16.5. Others	1716
18.5.16.5.1. Considerations	1716
18.5.16.5.2. Related system tables	1717
18.5.17. Hierarchical queries	1718
18.5.18. Materialized views	1719
18.5.19. SQL modes	1721
18.5.20. System views	1723

18.5.20.1. Overview	1723
18.5.20.2. v\$statname	1723
18.5.20.3. v\$event_name	1724
18.5.20.4. v\$session_event	1725
18.5.20.5. v\$session_wait	1726
18.5.20.6. v\$session_wait_history	1727
18.5.20.7. v\$sesstat	1727
18.5.20.8. v\$sysstat	1728
18.5.20.9. v\$system_event	1728
18.5.20.10. v\$memory	1729
18.5.20.11. v\$memstore	1730
18.5.20.12. v\$memstore_info	1730
18.5.20.13. v\$plan_cache_stat	1731
18.5.20.14. v\$plan_cache_plan_stat	1732
18.5.20.15. v\$plan_cache_plan_explain	1733
18.5.20.16. v\$sql_audit	1734
18.5.20.17. v\$latch	1737
18.5.20.18. v\$obrpc_outgoing	1738
18.5.20.19. v\$obrpc_incoming	1739
18.5.20.20. v\$sql	1740
18.5.20.21. v\$sql_monitor	1742
18.5.20.22. v\$sql_plan_monitor	1743
18.5.20.23. gv\$plan_cache_stat	1744
18.5.20.24. gv\$plan_cache_plan_stat	1745
18.5.20.25. gv\$session_event	1748
18.5.20.26. gv\$session_wait	1749
18.5.20.27. gv\$session_wait_history	1751
18.5.20.28. gv\$system_event	1752

18.5.20.29. gv\$sesstat	1753
18.5.20.30. gv\$sysstat	1754
18.5.20.31. gv\$sql_audit	1755
18.5.20.32. gv\$latch	1758
18.5.20.33. gv\$memory	1760
18.5.20.34. gv\$memstore	1761
18.5.20.35. gv\$memstore_info	1762
18.5.20.36. gv\$plan_cache_plan_explain	1763
18.5.20.37. gv\$obrpc_outgoing	1764
18.5.20.38. gv\$obrpc_incoming	1765
18.5.20.39. gv\$sql	1766
18.5.20.40. gv\$sql_monitor	1771
18.5.20.41. gv\$sql_plan_monitor	1772
18.5.21. Information Schema	1774
18.5.21.1. Overview	1774
18.5.21.2. INFORMATION_SCHEMA tables	1775
18.5.21.3. INFORMATION_SCHEMA.SCHEMATA table	1776
18.5.21.4. INFORMATION_SCHEMA.TABLES table	1776
18.5.21.5. INFORMATION_SCHEMA.COLUMNS table	1778
18.5.21.6. INFORMATION_SCHEMA.STATISTICS table	1779
18.5.21.7. INFORMATION_SCHEMA.USER_PRIVILEGES table	1780
18.5.21.8. INFORMATION_SCHEMA.SCHEMA_PRIVILEGES table	1781
18.5.21.9. INFORMATION_SCHEMA.TABLE_PRIVILEGES table	1781
18.5.21.10. INFORMATION_SCHEMA.CHARACTER_SETS table	1782
18.5.21.11. INFORMATION_SCHEMA.COLLATIONS table	1782
18.5.21.12. INFORMATION_SCHEMA.COLLATION_CHARACTER_...	1783
18.5.21.13. INFORMATION_SCHEMA.TABLE_CONSTRAINTS tab...	1783
18.5.21.14. INFORMATION_SCHEMA.REFERENTIAL_CONSTRAIN...	1784

18.5.21.15. INFORMATION_SCHEMA.KEY_COLUMN_USAGE tab...	1784
18.5.21.16. INFORMATION_SCHEMA.ROUTINES table	1786
18.5.21.17. INFORMATION_SCHEMA.VIEWS table	1788
18.5.21.18. INFORMATION_SCHEMA.TRIGGERS table	1789
18.5.21.19. INFORMATION_SCHEMA.TABLESPACE table	1791
18.5.21.20. INFORMATION_SCHEMA.PARTITIONS table	1791
18.5.21.21. INFORMATION_SCHEMA.EVENTS table	1792
18.5.21.22. INFORMATION_SCHEMA.FILES table	1794
18.5.21.23. INFORMATION_SCHEMA.GLOBAL_STATUS table	1795
18.5.21.24. INFORMATION_SCHEMA.GLOBAL_VARIABLES table	1796
18.5.21.25. INFORMATION_SCHEMA.PROCESSLIST table	1796
18.5.21.26. INFORMATION_SCHEMA.SESSION_STATUS table	1796
18.5.21.27. INFORMATION_SCHEMA.SESSION_VARIABLES table	1796
18.5.21.28. INFORMATION_SCHEMA.PROFILING table	1797
18.5.21.29. INFORMATION_SCHEMA.PARAMETERS table	1797
18.5.21.30. INFORMATION_SCHEMA.OPTIMIZER_TRACE table	1798
18.5.21.31. INFORMATION_SCHEMA.ENGINES table	1799
18.5.21.32. INFORMATION_SCHEMA.PLUGINS table	1799
18.5.22. MySQL dictionary tables	1800
18.5.23. Error codes	1800
18.5.23.1. ApsaraDB for OceanBase error codes	1800
18.5.24. Logs	1820
19.Data Transmission Service (DTS)	1823
19.1. What is DTS?	1823
19.2. Log on to the DTS console	1823
19.3. Data migration	1824
19.3.1. Database and migration types	1824
19.3.2. Create a data migration instance	1826

19.3.3. Configure data migration tasks	1826
19.3.3.1. Migrate data between ApsaraDB RDS for Postgre...-----	1826
19.3.3.2. Migrate incremental data from a user-created S...-----	1830
19.3.3.3. Migrate data from a user-created MySQL databa...-----	1836
19.3.3.4. Migrate data from a user-created MySQL databa...-----	1840
19.3.3.5. Migrate data from an ApsaraDB RDS for MySQL ...-----	1844
19.3.3.6. Migrate data between user-created Oracle data... -----	1846
19.3.3.7. Migrate data from a user-created Oracle databa...-----	1850
19.3.3.8. Migrate data from a user-created Oracle databa...-----	1853
19.3.3.9. Migrate data from a user-created Oracle databa...-----	1856
19.3.4. Manage data migration tasks	1859
19.3.4.1. Object name mapping	1859
19.3.4.2. Specify an SQL condition to filter data	1861
19.3.4.3. Troubleshoot a failed data migration task	1863
19.3.5. Precheck items	1864
19.3.5.1. Source database connectivity	1864
19.3.5.2. Check the destination database connectivity	1865
19.3.5.3. Binary logging configurations of the source dat... -----	1866
19.3.5.4. Integrity of the FOREIGN KEY constraints	1866
19.3.5.5. Existence of FEDERATED tables	1867
19.3.5.6. Permissions	1867
19.3.5.7. Object name conflict	1867
19.3.5.8. Schema existence	1867
19.3.5.9. Value of server_id in the source database	1868
19.3.5.10. Source database version	1868
19.3.6. Data type mappings between heterogeneous datab...-----	1868
19.4. Data synchronization	1872
19.4.1. Database types, initial synchronization types, and s...-----	1872

19.4.2. Create a data synchronization instance	1873
19.4.3. Synchronization topologies	1874
19.4.4. Configure data synchronization tasks	1875
19.4.4.1. Configure data synchronization between Apsara...	1876
19.4.4.2. Synchronize data from an ApsaraDB RDS for My...	1880
19.4.4.3. Synchronize data from an ApsaraDB RDS for My...	1885
19.4.4.4. Synchronize data from an ApsaraDB RDS for My...	1891
19.4.4.5. Synchronize data between Cloud Native Distribu...	1894
19.4.4.6. Synchronize data from a Cloud Native Distribute...	1898
19.4.4.7. Synchronize data from a Cloud Native Distribute...	1902
19.4.4.8. Configure two-way data synchronization betwee...	1904
19.4.4.8.1. Overview	1904
19.4.4.8.2. Supported synchronization statements	1904
19.4.4.8.3. Detect and resolve conflicts	1904
19.4.4.8.4. Synchronization restrictions	1905
19.4.4.8.5. Configure two-way data synchronization bet...	1906
19.4.5. Manage data synchronization instances	1908
19.4.5.1. Specify the name of an object in the destination...	1908
19.4.5.2. Check the synchronization performance	1911
19.4.5.3. Add objects to a data synchronization task	1911
19.4.5.4. Remove objects from a data synchronization ta...	1912
19.4.5.5. Troubleshoot precheck failures	1913
19.5. Change tracking	1916
19.5.1. Overview	1916
19.5.2. Create a change tracking instance	1917
19.5.3. Configure change tracking tasks	1917
19.5.3.1. Track data changes from a user-created MySQL ...	1917
19.5.3.2. Track data changes from a PolarDB-X instance	1921

19.5.3.3. Track data changes from a user-created Oracle ...	1923
19.5.4. Manage change tracking tasks	1926
19.5.4.1. Modify the consumption checkpoint	1926
19.5.4.2. Modify the objects for change tracking	1927
19.5.4.3. Create a consumer group	1928
19.5.4.4. Manage consumer groups	1928
19.5.5. Use the SDK to consume tracked data	1929
19.5.5.1. Methods provided by SDK	1929
19.5.5.2. Quick start	1933
19.5.5.3. Parse tracked SQL statements	1935
19.5.5.4. Run the SDK demo code	1939
19.5.6. Use a Kafka client to consume tracked data	1940
20.Data Management (DMS)	1945
20.1. What is Data Management?	1945
20.2. Log on to an ApsaraDB for RDS instance by using DMS	1946
20.3. SQL operations	1946
20.3.1. Use the command window	1946
20.3.2. Use the SQL window	1948
20.3.2.1. Open an empty SQL window	1948
20.3.2.2. Restore a saved SQL window	1954
20.3.2.3. Manage common SQL statements	1955
20.3.2.4. Use the SQL template	1956
20.3.3. Table operations (based on the Table directory tre...	1956
20.3.3.1. Open a table-based SQL window	1956
20.3.3.2. Edit table data	1956
20.4. Database development	1957
20.4.1. Overview	1957
20.4.2. Table	1957

20.4.2.1. Create a table	1957
20.4.2.2. Edit a table	1958
20.4.2.3. Delete a table	1958
20.4.2.4. Duplicate a table	1959
20.4.2.5. Generate SQL statement templates	1959
20.4.2.6. Query table information	1959
20.4.2.7. Clear data	1960
20.4.2.8. Perform operations on multiple tables	1960
20.4.2.9. Maintain a table	1960
20.4.3. Manage indexes	1961
20.4.4. Manage foreign keys	1962
20.4.5. Create a partition	1962
20.4.6. Create a stored procedure	1962
20.4.7. Create a function	1963
20.4.8. Create a view	1964
20.4.9. Create a trigger	1964
20.4.10. Create an event	1966
20.5. Data processing	1967
20.5.1. Import data	1967
20.5.2. Export data	1968
20.5.2.1. Export a database	1968
20.5.2.2. Export an SQL result set	1968
20.6. Performance	1969
20.6.1. Lock wait	1969
20.6.1.1. View sessions in the lock wait state	1969
20.6.1.2. Terminate a session in the lock wait state	1969
20.7. Extended tools	1970
20.7.1. View statistics on table data volumes	1970

20.7.2. View ER diagrams	1970
21. Server Load Balancer (SLB)	1972
21.1. What is SLB?	1972
21.2. Log on to the SLB console	1973
21.3. Quick start	1973
21.3.1. Overview	1973
21.3.2. Before you begin	1974
21.3.3. Create an SLB instance	1976
21.3.4. Configure an SLB instance	1976
21.3.5. Release an SLB instance	1978
21.4. SLB instances	1978
21.4.1. SLB instance overview	1978
21.4.2. Create an SLB instance	1980
21.4.3. Start and stop an SLB instance	1981
21.4.4. Tags	1982
21.4.4.1. Tag overview	1982
21.4.4.2. Add tags	1982
21.4.4.3. Query SLB instances by tag	1983
21.4.4.4. Remove tags	1983
21.4.5. Release an SLB instance	1984
21.5. Listeners	1985
21.5.1. Listener overview	1985
21.5.2. Add a TCP listener	1985
21.5.3. Add a UDP listener	1988
21.5.4. Add an HTTP listener	1990
21.5.5. Add an HTTPS listener	1993
21.5.6. Enable access control	1997
21.5.7. Disable access control	1998

21.6. Backend servers	1998
21.6.1. Backend server overview	1998
21.6.2. Default server groups	1999
21.6.2.1. Add ECS instances to the default server group	1999
21.6.2.2. Add IDC servers to the default server group	2000
21.6.2.3. Change the weight of a backend server	2002
21.6.2.4. Remove a backend server	2002
21.6.3. VServer groups	2002
21.6.3.1. Add ECS instances to a VServer group	2002
21.6.3.2. Add IDC servers to a VServer group	2003
21.6.3.3. Modify a VServer group	2004
21.6.3.4. Delete a VServer group	2005
21.6.4. Active/standby server groups	2005
21.6.4.1. Add ECS instances to a primary/secondary serve...	2005
21.6.4.2. Add IDC servers to a primary/secondary server ...	2008
21.6.4.3. Delete a primary/secondary server group	2010
21.7. Health check	2010
21.7.1. Health check overview	2011
21.7.2. Configure health check	2018
21.7.3. Disable the health check feature	2020
21.8. Certificate management	2020
21.8.1. Certificate overview	2020
21.8.2. Certificate requirements	2021
21.8.3. Upload a certificate	2023
21.8.4. Generate a CA certificate	2024
21.8.5. Convert the certificate format	2027
21.8.6. Replace a certificate	2028
22.Virtual Private Cloud (VPC)	2029

22.1. What is a VPC?	2029
22.2. Log on to the VPC console	2030
22.3. Quick start	2030
22.3.1. Plan and design a VPC	2030
22.3.2. Create an IPv4 VPC	2033
22.3.3. Create an IPv6 VPC	2036
22.4. VPCs and VSwitches	2040
22.4.1. Overview	2040
22.4.2. VPC management	2042
22.4.2.1. Create a VPC	2042
22.4.2.2. Modify the name and description of a VPC	2044
22.4.2.3. Delete a VPC	2044
22.4.3. VSwitch management	2044
22.4.3.1. Create a VSwitch	2044
22.4.3.2. Create cloud resources in a VSwitch	2046
22.4.3.3. Modify the name and description of a VSwitch	2046
22.4.3.4. Delete a VSwitch	2046
22.5. Route tables	2047
22.5.1. Overview	2047
22.5.2. Add a custom route entry	2051
22.5.3. Export route entries	2052
22.5.4. Modify a route table	2053
22.5.5. Delete a custom route entry	2053
22.6. HAVIPs	2053
22.6.1. Overview	2053
22.6.2. Create an HAVIP	2055
22.6.3. Associate an HAVIP with an ECS instance	2056
22.6.4. Associate an HAVIP with an EIP	2057

22.6.5. Disassociate an HAVIP from an ECS instance	2057
22.6.6. Disassociate an EIP from an HAVIP	2058
22.6.7. Delete an HAVIP	2058
23. IPv6 Gateway	2059
23.1. What is an IPv6 Gateway?	2059
23.2. Log on to the IPv6 Gateway console	2060
23.3. Quick start	2061
23.3.1. Create an IPv6 VPC	2061
23.4. Enable IPv6 for VPCs	2065
23.4.1. Create an IPv4 and IPv6 dual-stack VPC	2065
23.4.2. Enable an IPv6 CIDR block for a VPC network	2066
23.5. Enable IPv6 for VSwitches	2066
23.5.1. Create an IPv4 and IPv6 dual-stack VSwitch	2066
23.5.2. Enable IPv6 for a VSwitch	2068
23.6. Manage IPv6 Gateways	2068
23.6.1. Editions of IPv6 gateways	2068
23.6.2. Create an IPv6 gateway	2069
23.6.3. Modify an IPv6 gateway	2070
23.6.4. Delete an IPv6 gateway	2070
23.7. Manage IPv6 Internet bandwidth	2070
23.7.1. Enable Internet connectivity for an IPv6 address	2070
23.7.2. Modify the maximum bandwidth of an IPv6 address	2071
23.7.3. Disable Internet connectivity for an IPv6 address	2071
23.8. Manage egress-only rules	2072
23.8.1. Create an egress-only rule	2072
23.8.2. Delete an egress-only rule	2072
24. NAT Gateway	2073
24.1. What is NAT Gateway?	2073

24.2. Log on to the NAT Gateway console	2073
24.3. Quick Start	2074
24.3.1. Overview	2074
24.3.2. Create a NAT gateway	2075
24.3.3. Associate an EIP with the a NAT gateway	2076
24.3.4. Create a DNAT entry	2076
24.3.5. Create an SNAT entry	2077
24.4. Manage a NAT gateway	2079
24.4.1. Sizes of NAT gateways	2079
24.4.2. Create a NAT gateway	2080
24.4.3. Modify a NAT gateway	2081
24.4.4. Delete a NAT gateway	2081
24.5. Manage EIPs	2081
24.5.1. Associate an EIP with a NAT gateway	2081
24.5.2. Disassociate an EIP from a NAT gateway	2082
24.6. Manage a DNAT table	2082
24.6.1. DNAT table overview	2082
24.6.2. Create a DNAT entry	2083
24.6.3. Modify a DNAT entry	2084
24.6.4. Delete a DNAT entry	2085
24.7. Manage an SNAT table	2085
24.7.1. SNAT table overview	2085
24.7.2. Create an SNAT entry	2085
24.7.3. Modify an SNAT entry	2087
24.7.4. Delete a SNAT entry	2087
24.8. NAT service plan	2088
24.8.1. Create a NAT service plan	2088
24.8.2. Modify the bandwidth of a NAT service plan	2088

24.8.3. Add an IP address	2089
24.8.4. Release an IP address	2089
24.8.5. Delete a NAT service plan	2089
24.9. Anti-DDoS Basic	2090
25.VPN Gateway	2091
25.1. What is VPN Gateway?	2091
25.2. Log on to the VPN Gateway console	2091
25.3. Get started with IPsec-VPN	2092
25.3.1. Connect on-premises data centers to VPC networks	2092
25.4. Get started with SSL-VPN	2095
25.4.1. Initiate a connection from a Linux client	2095
25.4.2. Initiate a connection from a Windows client	2098
25.4.3. Initiate a connection from a macOS client	2100
25.5. Manage a VPN Gateway	2103
25.5.1. Create a VPN gateway	2103
25.5.2. Modify a VPN gateway	2104
25.5.3. Configure routes of a VPN Gateway	2104
25.5.3.1. VPN Gateway route overview	2104
25.5.3.2. Add a policy-based route entry	2105
25.5.3.3. Add a destination-based route entry	2105
25.5.4. Delete a VPN gateway	2106
25.6. Manage a customer gateway	2106
25.6.1. Create a customer gateway	2106
25.6.2. Modify a customer gateway	2107
25.6.3. Delete a customer gateway	2108
25.7. Configure SSL-VPN	2108
25.7.1. Configuration overview	2108
25.7.2. Manage an SSL server	2109

25.7.2.1. Create an SSL Server	2109
25.7.2.2. Modify an SSL server	2110
25.7.2.3. Configure a routing group	2110
25.7.2.4. Delete an SSL server	2112
25.7.3. Manage an SSL client certificate	2112
25.7.3.1. Create an SSL client certificate	2112
25.7.3.2. Download an SSL client certificate	2113
25.7.3.3. Delete an SSL client certificate	2113
25.8. Configure IPsec-VPN connections	2113
25.8.1. Configuration overview	2113
25.8.2. Manage an IPsec-VPN connection	2114
25.8.2.1. Create an IPsec-VPN connection	2114
25.8.2.2. Modify an IPsec-VPN connection	2116
25.8.2.3. Download the configuration file of an IPsec-VP...	2117
25.8.2.4. Configure a routing group	2117
25.8.2.5. View IPsec-VPN connection logs	2118
25.8.2.6. Delete an IPsec-VPN connection	2119
25.8.3. MTU notes	2119
26.Elastic IP Address	2120
26.1. What is an EIP?	2120
26.2. Log on to the EIP console	2120
26.3. Quick start	2121
26.3.1. Tutorial overview	2121
26.3.2. Create a EIP	2121
26.3.3. Associate an EIP with an ECS instance	2122
26.3.4. Disassociate an EIP from a cloud resource	2123
26.3.5. Release an EIP	2123
26.4. Manage EIPs	2123

26.4.1. Create a EIP	2123
26.4.2. Bind an EIP to a cloud instance	2124
26.4.2.1. Associate an EIP with an ECS instance	2124
26.4.2.2. Associate an EIP with an SLB instance	2125
26.4.2.3. Associate an EIP with an HAVIP	2125
26.4.2.4. Associate an EIP with a NAT gateway	2126
26.4.2.5. Bind an EIP to a secondary ENI	2126
26.4.2.5.1. Overview	2126
26.4.2.5.2. Associate an EIP with an ENI in the NAT mo...-----	2128
26.4.2.5.3. Associate an EIP with an ENI in the cut-thro...-----	2129
26.4.3. Upgrade a subscription EIP	2131
26.4.4. Disassociate an EIP from a cloud resource	2131
26.4.5. Release an EIP	2131
27.Express Connect	2133
27.1. What is Express Connect?	2133
27.2. Log on to the Express Connect console	2133
27.3. Connect two VPCs	2134
27.4. Delete a peering connection	2135
28.Apsara Stack Security	2136
28.1. What is Apsara Stack Security?	2136
28.2. Precautions	2136
28.3. Quick start	2137
28.3.1. User roles and permissions	2137
28.3.2. Log on to Apsara Stack Security Center	2137
28.4. Threat Detection Service	2138
28.4.1. Threat Detection Service overview	2138
28.4.2. Security overview	2138
28.4.2.1. View security overview information	2138

28.4.3. Security alerts	2140
28.4.3.1. View security alerts	2140
28.4.3.2. Manage quarantined files	2140
28.4.3.3. Configure security alerts	2141
28.4.4. Attack analysis	2142
28.4.5. Cloud service check	2144
28.4.5.1. Overview	2144
28.4.5.2. Run cloud service checks	2147
28.4.5.3. View and manage check results of Alibaba Clou... ..	2148
28.4.6. Application whitelist	2149
28.4.7. Assets	2152
28.4.7.1. View the security status of a server	2152
28.4.7.2. View the security status of cloud services	2155
28.4.7.3. View the details of a single asset	2156
28.4.7.4. Enable and disable server protection	2160
28.4.7.5. Perform a one-click security check	2160
28.4.7.6. Manage asset groups	2161
28.4.7.7. Manage asset tags	2163
28.4.8. Create a security report	2165
28.5. Network Traffic Monitoring System	2167
28.5.1. View traffic trends	2167
28.5.2. View traffic at the Internet border	2167
28.5.3. View traffic at the internal network border	2169
28.6. Server security	2169
28.6.1. Server security overview	2169
28.6.2. Server fingerprints	2170
28.6.2.1. Manage listener ports	2170
28.6.2.2. Manage software versions	2170

28.6.2.3. Manage processes	2171
28.6.2.4. Manage account information	2171
28.6.2.5. Manage scheduled tasks	2172
28.6.2.6. Set the server fingerprint collection frequency	2172
28.6.3. Threat protection	2172
28.6.3.1. Vulnerability management	2172
28.6.3.1.1. Manage Linux software vulnerabilities	2172
28.6.3.1.2. Manage Windows vulnerabilities	2173
28.6.3.1.3. Manage Web CMS vulnerabilities	2174
28.6.3.1.4. Manage emergency vulnerabilities	2175
28.6.3.1.5. Configure vulnerability management policies	2176
28.6.3.2. Baseline check	2177
28.6.3.2.1. Baseline check overview	2177
28.6.3.2.2. Configure baseline check policies	2179
28.6.3.2.3. View baseline check results and manage fai...	2180
28.6.4. Intrusion detection	2182
28.6.4.1. Intrusion events	2182
28.6.4.1.1. Intrusion event types	2183
28.6.4.1.2. View and handle intrusion events	2184
28.6.4.1.3. View exceptions related to an alert	2185
28.6.4.1.4. Use the file quarantine function	2186
28.6.4.1.5. Configure security alerts	2186
28.6.4.1.6. Virus removal	2187
28.6.4.2. Website tamper-proofing	2188
28.6.4.2.1. Overview	2189
28.6.4.2.2. Configure tamper protection	2190
28.6.4.2.3. View the protection status	2193
28.6.4.3. Configure the Virus Removal feature	2193

28.6.5. Log retrieval	2194
28.6.5.1. Log retrieval overview	2194
28.6.5.2. Log retrieval	2195
28.6.5.3. Supported log sources and fields	2196
28.6.5.4. Logical operators	2199
28.6.6. Settings	2200
28.6.6.1. Install the Server Guard agent	2200
28.6.6.2. Manage protection modes	2201
28.7. Application security	2201
28.7.1. Quick start	2201
28.7.2. Detection overview	2202
28.7.2.1. View protection overview	2202
28.7.2.2. View web service access information	2203
28.7.3. Protection logs	2204
28.7.3.1. View attack detection logs	2204
28.7.3.2. View HTTP flood protection logs	2204
28.7.3.3. View system operation logs	2204
28.7.3.4. View access logs	2205
28.7.4. Protection configuration	2205
28.7.4.1. Configure protection policies	2205
28.7.4.2. Create a custom rule	2206
28.7.4.3. Configure an HTTP flood detection rule	2208
28.7.4.4. Configure an HTTP flood protection whitelist	2210
28.7.4.5. Manage SSL certificates	2211
28.7.4.6. Add Internet websites for protection	2212
28.7.4.7. Add VPC websites for protection	2216
28.7.4.8. Verify the access configuration of a domain on ...	2219
28.7.4.9. Modify DNS resolution settings	2220

28.7.5. System management	2220
28.7.5.1. View the payload status of nodes	2220
28.7.5.2. View the network status of nodes	2221
28.7.5.3. View the disk status of nodes	2223
28.7.5.4. Configure alert service	2223
28.7.5.5. Configure alert thresholds	2224
28.8. Security Operations Center (SOC)	2225
28.8.1. View the dashboard	2225
28.8.2. Security Monitoring	2225
28.8.2.1. View security monitoring data of tenants	2225
28.8.2.2. View security monitoring data of platforms	2227
28.8.3. Asset Management	2229
28.8.3.1. View tenant assets	2229
28.8.3.2. View platform assets	2229
28.8.4. Log Analysis	2230
28.8.4.1. View log analysis results	2230
28.8.4.2. Security Audit	2230
28.8.4.2.1. Overview	2230
28.8.4.2.2. View security audit overview	2230
28.8.4.2.3. Query audit events	2231
28.8.4.2.4. View raw logs	2232
28.8.4.2.5. Manage log sources	2233
28.8.4.2.6. Policy settings	2233
28.8.4.2.6.1. Manage audit rules	2233
28.8.4.2.6.2. Configure alert recipients	2235
28.8.4.2.6.3. Manage archives of events and logs	2236
28.8.4.2.6.4. Manage export tasks	2236
28.8.4.2.6.5. Modify system settings	2236

28.8.5. Rules	2237
28.8.5.1. Create traffic monitoring IPS rules	2237
28.8.5.2. Manage IPS rules of Cloud Firewall	2238
28.8.5.3. Create traffic monitoring IDS rules	2238
28.8.5.4. Manage traffic monitoring IDS rules	2239
28.8.5.5. Customize DDoS traffic scrubbing policies and tr...	2240
28.8.5.6. View Server Guard rules	2240
28.8.6. Create a report task	2241
28.8.7. System Configurations	2242
28.8.7.1. Alert settings	2242
28.8.7.1.1. Set alert recipients	2242
28.8.7.1.2. Set alert notifications	2243
28.8.7.2. Updates	2244
28.8.7.2.1. Overview of the system updates feature	2244
28.8.7.2.2. Automatically download an update package	2244
28.8.7.2.3. Manually import an update package and up...	2245
28.8.7.2.4. Roll back a rule library	2245
28.8.7.2.5. View update history of a rule library	2246
28.8.7.3. Global Settings	2246
28.8.7.3.1. Set CIDR blocks for traffic monitoring	2246
28.8.7.3.1.1. Add a CIDR block for traffic monitoring	2246
28.8.7.3.1.2. Manage CIDR blocks for traffic monitoring	2247
28.8.7.3.2. Region settings	2247
28.8.7.3.2.1. Add a CIDR block for a region	2247
28.8.7.3.2.2. Manage CIDR blocks for a region	2248
28.8.7.3.3. Configure whitelists	2248
28.8.7.3.4. Configure request blocking policies	2249
28.8.7.3.5. Block IP Addresses	2250

28.8.7.3.6. Configure custom IP addresses and locations	2251
28.8.7.3.6.1. Add custom IP addresses and locations	2251
28.8.7.3.6.2. Manage custom IP addresses and locatio...	2251
28.8.7.4. System Monitoring	2251
28.8.7.4.1. Configure CIDR blocks for traffic redirection i...	2251
28.8.7.5. Remote operations	2252
28.8.7.5.1. Enable Remote O&M	2252
28.8.7.6. Account management	2253
28.8.7.6.1. View and modify Apsara Stack accounts	2253
28.8.7.6.2. Add a public cloud account	2254
28.9. Optional security products	2254
28.9.1. Anti-DDoS settings	2254
28.9.1.1. Overview	2254
28.9.1.2. View and configure anti-DDoS policies	2255
28.9.1.3. View DDoS events	2256
28.9.2. Cloud Firewall	2257
28.9.2.1. Access control	2257
28.9.2.1.1. Configure the Internet firewall switch policy	2257
28.9.2.1.2. Create a VPC firewall	2258
28.9.2.1.3. Create an IDC-VPC firewall	2259
28.9.2.1.4. Manage address books	2262
28.9.2.1.5. Configure an Internet firewall	2263
28.9.2.1.6. Create a policy group	2265
28.9.2.1.7. Manage an internal firewall	2266
28.9.2.1.8. Manage a VPC firewall	2268
28.9.2.1.9. Manage an IDC-VPC firewall	2271
28.9.2.2. Intrusion prevention	2273
28.9.2.2.1. Configure intrusion prevention policies	2273

28.9.2.2.2. View event logs	2275
28.9.2.3. Log analysis	2275
28.9.2.3.1. View traffic logs	2275
28.9.3. Sensitive Data Discovery and Protection	2276
28.9.3.1. Grant access permissions	2276
28.9.3.2. Overview	2277
28.9.3.3. Detect sensitive data	2278
28.9.3.3.1. Sensitive data overview	2278
28.9.3.3.2. View statistics on sensitive data in MaxComp... ..	2278
28.9.3.3.3. View statistics on sensitive data in Tablestore	2280
28.9.3.3.4. View statistics on sensitive data in OSS	2282
28.9.3.3.5. View statistics on sensitive data in AnalyticDB	2283
28.9.3.3.6. View statistics on sensitive data in ApsaraDB... ..	2284
28.9.3.4. Check data permissions	2285
28.9.3.4.1. View permission statistics	2285
28.9.3.4.2. View permissions of an account	2286
28.9.3.5. Monitor data flows	2287
28.9.3.5.1. View data flows in DataHub	2287
28.9.3.5.2. View data flows in Data Integration	2288
28.9.3.6. Sensitive data masking	2289
28.9.3.6.1. Add a static desensitization task	2289
28.9.3.7. Abnormal activity detection	2291
28.9.3.7.1. Add a custom rule for abnormal activities	2291
28.9.3.7.2. Process abnormal activities	2292
28.9.3.8. Intelligent audit	2294
28.9.3.8.1. View and download audit reports	2294
28.9.3.8.2. View audit logs	2295
28.9.3.8.3. View raw logs	2295

28.9.3.8.4. Add an audit rule	2296
28.9.3.9. Security configuration	2296
28.9.3.9.1. Manage rules used to identify sensitive data	2296
28.9.3.9.2. Manage thresholds and rules used to detect... ..	2298
28.9.3.9.3. Configure an authorized asset	2299
28.9.3.9.4. Configure desensitization algorithms	2300
29.Key Management Service (KMS)	2305
29.1. Manage keys in the KMS console	2305
29.1.1. Log on to the KMS console	2305
29.1.2. Create a CMK	2305
29.1.3. View CMK details	2306
29.1.4. Enable a CMK	2306
29.1.5. Disable a CMK	2306
29.1.6. Schedule the deletion of a CMK	2306
29.1.7. Configure automatic rotation of a CMK	2307
29.2. Use a CLI to manage CMKs	2307
29.3. Use RAM for access control	2312
29.4. Use aliases	2316
29.5. CMK overview	2321
29.6. Use symmetric keys	2322
29.6.1. Overview of symmetric encryption	2322
29.6.2. EncryptionContext	2323
29.6.3. Import and delete key material	2324
29.7. Use asymmetric keys	2328
29.7.1. Overview of asymmetric keys	2328
29.7.2. Encrypt and decrypt data by using an asymmetric C... ..	2330
29.7.3. Generate and verify a digital signature by using an... ..	2332
29.8. Key rotation	2335

29.8.1. Overview	2335
29.8.2. Automatic key rotation	2336
29.8.3. Manual key rotation	2338
30. Log Service	2340
30.1. What is Log Service?	2340
30.2. Quick start	2340
30.2.1. Procedure	2340
30.2.2. Log on to the Log Service console	2341
30.2.3. Obtain an AccessKey pair	2342
30.2.4. Manage projects	2343
30.2.5. Manage Logstores	2345
30.2.6. Manage shards	2348
30.3. Data collection	2351
30.3.1. Collection by Logtail	2351
30.3.1.1. Overview	2351
30.3.1.1.1. Logtail overview	2351
30.3.1.1.2. Log collection process of Logtail	2354
30.3.1.1.3. Logtail configuration files and record files	2356
30.3.1.2. Installation	2364
30.3.1.2.1. Install Logtail in Linux	2364
30.3.1.2.2. Install Logtail in Windows	2366
30.3.1.2.3. Set Logtail startup parameters	2368
30.3.1.3. Logtail machine group	2371
30.3.1.3.1. Overview	2371
30.3.1.3.2. Create a machine group based on a server I...	2372
30.3.1.3.3. Create a machine group based on a custom...	2373
30.3.1.3.4. View server groups	2376
30.3.1.3.5. Modify a server group	2377

30.3.1.3.6. View the status of a server group	2377
30.3.1.3.7. Delete a server group	2377
30.3.1.3.8. Manage server group configurations	2378
30.3.1.3.9. Manage a Logtail configuration	2378
30.3.1.3.10. Configure an account ID on a server	2379
30.3.1.4. Text logs	2381
30.3.1.4.1. Configure text log collection	2381
30.3.1.4.2. Collect logs by line	2385
30.3.1.4.3. Use regular expressions to collect logs	2388
30.3.1.4.4. Collect DSV formatted logs	2392
30.3.1.4.5. Collect JSON logs	2399
30.3.1.4.6. Collect NGINX logs	2403
30.3.1.4.7. Collect IIS logs	2407
30.3.1.4.8. Collect Apache logs	2413
30.3.1.4.9. Configure parsing scripts	2418
30.3.1.4.10. Configure the time format	2419
30.3.1.4.11. Import historical logs	2422
30.3.1.4.12. Generate a topic	2424
30.3.1.5. Custom plug-ins	2426
30.3.1.5.1. Collect MySQL binary logs	2426
30.3.1.5.2. Collect query results from a MySQL database	2437
30.3.1.5.3. Collect syslogs	2442
30.3.1.5.4. Configure data processing methods	2446
30.3.1.6. Collect container logs	2467
30.3.1.6.1. Collect standard Docker logs	2467
30.3.1.6.2. Collect Kubernetes logs	2471
30.3.1.6.3. Collect container text logs	2476
30.3.1.6.4. Collect stdout and stderr logs from containe...	2480

30.3.1.7. Limits	2491
30.3.2. Other collection methods	2493
30.3.2.1. WebTracking	2493
30.3.2.2. Collect logs over the syslog protocol	2498
30.3.2.3. Use SDKs to collect logs	2502
30.3.2.3.1. Producer Library	2502
30.3.2.3.2. Log4j Appender	2502
30.3.2.3.3. Logback Appender	2503
30.3.2.3.4. Golang Producer Library	2503
30.3.2.3.5. Python logging	2503
30.3.2.4. Common log formats	2507
30.3.2.4.1. Log4j logs	2507
30.3.2.4.2. Python logs	2509
30.3.2.4.3. Node.js logs	2514
30.3.2.4.4. WordPress logs	2515
30.3.2.4.5. Unity3D logs	2516
30.4. Query and analysis	2518
30.4.1. Overview	2518
30.4.2. Real-time analysis	2519
30.4.3. Enable the indexing feature and configure indexes...	2521
30.4.4. Query logs	2525
30.4.5. Export logs	2528
30.4.6. Index data type	2528
30.4.6.1. Overview	2528
30.4.6.2. Query text data	2531
30.4.6.3. Numeric type	2532
30.4.6.4. JSON indexes	2532
30.4.7. Query syntax and functions	2535

30.4.7.1. Search syntax	2535
30.4.7.2. LiveTail	2540
30.4.7.3. LogReduce	2544
30.4.7.4. Contextual query	2548
30.4.7.5. Saved search	2550
30.4.7.6. Quick analysis	2551
30.4.7.7. Other features	2554
30.4.8. Analysis grammar	2555
30.4.8.1. General aggregate functions	2555
30.4.8.2. Security check functions	2557
30.4.8.3. Map functions	2559
30.4.8.4. Approximate functions	2560
30.4.8.5. Mathematical statistics functions	2561
30.4.8.6. Mathematical calculation functions	2562
30.4.8.7. String functions	2564
30.4.8.8. Date and time functions	2566
30.4.8.9. URL functions	2570
30.4.8.10. Regular expression functions	2571
30.4.8.11. JSON functions	2572
30.4.8.12. Type conversion functions	2573
30.4.8.13. IP functions	2574
30.4.8.14. GROUP BY syntax	2577
30.4.8.15. Window functions	2578
30.4.8.16. HAVING syntax	2580
30.4.8.17. ORDER BY syntax	2580
30.4.8.18. LIMIT syntax	2581
30.4.8.19. Syntax for CASE statements and if() functions	2581
30.4.8.20. Nested subqueries	2583

30.4.8.21. Array functions	2583
30.4.8.22. Binary string functions	2585
30.4.8.23. Bitwise operations	2586
30.4.8.24. Interval-valued comparison and periodicity-valued...	2586
30.4.8.25. Comparison functions and operators	2589
30.4.8.26. Lambda functions	2591
30.4.8.27. Logical functions	2593
30.4.8.28. Field aliases	2594
30.4.8.29. JOIN operations between Logstores and Relatio...	2594
30.4.8.30. Geospatial functions	2597
30.4.8.31. Geography functions	2600
30.4.8.32. JOIN syntax	2601
30.4.8.33. UNNEST function	2602
30.4.9. Machine learning syntax and functions	2603
30.4.9.1. Overview	2603
30.4.9.2. Smooth functions	2605
30.4.9.3. Multi-period estimation functions	2609
30.4.9.4. Change point detection functions	2611
30.4.9.5. Maximum value detection function	2613
30.4.9.6. Prediction and anomaly detection functions	2614
30.4.9.7. Time series decomposition function	2621
30.4.9.8. Time series clustering functions	2621
30.4.9.9. Frequent pattern statistics function	2626
30.4.9.10. Differential pattern statistics function	2627
30.4.9.11. Root cause analysis function	2628
30.4.9.12. Correlation analysis functions	2631
30.4.9.13. Kernel density estimation function	2634
30.4.10. Advanced analysis	2635

30.4.10.1. Optimize queries	2635
30.4.10.2. Use cases	2637
30.4.10.3. Time field conversion examples	2638
30.4.11. Visual analysis	2639
30.4.11.1. Analysis graph	2639
30.4.11.1.1. Overview	2639
30.4.11.1.2. Table	2640
30.4.11.1.3. Line chart	2641
30.4.11.1.4. Column chart	2643
30.4.11.1.5. Bar chart	2644
30.4.11.1.6. Pie chart	2646
30.4.11.1.7. Area chart	2649
30.4.11.1.8. Individual value plot	2650
30.4.11.1.9. Progress bar	2655
30.4.11.1.10. Map	2657
30.4.11.1.11. Flow diagram	2660
30.4.11.1.12. Sankey diagram	2662
30.4.11.1.13. Word cloud	2663
30.4.11.1.14. Treemap chart	2664
30.4.11.2. Dashboard	2665
30.4.11.2.1. Overview	2665
30.4.11.2.2. Create and delete a dashboard	2666
30.4.11.2.3. Configure the display mode of a dashboard	2669
30.4.11.2.4. Edit mode	2671
30.4.11.2.5. Drill-down analysis	2673
30.4.11.2.6. Configure and use a filter on a dashboard ...	2678
30.4.11.2.7. Markdown chart	2682
30.5. Alerts	2685

30.5.1. Overview	2685
30.5.2. Configure an alarm	2686
30.5.2.1. Configure alerts	2686
30.5.2.2. Grant permissions on alerts to a RAM user	2688
30.5.2.3. Notification methods	2689
30.5.3. Modify and view an alarm	2692
30.5.3.1. Modify an alert	2692
30.5.3.2. View history alerts	2693
30.5.3.3. Manage an alert	2694
30.5.4. Relevant syntax and fields for reference	2695
30.5.4.1. Conditional expression syntax of an alert	2695
30.5.4.2. Fields in alert log entries	2698
30.6. Real-time consumption	2701
30.6.1. Overview	2701
30.6.2. Consume log data	2702
30.6.3. Consumption by consumer groups	2704
30.6.3.1. Use consumer groups to consume log data	2704
30.6.3.2. View the status of a consumer group	2711
30.6.4. Use LogHub Storm to consume log data	2713
30.6.5. Use Flume to consume log data	2717
30.6.6. Use open source Flink to consume log data	2720
30.6.7. Use Logstash to consume log data	2726
30.6.8. Use Spark Streaming to consume log data	2726
30.6.9. Use Realtime Compute to consume log data	2730
30.7. RAM	2733
30.7.1. Overview	2734
30.7.2. Create a RAM role	2734
30.7.3. Create a user	2734

30.7.4. Create a RAM user group	2735
30.7.5. Add a RAM user to a RAM user group	2736
30.7.6. Create a permission policy	2736
30.7.7. Grant permissions to a RAM role	2737
30.7.8. Use custom policies to grant RAM user the required... ..	2737
30.8. FAQ	2742
30.8.1. Log collection	2742
30.8.1.1. How do I troubleshoot Logtail collection errors?	2742
30.8.1.2. What can I do if Log Service does not receive h... ..	2743
30.8.1.3. How do I query the local log collection statuses... ..	2745
30.8.1.4. How do I test a regular expression?	2757
30.8.1.5. How do I optimize regular expressions?	2759
30.8.1.6. How do I use the full regex mode to collect log... ..	2759
30.8.1.7. How do I set the time format for logs?	2760
30.8.1.8. How do I configure non-printable characters in	2761
30.8.1.9. How do I troubleshoot errors during container l... ..	2762
30.8.2. Log search and analysis	2765
30.8.2.1. FAQ about log query	2765
30.8.2.2. What can I do if no log data is retrieved?	2766
30.8.2.3. What are the differences between log consump... ..	2767
30.8.2.4. How do I resolve common errors returned in lo... ..	2768
30.8.2.5. Why data queries are inaccurate?	2769
30.8.2.6. How do I configure indexes for historical log d... ..	2770
30.8.3. Alarm	2770
30.8.3.1. FAQ about alerts	2770
31. Apsara Stack DNS	2772
31.1. What is Apsara Stack DNS?	2772
31.2. User roles and permissions	2772

31.3. Log on to the Apsara Stack DNS console	2773
31.4. Internal DNS resolution management	2773
31.4.1. Global internal domain names	2773
31.4.1.1. Overview	2773
31.4.1.2. View an internal domain name	2773
31.4.1.3. Add a domain name	2773
31.4.1.4. Add a description for a domain name	2774
31.4.1.5. Delete a domain name	2774
31.4.1.6. Delete multiple domain names	2774
31.4.1.7. Configure DNS records	2774
31.4.1.8. View a resolution policy	2775
31.4.2. Global forwarding configurations	2775
31.4.2.1. Global forwarding domain names	2775
31.4.2.1.1. Overview	2775
31.4.2.1.2. View global forwarding domain names	2776
31.4.2.1.3. Add a domain name	2776
31.4.2.1.4. Add a description for a domain name	2776
31.4.2.1.5. Modify the forwarding configurations of a do...	2776
31.4.2.1.6. Delete a domain name	2777
31.4.2.1.7. Delete multiple domain names	2777
31.4.2.2. Global default forwarding configurations	2777
31.4.2.2.1. Enable default forwarding	2777
31.4.2.2.2. Modify default forwarding configurations	2777
31.4.2.2.3. Disable default forwarding	2778
31.4.3. Global recursive resolution	2778
31.4.3.1. Enable global recursive resolution	2778
31.4.3.2. Disable global recursive resolution	2778
31.5. PrivateZone (DNS Standard Edition only)	2779

31.5.1. Tenant internal domain name	2779
31.5.1.1. View a domain name	2779
31.5.1.2. Add a domain name	2779
31.5.1.3. Bind an organization to a VPC	2779
31.5.1.4. Unbind a domain name from a VPC	2779
31.5.1.5. Add a description for a domain name	2780
31.5.1.6. Delete a domain name	2780
31.5.1.7. Delete multiple domain names	2780
31.5.1.8. Configure DNS records	2780
31.5.1.9. View a resolution policy	2785
31.5.2. Tenant forwarding configurations	2785
31.5.2.1. Tenant forwarding domain names	2785
31.5.2.1.1. View a tenant forwarding domain name	2785
31.5.2.1.2. Add a tenant forwarding domain name	2786
31.5.2.1.3. Bind an organization to a VPC	2786
31.5.2.1.4. Unbind a domain name from a VPC	2787
31.5.2.1.5. Modify the forwarding configurations of a do...	2787
31.5.2.1.6. Add a description for a tenant forwarding d...	2787
31.5.2.1.7. Delete a tenant forwarding domain name	2787
31.5.2.1.8. Delete multiple tenant forwarding domain n...	2788
31.5.2.2. Tenant default forwarding configurations	2788
31.5.2.2.1. View default forwarding configurations	2788
31.5.2.2.2. Add a default forwarding configuration	2788
31.5.2.2.3. Bind an organization to a VPC	2789
31.5.2.2.4. Unbind a domain name from a VPC	2789
31.5.2.2.5. Modify a default forwarding configuration	2790
31.5.2.2.6. Add a default forwarding configuration	2790
31.5.2.2.7. Delete a default forwarding configuration	2790

31.5.2.2.8. Delete multiple default forwarding configura...	2790
31.6. Internal Global Traffic Manager (internal GTM Standard..	2791
31.6.1. Scheduling instance management	2791
31.6.1.1. Scheduling Instance	2791
31.6.1.1.1. Create a scheduling instance	2791
31.6.1.1.2. Modify a scheduling instance	2791
31.6.1.1.3. Configure a scheduling instance	2791
31.6.1.1.3.1. Create an access policy for a scheduling i...	2791
31.6.1.1.3.2. Modify the access policy of a scheduling i...	2791
31.6.1.1.3.3. Delete the access policy of a scheduling i...	2792
31.6.1.1.4. Delete a scheduling instance	2792
31.6.1.2. Address Pool	2792
31.6.1.2.1. Create an address pool	2792
31.6.1.2.2. Modify the configurations of an address pool	2792
31.6.1.2.3. Delete an address pool	2792
31.6.1.3. Scheduling Domain	2792
31.6.1.3.1. Create a scheduling domain	2792
31.6.1.3.2. Add a description for a scheduling domain	2793
31.6.1.3.3. Delete a scheduling domain	2793
31.6.2. Data synchronization management	2793
31.6.2.1. Data Synchronization Link	2793
31.6.2.1.1. Standalone node on the Data Synchronizatio...	2793
31.6.2.1.2. Master node on the Data Synchronization Li...	2793
31.6.2.1.3. Slave node on the Data Synchronization Link...	2793
31.6.2.2. Link Change Messages	2793

1.ASCM console

1.1. What is the ASCM console?

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

Overview

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

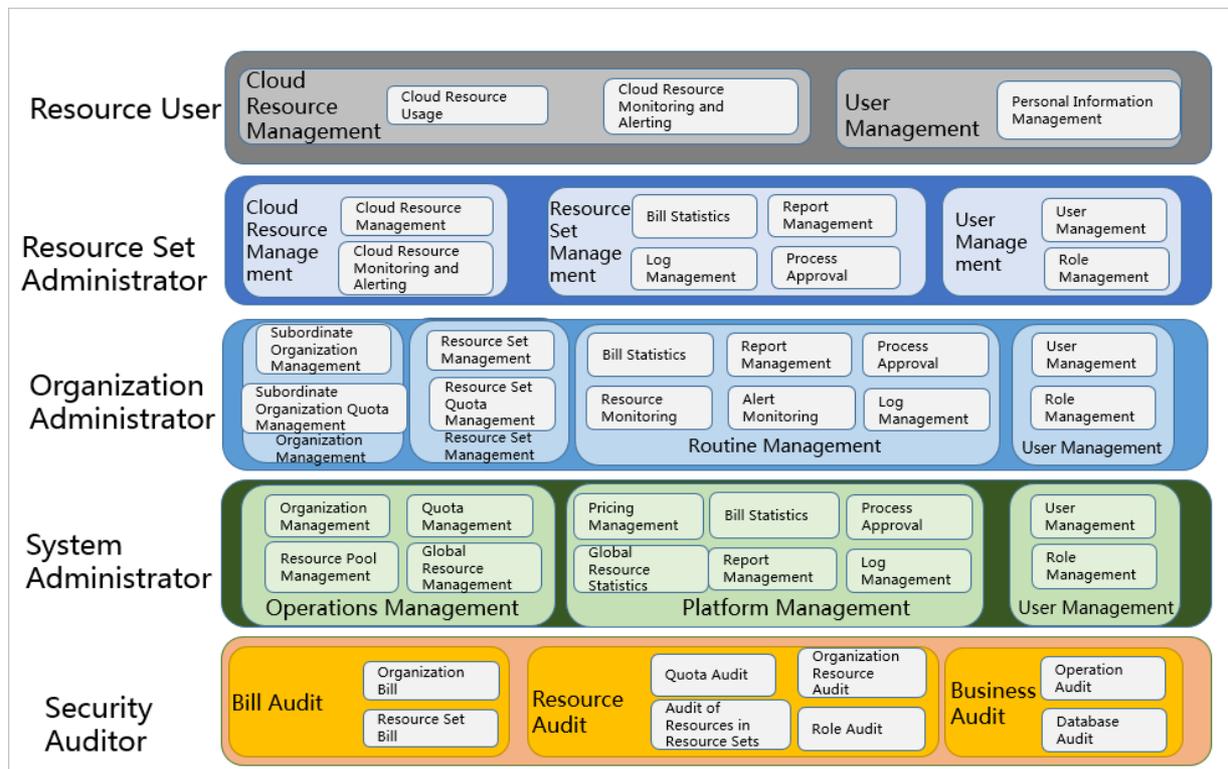
Workflow

ASCM console operations are divided into the following parts:

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

1.2. User roles and permissions

This topic describes roles and their permissions.



Roles and permissions

Role	Permission
Resource user	This role has the permissions to view and modify resources in a resource set and create alert rules.
Resource set administrator	This role has the permissions to create, modify, and delete resources in a resource set and manage the users of the resource set.
Organization administrator	This role has the permissions to manage an organization and its subordinate organizations, create, modify, and delete the resources of organizations, create and view alert rules for resources, and export reports.
Operations administrator	This role has read and write permissions on all resources.
Security auditor	This role performs security audit on the ASCM console and has the read-only permissions on operation logs of the ASCM console.
Platform administrator	This role has the permissions to initialize the system and create operations administrators.
Resource auditor	This role has the read-only permissions on all resources in the ASCM console.
Organization security administrator	This role manages the security of an organization, including the security of hosts, applications, and networks. This role has the read-only permissions on operation logs of the ASCM console and read and write permissions on cloud databases, ECS instances, and Apsara Stack Security.
Security system configuration administrator	This role configures system security features such as the upgrade center and global configurations. This role has read and write permissions on the upgrade, protection, and configuration features of Apsara Stack Security.
Global organization security administrator	This role manages the security of global tenants by using Cloud Security Operation Center (SOC). This role has read and write permissions on all features of Apsara Stack Security.
Platform security administrator	This role manages the security of the ASCM console by using SOC.
Global organization security auditor	This role checks the security conditions of all organizations by using SOC. This role has the read-only permissions on operation logs of the ASCM console and all features of Apsara Stack Security.
Platform security auditor	This role checks the security conditions of the ASCM console by using SOC. This role has the read-only permissions on operation logs of the ASCM console, Server Guard, Cloud Firewall, Sensitive Data Discovery and Protection, SOC, system configurations, and Web Application Firewall (WAF) configurations as well as read and write permissions on Anti-DDoS, Threat Detection, and Update Center of Apsara Stack Security.
Platform Security Configuration Administrator	This role configures and has read and write permissions on security services in the ASCM console, such as Server Guard and WAF.
Organization resource auditor	This role has the read-only permissions on all resources in an organization to which it belongs.

1.3. Log on to the ASCM console

This topic describes how to log on to the Apsara Stack Cloud Management (ASCM) console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

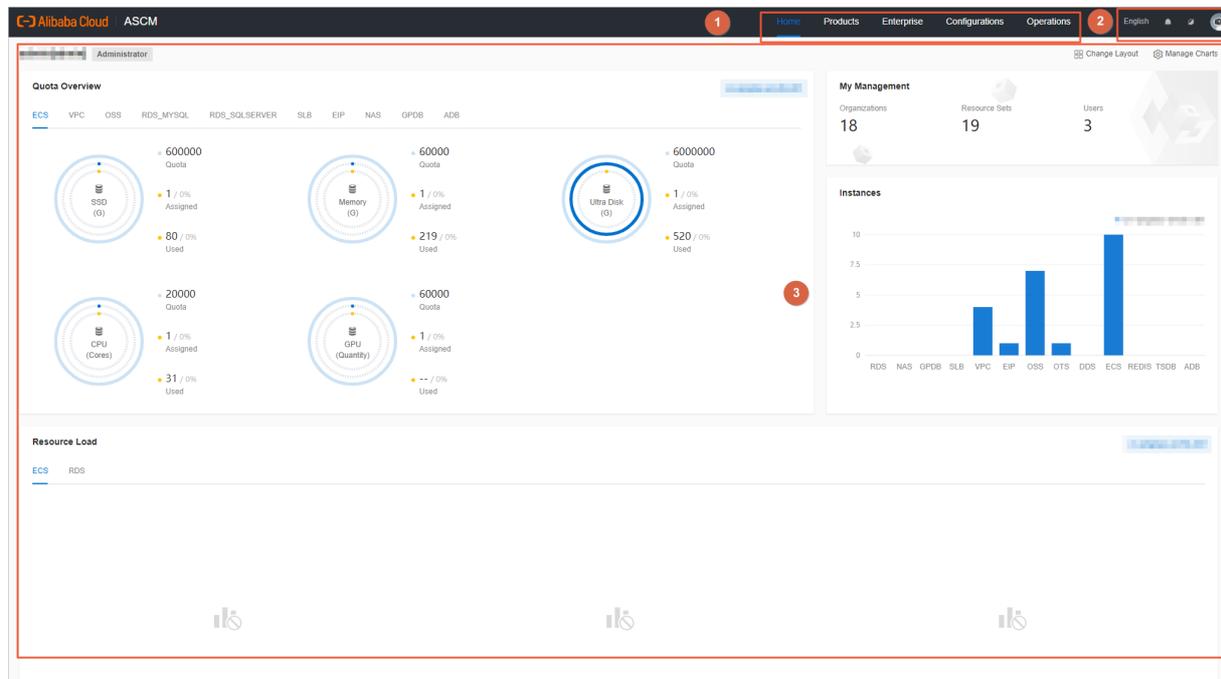
- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click Login to go to the ASCM console homepage.

1.4. Web page introduction

The web page of the ASCM console consists of the top navigation bar, information section of the current logon user, and operation section.

ASCM console page



Functional sections of the web page

Section		Description
1	Top navigation bar	<p>This section includes the following modules:</p> <ul style="list-style-type: none"> • Home: uses charts to display the usage and monitoring data of existing system resources in each region. • Products: manages all types of basic cloud services and resources. • Enterprise: manages organizations, resource sets, roles, users, logon policies, user groups, ownership, and resource pools. • Configurations: manages resource pools, password policies, specifications, menus, and RAM roles. • Operations: manages the daily operations of cloud resources, including usage statistics and quotas. • Security: provides operation logs and system logs.
2	Information section of the current logon user	<ul style="list-style-type: none"> •  English: allows you to switch between English, simplified Chinese, and traditional Chinese. • : provides message notifications. • : allows you to switch between day and night mode. • User Information: Click the  icon of the current logon user. The User Information and Exit menu items are displayed. On the User Information page, you can perform the following operations: <ul style="list-style-type: none"> ◦ View basic information. ◦ Modify personal information. ◦ Change the logon password. ◦ View the AccessKey pair of your Apsara Stack tenant account. ◦ Switch the current role. ◦ Enable or disable alert notification.
3	Operation section	<p>Operation section: the information display and operation section.</p>

1.5. Initial configuration

1.5.1. Configuration description

Before you use the ASCM console, you must complete a series of basic configuration operations as an administrator, such as creating organizations, resource sets, users, and roles and initializing resources. This is the initial system configuration.

Based on the service-oriented principle, the ASCM console manages the organizations, resource sets, users, and roles of cloud data centers in a centralized manner to grant different resource access permissions to different users.

- **Organization**

After the ASCM console is deployed, a root organization is automatically generated. You can create other organizations under the root organization.

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

- Resource set

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

- User

A user is a resource manager and user.

- User group

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

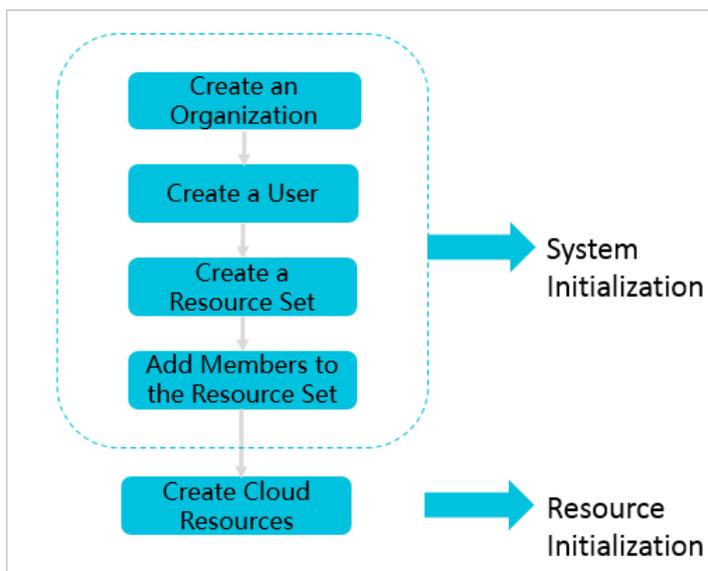
The following table describes the relationships among organizations, resource sets, users, roles, and cloud resources.

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

1.5.2. Configuration process

This topic describes the initial configuration process.

Before using the Apsara Stack Cloud Management (ASCM) console, you must complete the initial system configurations as an administrator according to the process shown in the following figure.



1. **Create an organization**

Create an organization to store resource sets and their resources.

2. **Create a user**

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

3. **Create a resource set**

Create a resource set before you apply for resources.

4. **Add a member to a resource set**

Add members to the resource set.

5. **Create cloud resources**

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

1.6. Monitoring

1.6.1. View the workbench

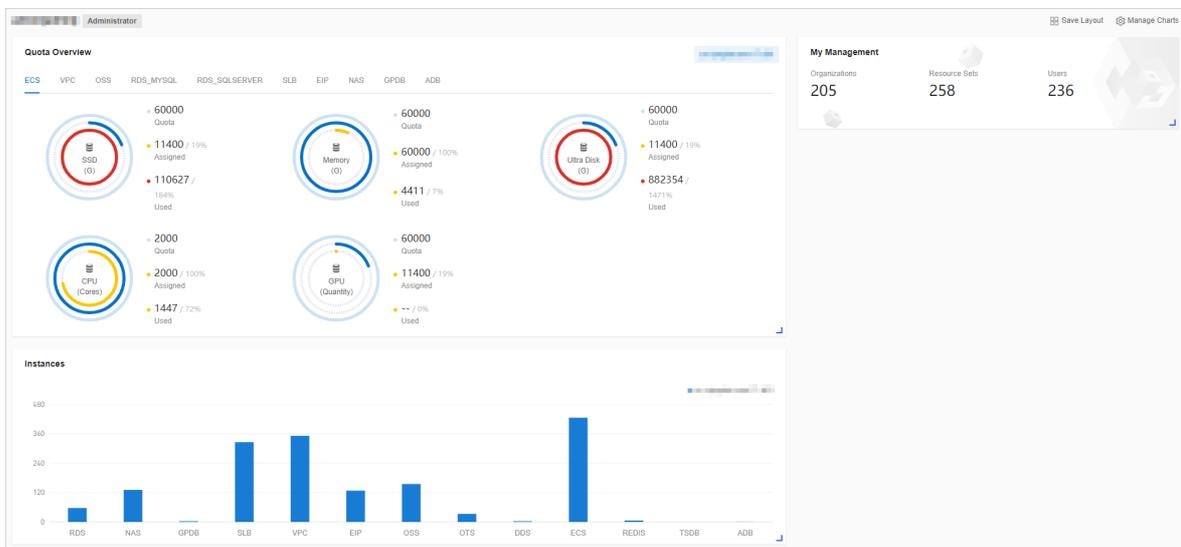
The ASCM console uses charts to keep you up to date on the current usage of resources.

Context

Note The resource types displayed may vary by region type. See your dashboard for available resource types.

Procedure

1. **Log on to the ASCM console.**By default, the workbench page appears when you log on to the ASCM console. To return to the workbench page from other pages, click Home in the top navigation bar.



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

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

- o **Quota Overview**

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

- o **Instances**

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

- **Instance Trends**

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

- **Resource Load**

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

- **Alert Rules**

Shows the number of alerts and details of the alerts.

- **Instances by Resource Set**

Displays the number of instances in all resource sets.

- **My Management**

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

1.6.2. CloudMonitor

1.6.2.1. Cloud Monitor overview

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

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

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

1.6.2.2. Metrics

This topic describes the metrics available for each service.

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

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

Operating system metrics for ECS

Metric	Description	Unit
Host.cpu.total	The CPU utilization of an ECS instance.	%
Host.mem.usedutilization	The memory usage of an ECS instance.	%
Host.load1	The system loads over the last one minute. This metric is not available for Windows operating systems.	N/A

Metric	Description	Unit
Host.load5	The system loads over the last five minutes. This metric is not available for Windows operating systems.	N/A
Host.load15	The system loads over the last 15 minutes. This metric is not available for Windows operating systems.	N/A
Host.disk.utilization	The disk usage of an ECS instance.	%
Host.disk.readbytes	The number of bytes read from the disk per second.	byte/s
Host.disk.writebytes	The number of bytes written to the disk per second.	byte/s
Host.disk.readlops	The number of read requests received by the disk per second.	count/s
Host.disk.writelops	The number of write requests received by the disk per second.	count/s
Host.fs.inode	The inode usage.	%

Basic metrics for ECS

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

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

 **Note**

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

Installation method: On the **CloudMonitor** page, select the target instance from the ECS instance list and click **Batch Install** in the lower part of the page.

Metric data is displayed in the monitoring chart within 5 to 10 minutes after the monitoring plug-in is installed.

Metrics for ApsaraDB RDS for PostgreSQL

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

Metrics for ApsaraDB RDS for MySQL

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

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

Metrics for ApsaraDB RDS for SQL Server

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

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

Metrics for PolarDB

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

Metrics for SLB

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

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

Metrics for monitoring service overview of OSS

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

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

Metrics for request status details of OSS

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

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

Metrics for maximum latency of OSS

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

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

Metrics for successful request category of OSS

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

Metrics for KVStore for Redis

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

Metrics for VPN Gateway

Metric	Dimension	Monitoring period	Unit
Number of inbound packets in a connection per second	User and instance	One minute	pps

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

Metrics for AnalyticDB for PostgreSQL

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

Metrics for ApsaraDB for MongoDB

Tab	Metric	Description	Unit
Basic metric	CPU utilization	The CPU utilization of an ApsaraDB for MongoDB instance.	%
	Memory usage	The memory usage of an ApsaraDB for MongoDB instance.	%
	Disk usage	The disk usage of an ApsaraDB for MongoDB instance.	%
	IOPS usage	The percentage of the IOPS used by an ApsaraDB for MongoDB instance out of the maximum available IOPS.	%

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

Metrics for EIP

Metric	Description	Dimension	Monitoring period	Unit
Inbound bandwidth	The traffic that passes through EIP to ECS per second.	Instance	One minute	bit/s

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

Metrics for API Gateway

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

1.6.2.3. View monitoring charts

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Monitoring Charts** in the **Actions** column corresponding to an instance. On the Monitoring Charts page that appears, you can select a date and time to view the monitoring data of each metric.

1.6.3. Alerts

1.6.3.1. View alarm overview

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Overview**.
4. On the **Overview** page, view the alarm status statistics and alarm logs generated in the last 24 hours.

1.6.3.2. Enable or disable alert notification

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

Prerequisites

You must specify valid contact information when you create a user. If your contact information has changed, you must modify personal information. For more information, see [Modify personal information](#).

Procedure

1. [Log on to the ASCM console](#).
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Notification By** section, select **SMS**, **Email**, or **DingTalk** to enable alert notification. To disable alert notification, you can clear the corresponding check box.

1.6.3.3. View alert logs

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

Context

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

Note

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

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, choose **Alarms > Alarm History**.
4. On the **Alarm Rule History List** page, filter alert information by rule ID, rule name, service, metric, and date. The following table describes the fields in the query result.

Alert information fields

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

1.6.3.4. Alarm rules**1.6.3.4.1. Query alert rules**

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

Context

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

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the Actions column corresponding to an instance. On the **Alarm Rules** page, view the detailed information of alert rules.

1.6.3.4.2. Create an alarm rule

You can create an alarm rule to monitor an instance.

Prerequisites

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

The installation methods are as follows:

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane, choose **Cloud Service Monitoring > ECS**.
4. In the ECS instance list, select the instances that you want to monitor, and click **Batch Install**.

 **Note**

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance to go to its **Alarm Rules** page.



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

6. Click **Create Alarm Rule**.

Parameters for creating an alarm rule

Parameter	Description
Product	The monitored cloud product.
Resource Range	The range of resources associated with the alarm rule, including instances and application groups.
Rule Description	The description of the alarm rule.
Add Alarm Rule	Click Add Rule Description to go to the rule configuration page. For more information, see Parameters for adding an alarm rule .
Effective Time	Only a single alarm is sent during each mute duration, even if the metric value exceeds the alarm rule threshold several consecutive times.
Effective From	An alarm is sent only when the threshold is met during the effective period.
HTTP Callback	The callback URL when the alarm conditions are met.
Alarm Contact Group	The group to which alerts are sent.

Parameters for adding an alarm rule

Parameter	Description
Rule Name	The name of the alarm rule. The name must be 1 to 64 characters in length and can contain letters and digits.
Metric Name	Different products have different monitoring metrics. For more information, see Metrics .

Parameter	Description
Comparison	The comparison between thresholds and observed values. The comparison operators include >, >=, <, and <=. When the comparison rule is satisfied, an alarm rule is triggered.
Threshold And Alarm Level	Different metrics have different reference thresholds.

7. Click OK.

1.6.3.4.3. Disable an alarm rule

You can disable one or more alarm rules as needed.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance.
6. On the **Alarm Rules** page, choose **More > Disable** in the **Actions** column corresponding to the alarm rule to be disabled.
7. In the message that appears, click **OK**.

1.6.3.4.4. Enable an alarm rule

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

Procedure

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

1.6.3.4.5. Delete an alarm rule

You can delete alarm rules that are no longer needed.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, choose **Products > Monitoring and O&M > CloudMonitor**.
3. In the left-side navigation pane of the CloudMonitor page, click **Cloud Service Monitoring**.
4. Click a cloud service.
5. Click **Alarm Rules** in the **Actions** column corresponding to an instance to go to its **Alarm Rules** page.
6. Select the alarm rule that you want to delete and click **Delete** in the **Actions** column.
7. In the message that appears, click **OK**.

1.7. Enterprise

1.7.1. Organizations

1.7.1.1. Create an organization

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, move the pointer over the name of a parent organization, and click  on the right.
5. Choose **Add Organization** from the shortcut menu.
6. In the dialog box that appears, enter an organization name and click **OK**.

1.7.1.2. Query an organization

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the search box below **Organizations**, enter an organization name to query information about the corresponding organization.

1.7.1.3. View organization information

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

Procedure

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

1.7.1.4. Modify the name of an organization

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

Procedure

1. [Log on to the ASCM console](#).
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, move the pointer over the name of an organization and click  on the right.
5. Choose **Edit Organization** from the shortcut menu.
6. In the dialog box that appears, modify the organization name.
7. Click **OK**.

1.7.1.5. Change organization ownership

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

Prerequisites

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

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Change Ownership**.
4. On the **Change Ownership** page, select the target organization and click **Change Ownership** on the right.
5. In the **Change Organization** dialog box, select the destination organization and click **OK** to change the ownership of the target organization and resources sets and users under this organization.

1.7.1.6. Obtain the AccessKey pair of an organization

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

Prerequisites

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

Procedure

1. [Log on to the ASCM console](#).
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, move the pointer over the name of a level-1 organization and click the  icon on the right.
5. Select **AccessKey**.

6. In the message that appears, view the AccessKey pair of the organization.

1.7.1.7. Delete an organization

Administrators can delete organizations that are no longer needed.

Prerequisites

 **Note** Before deleting an organization, make sure that the organization does not contain any users, resource sets, or subordinate organizations. Otherwise, the organization cannot be deleted.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, move the pointer over the name of an organization, and click  on the right.
5. Choose **Delete Organization** from the shortcut menu.
6. In the message that appears, click **OK**.

1.7.2. Resource sets

1.7.2.1. Create a resource set

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. In the upper-left corner of the **Resource Sets** page, click **Create Resource Set**.
5. In the **Create Resource Set** dialog box, set **Name** and **Organization**.
6. Click **OK**.

1.7.2.2. View the details of a resource set

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

Procedure

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

automatically filtered and displayed based on the organization and resource set.

1.7.2.3. Modify the name of a resource set

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

Procedure

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

1.7.2.4. Add a member to a resource set

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

Prerequisites

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

- A resource set is created. For more information, see [Create a resource set](#).
- A user is created. For more information, see [Create a user](#).

Context

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

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

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

Procedure

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

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

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

Prerequisites

- A resource set is created. For more information, see [Create a resource set](#).
- A user group is created. For more information, see [Create a user group](#).

Context

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

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

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

Procedure

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

1.7.2.6. Delete a resource set

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

Prerequisites

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

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Sets**.
4. Click **More** in the **Actions** column corresponding to the target resource set, and select **Delete**.
5. In the message that appears, click **OK**.

1.7.3. Roles

1.7.3.1. Create a custom role

You can add custom roles in the ASCM console to more efficiently grant permissions to users so that different personnel can work with different functions.

Context

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

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.

3. In the left-side navigation pane of the Enterprise page, click Roles.
4. In the upper-right corner of the page, click Create Custom Role.
5. On the Roles page, set the role name and management permissions.

The screenshot shows the 'Roles' page with four numbered steps: 1. Role Name and Management Permissions, 2. Application Permissions, 3. Menu Permissions, and 4. Associated Users. Step 1 is active, showing a form with the following elements:

- Role Name:** A text input field with a character count of 0/64.
- Description:** A text area with a character count of 0/100.
- Sharing Scope:** Three radio buttons: Global, Current Organization, and Subordinate Organization.
- Scope:** Three radio buttons: All Organizations, Specified Organization and Subordinate Organizations, and Resource Set.

The following table describes the role parameters.

Role parameters

Parameter	Description
Role Name	The name of the role. The name can be up to 15 characters in length and can contain only letters and digits.
Description	Optional. The description of the role. The description can be up to 100 characters in length and can contain letters, digits, commas (,), semicolons (;), and underscores (_).
Sharing Scope	<ul style="list-style-type: none"> ○ Global The role is visible and valid to all organizations involved. The default value is Global. ○ Current Organization The role is visible and valid to the organization to which the user belongs. ○ Subordinate Organization The role is visible and valid to the organization to which the user belongs and its subordinate organizations.
Scope	<ul style="list-style-type: none"> ○ All Organizations The permissions apply to all organizations involved. ○ Specified Organization and Subordinate Organizations The permissions apply to the organization to which the user belongs and its subordinate organizations. ○ Resource Set The permissions apply to the resource sets assigned to the user.

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

1.7.3.2. View the details of a role

If you are not certain about the specific permissions of a role, go to the Roles page to view the role permissions.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. Click the name of the role that you want to view. On the **Roles** page, view information about the role.

1.7.3.3. Modify custom role information

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

Context

Information about preset roles cannot be modified.

Procedure

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

1.7.3.4. Copy a role

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

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, click **More** in the **Actions** column corresponding to the target role, and select **Copy**.
5. On the **Roles** page, set the new role name, sharing scope, and management permissions.

Roles

1 **Role Name and Management Permissions**
Specify the role name and management permissions.

2 **Application Permissions**
Specify permissions on applications.

3 **Menu Permissions**
Specify permissions on menus in the console.

4 **Associated Users**
Associate the specified role with users.

*Role Name: 13/64

Description: 73/100

*Sharing Scope: Global Current Organization Subordinate Organization

*Scope: All Organizations Specified Organization and Subordinate Organizations Resource Set

Note The role name must be unique.

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

1.7.3.5. Disable a role

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

Procedure

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

1.7.3.6. Enable a role

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

Procedure

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

1.7.3.7. Delete a custom role

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

Prerequisites

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

Procedure

1. **Log on to the ASCM console** as an administrator.

2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. Click **More** in the **Actions** column corresponding to the target role, and select **Delete**.
5. In the message that appears, click **OK**.

1.7.4. Users

1.7.4.1. System users

1.7.4.1.1. Create a user

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. Use one of the following methods to open the **Create User** window:
 - In the left-side navigation pane of the **Enterprise** page, click **Organizations**. In the **Users** section of the **Organizations** page, click **Create User**.
 - In the left-side navigation pane of the **Enterprise** page, click **Users**. On the **System Users** tab of the **Users** page, click **Create**.
4. In the dialog box that appears, configure the parameters.

Parameter	Description
Username	The Apsara Stack account name of the user. The name must be 3 to 30 characters in length, and can contain letters, digits, hyphens (-), underscores (_), periods (.), and at signs (@). It must start with a letter or digit.
Display Name	The display name of the user. The name must be 2 to 30 characters in length, and can contain letters, digits, hyphens (-), underscores (_), periods (.), and at signs (@).
Roles	The role to be assigned to the user.
Organization	The organization to which the user belongs.
Logon Policy	<p>The logon policy that restricts the logon time and IP addresses of the RAM user. The default policy is automatically bound to new users.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Note The default policy does not restrict the time period and IP addresses for users to log on. To restrict the logon time and IP addresses of a user, you can modify the logon policy of the user or create a logon policy for the user. For more information, see Create a logon policy.</p> </div>
Mobile Number	<p>The mobile number of the user. The mobile number is used by the system to notify users of resource application and usage. Make sure that the entered mobile number is correct.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Note If the mobile number is changed, be sure to update it on the system in a timely manner.</p> </div>

Parameter	Description
Landline Number	Optional. The landline number of the user. It must be 4 to 20 characters in length and can contain only digits and hyphens (-).
Email	<p>The email address of the user. Emails about the usage and requests for resources will be sent to the email address. Make sure that the specified email address is correct.</p> <p> Note If the email address is changed, be sure to update it on the system in a timely manner.</p>
DingTalk Key	The key of the chatbot for the DingTalk group where the user is a member. For more information about how to configure the key, see DingTalk development documentation .
Notify User by SMS	<p>After this option is selected, the ASCM console will inform the user configured as the alert contact by SMS whenever an alert is generated.</p> <p> Note You must configure an SMS server to receive an SMS message each time an alert is triggered. For more information, contact on-site O&M engineers.</p>
Notify User by Email	<p>After this option is selected, the ASCM console will inform the user configured as the alert contact by email whenever an alert is generated.</p> <p> Note You must configure an email server to receive an email each time an alert is triggered. For more information, contact on-site O&M engineers.</p>
Notify User by DingTalk	After this option is selected, the ASCM console will inform the user configured as the alert contact by DingTalk whenever an alert is generated.

5. Click OK.

1.7.4.1.2. Query a user

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

Procedure

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

1.7.4.1.3. Modify user information

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Click **More** in the **Actions** column corresponding to a user, and choose **Edit** from the shortcut menu.
6. In the **Modify User Information** dialog box, enter the relevant information and click **OK**.

1.7.4.1.4. Change user roles

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

Change user roles by using user management

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Click **More** in the **Actions** column corresponding to the target user, and select **Authorize**.
6. In the **Role** field, add, delete, or change user roles.
7. Click **OK**.

Change user roles by changing ownership

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

Note

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

8. Click **OK**.

1.7.4.1.5. Modify the information of a user group

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, click **Users**.
4. Click the **System Users** tab, select a target user, and then click **More** in the **Actions** column.
 - Select **Add to User Group**. In the dialog box that appears, select the target user group and click **OK** to

add the user to the user group.

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

1.7.4.1.6. Modify a user logon policy

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

Prerequisites

A new logon policy is created. For more information about how to create a logon policy, see [Create a logon policy](#).

Modify a user logon policy

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

Modify multiple user logon policies at a time

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Select multiple users.
6. In the upper-right corner of the page, click **Logon Policy**.
7. In the **Assign Logon Policies** dialog box, select a logon policy.

1.7.4.1.7. View the initial password of a user

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

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Select the username that you want to query.
6. You can use one of the following methods to view the initial password of a user:
 - Click **View Initial Password** in the upper-right corner of the **Users** page to view the initial password.
 - Click **More** in the **Actions** column corresponding to the user, and choose **User Information** from the shortcut menu. On the user information page, click **View Password** to view the initial password.

1.7.4.1.8. Reset the password of a user

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

Prerequisites

The logon password of a user can be reset by only the user and those who have the permissions to create resource sets for the user.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. Select the user for which you want to reset the password.
6. Click **More** in the **Actions** column corresponding to the user, and choose **User Information** from the shortcut menu.
7. Click **Reset Password**. After the password is reset, a message is displayed, indicating that the password has been reset. If you want to view the initial password after password reset, click **View Password**.

1.7.4.1.9. Disable and enable a user

You can disable a user to prevent the user from logging on to the Apsara Stack Cloud Management (ASCM) console. Disabled users must be re-enabled before they can log on to the ASCM console again.

Context

By default, users are enabled when they are created.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **System Users** tab.
5. You can perform the following operations on the current tab:
 - Select a user whose **Status** is **Enabled**, click **More** in the **Actions** column, and choose **Disable** from the shortcut menu to disable the user.
 - Select a user whose **Status** is **Disabled**, click **More** in the **Actions** column, and choose **Enable** from the shortcut menu to enable the user.

1.7.4.1.10. Delete a user

You can delete a specific user as an administrator.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. On the **Enterprise** page, use one of the following methods to delete a user:
 - In the left-side navigation pane of the **Enterprise** page, click **Users**. On the page that appears, click the **System Users** tab. Click **More** in the **Actions** column corresponding to the target user, and select **Delete**.
 - In the left-side navigation pane of the **Enterprise** page, click **Organizations**. On the page that appears, find the **Users** section. Find the target user, click **More** in the **Actions** column, and then select **Delete**.
4. Click **OK**.

1.7.4.2. Historical users

1.7.4.2.1. Query historical users

You can check whether a user has been deleted, or quickly find and restore the user.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **Historical Users** tab.
5. Enter the username that you want to query in the **Username** search box.
6. Click **Search**.

1.7.4.2.2. Restore historical users

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

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Click the **Historical Users** tab.
5. Find the user to be restored and click **Restore** in the **Actions** column.
6. In the **Restore User** dialog box that appears, select an organization and a role.
7. Click **OK**.

1.7.5. Logon policies

1.7.5.1. Create a logon policy

To improve the security, you can create a logon policy as an administrator to control the logon time and IP addresses of a user.

Context

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

A default policy without restrictions on the logon time and IP addresses is automatically generated in the ASCM console. The default policy cannot be deleted.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. In the upper-right corner of the page, click **Create**.

- In the Create Logon Policy dialog box, set Name, Sharing Scope, Policy Properties, Time Period, and IP Address.

Create Logon Policy
✕

***Name:**

0/50

Description:

--

***Sharing Scope:**

Global
▾

***Policy Properties:**

Blacklist Whitelist

Time Period:

Select start time
🕒

—

Select end time
🕒

[+ Add Time Period](#)
 The logon time period cannot be empty and start time cannot be later than end time.

IP Address:

[+ Add CIDR Block](#)
 Specify the CIDR block in the format such as 192.168.1.0/24. Use a 32-bit subnet mask in the CIDR block to specify a single IP address.
 CIDR blocks cannot overlap each other.

OK
Cancel

Parameters for creating a logon policy

Parameter	Description
Name	The name of the logon policy. The name must be 2 to 50 characters in length and can contain only letters and digits. The name must be unique in the system.
Description	The description of the logon policy.
Sharing Scope	The scope in which the role is visible. <ul style="list-style-type: none"> ○ Global: The role is globally visible. The default value is Global. ○ Current Organization: The role is visible only in the current organization and is not visible in subordinate organizations. ○ Subordinate Organization: The role is visible in the current organization and all its subordinate organizations.
Policy Properties	The authentication method of the logon policy. <ul style="list-style-type: none"> ○ Whitelist: Logon is allowed if the parameter settings are met. ○ Blacklist: Logon is denied if the parameter settings are met.
Time Period	The permitted logon time period. When this policy is configured, users can log on to the ASCM console only during the configured period. Specify the time in minutes in a 24-hour clock. Example: 16:32 .

Parameter	Description
IP Address	<p>The permitted CIDR block.</p> <ul style="list-style-type: none"> ○ If the Policy properties parameter is set to Whitelist, IP addresses within this CIDR block are allowed to log on to the ASCM console. ○ If the Policy properties parameter is set to Blacklist, IP addresses within this CIDR block are not allowed to log on to the ASCM console.

1.7.5.2. Query a logon policy

When providing services, the Apsara Stack Cloud Management (ASCM) console automatically generates a default policy without restrictions on the logon time and IP addresses.

Context

When providing services, the Apsara Stack Cloud Management (ASCM) console automatically generates a default policy without restrictions on the logon time and IP addresses.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Enter the name of the policy that you want to view and click **Search**.
5. View the logon policy, including the permitted logon time and IP addresses.

1.7.5.3. Modify a logon policy

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Modify** from the shortcut menu.
5. In the **Modify Logon Policy** dialog box that appears, modify the logon policy information.
6. Click **OK**.

1.7.5.4. Disable a logon policy

You can disable logon policies that are no longer needed.

Procedure

1. [Log on to the ASCM console](#).
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Disable** from the shortcut menu.

1.7.5.5. Enable a logon policy

You can re-enable disabled logon policies.

Procedure

1. [Log on to the ASCM console](#).
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Enable** from the shortcut menu.

1.7.5.6. Delete a logon policy

You can delete logon policies that are no longer needed.

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Logon Policies**.
4. Click **More** in the **Actions** column corresponding to a policy, and choose **Delete** from the shortcut menu.
5. In the message that appears, click **OK**.

1.7.6. User groups

1.7.6.1. Create a user group

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

Prerequisites

Before creating a user group, you must create an organization. For more information, see [Create an organization](#).

Context

Relationship between user groups and users:

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

Relationship between user groups and organizations:

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

Relationship between user groups and roles:

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

Relationship between user groups and resource sets:

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

Procedure

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

6. Click OK.

1.7.6.2. Add users to a user group

You can add users to a user group.

Procedure

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

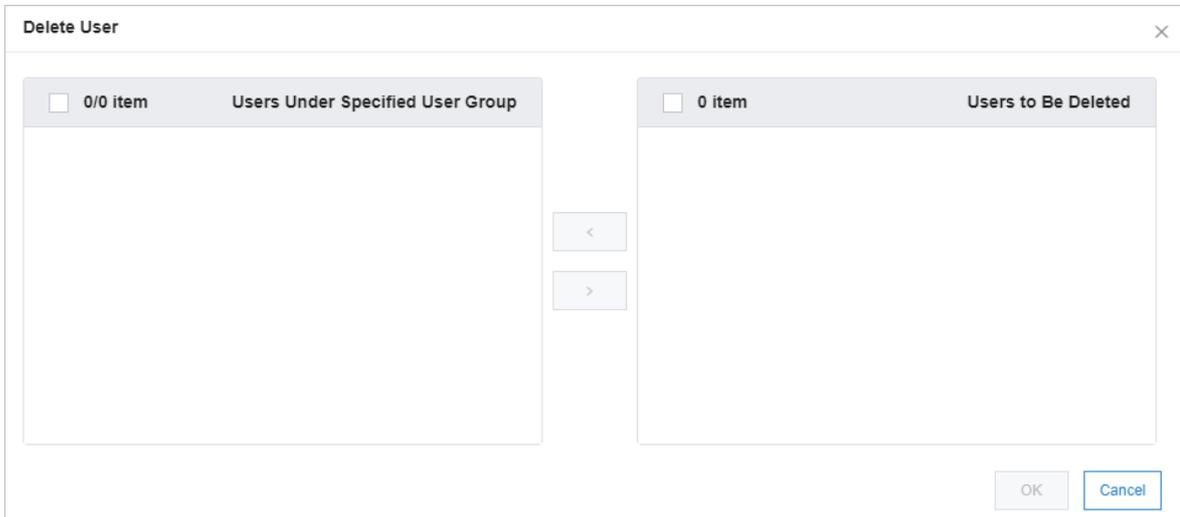
6. Click OK.

1.7.6.3. Delete users from a user group

You can delete users from a user group.

Procedure

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



6. Click OK.

1.7.6.4. Add a role

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

Context

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

Procedure

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

1.7.6.5. Delete a role

You can delete existing roles.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the top navigation bar, click Enterprise.
3. In the left-side navigation pane of the Enterprise page, click User Groups.
4. Click Delete Role in the Actions column corresponding to a user group.
5. In the Confirm message that appears, click OK.

1.7.6.6. Modify the name of a user group

You can modify the names of user groups.

Procedure

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

1.7.6.7. Delete a user group

You can delete user groups that are no longer needed.

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Click **Delete User Group** in the **Actions** column corresponding to a user group.
5. In the **Confirm** message that appears, click **OK**.

1.7.7. Resource pools

1.7.7.1. Update associations

The Apsara Stack Cloud Management (ASCM) console can be deployed in multiple regions. You can update the associations between organizations and regions.

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Resource Pool Management**.
4. In the left-side organization tree, click the name of an organization.
5. In the corresponding region list, select the names of regions to be associated.
6. Click **Update Association**.

1.7.8. Change ownership

You can change the ownership of instances in resource sets.

Change the ownership of an instance

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Change Ownership**.
4. Click the  icon to the left of the target organization and click a resource set.
5. In the resource list on the right-side of the page, set a service type and a resource type, enter an instance

ID, and then click **Search** to query the target instance.

6. You can change the ownership of a single instance or add multiple instances to the same resource set at a time.
 - **Single change:** Click **Change Ownership** in the **Actions** column corresponding to the target instance.
 - **Batch change:** Select multiple target instance IDs and click **Batch Change Ownership**.
7. In the **Change Resource Set** dialog box, select the target resource set and click **OK**.

1.8. Configurations

1.8.1. Password policies

You can configure password policies for user logons.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Password Policies**.
4. On the **Password Policy** page, set the password policy parameters.

The screenshot shows the 'Password Policy' configuration page. It includes the following settings:

- Password Length:** 10 To 32 Digits (Minimum: 8)
- The Password Must Contain:** Lowercase Letters, Uppercase Letters, Digits, Special Characters
- Logon Disabled After Password Expires:** Yes, No
- Password Validity Period (Days):** 90 (The value must be 0 to 1095. The value 0 specifies that the password will not expire.)
- Password Attempts:** allows a maximum of 5 password attempts within an hour. (The value must be 0 to 32. The value 0 specifies that the password history check is disabled.)
- Password History Check:** disables the first 5 passwords. (The value must be 0 to 24. The value 0 specifies that the password history check is disabled.)

Buttons for **Save** and **Reset** are located at the bottom of the form.

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

1.8.2. Menus

1.8.2.1. Create a menu

You can create a menu and add its URL to the ASCM console for quick access.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. On the **Main Menu** page, click **Create** in the upper-right corner.
5. In the **Create** dialog box that appears, set the menu parameters.

Create
✕

*Title:

URL:

*Console Type: asconsole oneconsole other
Different console types correspond to different service endpoints. If you select Other, the endpoint configured in the URL field is used.

Icon:

*Identifier:

*Order: +
-

*Parent Level: ▼

*Open With: Default New Window

Description:

OK
Cancel

Menu parameters

Parameter	Description
Title	The display name of the menu.
URL	The URL of the menu.
Console Type	<p>Different console types correspond to different domain names.</p> <ul style="list-style-type: none"> ○ oneconsole: You only need to enter the path in the URL field. The domain name is automatically matched. ○ asconsole: You only need to enter the path in the URL field. The domain name is automatically matched. ○ other: You must enter the domain name in the URL field.
Icon	The icon displayed in the left-side navigation pane. The icon cannot be changed.
Identifier	The unique identifier of the menu in the system. This identifier can be used to indicate whether the menu is selected in the navigation bar. The identifier cannot be changed.
Order	The display order among the same-level menus. The larger the value, the lower the display order. Leave the Order field empty.
Parent Level	The displayed tree structure.
Open With	Specifies whether to open the menu in the current window or in a new window.
Description	The description of the menu.

1.8.2.2. Modify a menu

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

Prerequisites

Default menus cannot be modified.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the top navigation bar, click Configurations.
3. In the left-side navigation pane of the Configurations page, click Menu Settings.
4. Click Edit in the Actions column corresponding to a menu.
5. In the Edit dialog box that appears, modify relevant information about the menu.

The screenshot shows an 'Edit' dialog box with the following fields and options:

- *Title:** A text input field containing a blurred value.
- URL:** A text input field containing the value `/module/config?identifier=blink&jumpUrl=true#/jump/blink`.
- *Console Type:** Radio buttons for `asconsole` (selected), `oneconsole`, and `other`. Below this is a note: "Different console types correspond to different service endpoints. If you select Other, the endpoint configured in the URL field is used."
- Icon:** A text input field containing the value `wind-rc-product-icon glyph-sc rotate-0`.
- *Identifier:** A text input field containing the value `blink`.
- *Order:** A text input field containing the value `21`, with '+' and '-' buttons to the right.
- *Parent Level:** A dropdown menu with the value `Products`.
- *Group:** A dropdown menu with the value `Please select`.
- *Open With:** Radio buttons for `Default` and `New Window` (selected).
- Description:** A text input field containing the value `Please input`.

At the bottom right of the dialog box are two buttons: **OK** and **Cancel**.

1.8.2.3. Delete a menu

You can delete menus that are no longer needed.

Prerequisites

Default menus cannot be deleted.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the top navigation bar, click Configurations.
3. In the left-side navigation pane of the Configurations page, click Menu Settings.

4. Click **Delete** in the **Actions** column corresponding to a menu.
5. In the message that appears, click **OK**.

1.8.2.4. Display or hide menus

You can display or hide menus as follows:

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Menu Settings**.
4. Select or clear the check box in the **Displayed** column corresponding to a menu.

1.8.3. Specifications

1.8.3.1. Specification parameters

This topic describes the specification parameters of each resource type.

OSS

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

NAT Gateway

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

AnalyticDB for PostgreSQL

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

Parameter	Description
Version	The version of AnalyticDB for PostgreSQL.
Node	The number of nodes that can be configured for AnalyticDB for PostgreSQL.

AnalyticDB for MySQL

Parameter	Description
Specifications	The specifications that can be configured for AnalyticDB for MySQL.
Minimum Nodes and Maximum Nodes	The minimum and maximum number of nodes that can be configured for AnalyticDB for MySQL.
Storage Space	The storage space that can be configured for AnalyticDB for MySQL.

SLB

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

ApsaraDB for RDS

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

Parameter	Description
CPUs	The number of CPU cores that can be configured for ApsaraDB for RDS.
Maximum Connections	The maximum number of connections that can be configured for ApsaraDB for RDS.
Storage	The storage space that can be configured for ApsaraDB for RDS.
Memory (GB)	The memory size that can be configured for ApsaraDB for RDS.
Share Type	The share type that can be configured for ApsaraDB for RDS.

DRDS

Parameter	Description
Instance Type	The instance type that can be configured for Distributed Relational Database Service (DRDS).
Instance Type Name	The name of the instance type that can be configured for DRDS.
Specifications	The specifications that can be configured for DRDS.
Specifications Name	The name of the specifications that can be configured for DRDS.

ECS

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

IPv6 Translation Service

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

REDIS

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

MongoDB

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

Parameter	Description
Storage Space	The storage space that can be configured for ApsaraDB for MongoDB.
Minimum Storage	The minimum storage space that can be configured for ApsaraDB for MongoDB.
Maximum Storage	The maximum storage space that can be configured for ApsaraDB for MongoDB.

HITSDB

Parameter	Description
Specifications Name	The name of the specifications that can be configured for Time Series Database (TSDB).
Maximum Time Limit	The maximum duration of connections in TSDB.
TPS	The number of transactions that can be processed per second by TSDB.
Storage Space	The storage space that can be configured for TSDB.

1.8.3.2. Create specifications

You can customize specifications for each resource type.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Click the resource type for which you want to create specifications.
5. In the upper-right corner of the page, click **Create Specifications**.
6. In the dialog box that appears, set the parameters. For more information about specification parameters, see [Specification parameters](#).
7. Click **OK**.

1.8.3.3. View specifications

You can view the specifications of each resource type.

Procedure

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

1.8.3.4. Disable specifications

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

Procedure

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

1.8.3.5. Export specifications

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. Click the resource type for which you want to create specifications.
5. In the upper-right corner of the page, click **Export**.
6. Save the specifications file to the target path.

1.8.3.6. View specifications of each resource type in previous versions

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Specifications**.
4. On the **Specifications** page, click the resource type for which you want to view specifications.
5. Click the **Specifications History** tab. View the detailed information in the specifications list.

1.8.4. Message center

1.8.4.1. View internal messages

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

Context

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, move the pointer over the  icon and click **More**.

3. In the left-side navigation pane of the **Message Center** page, click the target message scope.
 - Choose **Internal Messages > All Messages** to view all messages, including unread and read messages.
 - Choose **Internal Messages > Unread Messages** to view unread messages.
 - Choose **Internal Messages > Read Messages** to view read messages.

1.8.4.2. Mark messages as read

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, move the pointer over the  icon and click **More**.
3. In the left-side navigation pane of the **Message Center** page, choose **Internal Messages > Unread Messages**.
4. On the **Unread Messages** page, click **Mark as Read** in the **Actions** column.
5. In the **Mark as Read** message, click **OK**.

1.8.5. Resource pool management

You can modify the maximum usage of each resource.

Prerequisites

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **Resource Pool Management**.
4. On the **Resource Pools** page, click the  icon in the target module and modify the number of resources.

Resource Pool Configuration

Region

ECS				VPC				OSS			
Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used
CPU Quota	20,000	Unknown	31	VPC Quota	10,000	Unlimited	4	OSS Quota (GB)	Not Set	Unknown	0
Memory Quota (GB)	60,000	Unknown	219								
GPU Quota	60,000	Unknown	0								
SSD Quota (GB)	600,000	Unknown	80								
Ultra Disk Quota (GB)	6,000,000	Unknown	520								

RDS-MYSQL				RDS-SQLServer				RDS-postgreSQL			
Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used
CPU Quota	Not Set	Unknown	0	CPU Quota	Not Set	Unknown	0	CPU Quota	Not Set	Unknown	0
Memory Quota (GB)	Not Set	Unknown	0	Memory Quota (GB)	Not Set	Unknown	0	Memory Quota (GB)	Not Set	Unknown	0
Disk Quota (GB)	Not Set	Unknown	0	Disk Quota (GB)	Not Set	Unknown	0	Disk Quota (GB)	Not Set	Unknown	0

SLB				EIP				ODPS			
Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used	Item	Logical Inventory	Physical Inventory	Used
Virtual IP Quota	2,304	2,304	0	EIP Quota	10,000	Unlimited	1	CU Quota	Not Set	Unknown	0
Public Virtual IP Quota	512	512	0					Disk Quota (GB)	Not Set	Unknown	0

5. Click the icon to complete modification.

1.9. Operations

1.9.1. Quotas

1.9.1.1. Quota parameters

This topic describes the quota parameters of each service.

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

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

ECS

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

VPC

Parameter	Description
VPC Quota	The maximum number of VPCs that you can configure.

OSS

Parameter	Description
OSS Quota	The maximum capacity that you can allocate for OSS.

RDS-MySQL

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

RDS-SQLServer

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

RDS-PostgreSQL

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

SLB

Parameter	Description
Virtual IP Quota	The maximum number of internal IP addresses that you can configure for SLB.
Public Virtual IP Quota	The maximum number of public IP addresses that you can configure for SLB.

EIP

Parameter	Description
EIP Quota	The maximum number of EIPs that you can configure.

ODPS

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

DRDS

Parameter	Description
CPU Quota	The total number of CPUs that you can configure for DRDS.

NAS

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

GPDB

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

ADB

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

1.9.1.2. Set quotas for a cloud service

The Apsara Stack Cloud Management (ASCM) console allows you to set quotas to properly allocate resources among organizations.

Prerequisites

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

Context

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

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

Procedure

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

For more information about quota parameters, see [Quota parameters](#).

1.9.1.3. Modify quotas

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

Context

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

Procedure

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

For more information about quota parameters, see [Quota parameters](#).

1.9.1.4. Reset quotas

Administrators can reset quotas as needed.

Prerequisites

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Operations**.

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

1.9.2. Usage statistics

1.9.2.1. View the usage statistics of cloud resources

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

Context

The cloud resources that can be metered include ECS, VPC, SLB, OSS, ApsaraDB RDS for MySQL, EIP, Apsara File Storage NAS, Tablestore, PolarDB-X, KVStore for Redis, AnalyticDB for MySQL, AnalyticDB for PostgreSQL, ApsaraDB for MongoDB, Message Queue, ApsaraDB RDS for PostgreSQL, and ApsaraDB RDS for SQL Server.

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

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Operations**.
3. In the left-side navigation pane of the **Operations** page, click **Usage Statistics**.
4. In the **Resource Type** section, click **Elastic Compute Service ECS**.
5. In the **Search Conditions** section, set **Time Period**, **Organization**, **Resource Set**, **Region**, and **Instance ID** to filter resources. You can view the statistics in the console or click **Export** in the upper-right corner to export the statistics to your personal computer in the `.xls` format.

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

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

1.10. Security

1.10.1. View operation logs

You can view operation logs to obtain up-to-date information about various resources and functional modules in the Apsara Stack Cloud Management (ASCM) console. You can also export operation logs to your personal computer.

Procedure

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

The following table describes the fields in the query result.

Fields in the query result

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

- (Optional) Click **Export** to export the logs displayed on the current page to your personal computer in `xls` format.

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

1.11. RAM

1.11.1. RAM introduction

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

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

RAM provides the following features:

- RAM role

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

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

- User group

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

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

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

1.11.2. Permission policy structure and syntax

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

Policy characters and usage rules

- Characters in a policy

- The following characters are JSON tokens and are included in policies: `{ } [] " , : .`
- The following characters are special characters in the syntax and are not included in policies: `= < > () |`.

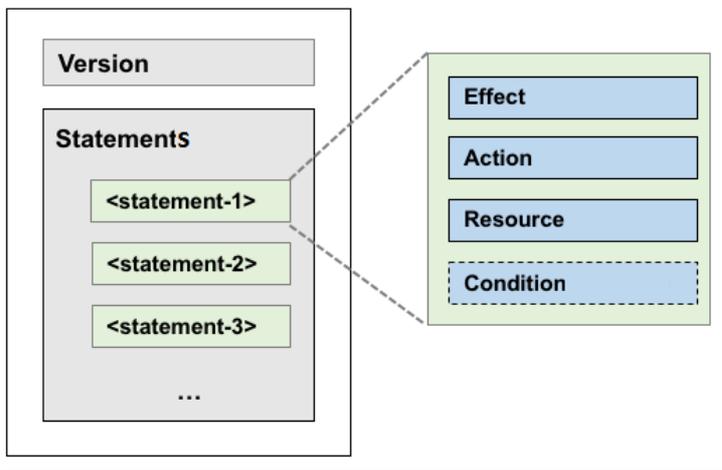
- Use of characters

- If an element can have more than one value, you can perform the following operations:
 - Separate multiple values by using commas (,) as delimiters between each value and use an ellipsis (...) to describe the remaining values. Example: [<action_string>, <action_string>, ...] .
 - Include only one value. Examples: "Action": [<action_string>] and "Action": <action_string> .
- A question mark (?) following an element indicates that the element is optional. Example: <condition_block ?> .
- A vertical bar (|) between elements indicates multiple options. Example: ("Allow" | "Deny") .
- Elements that must be text strings are enclosed in double quotation marks ("). Example: <version_block> = "Version" : ("1") .

Policy structure

The policy structure includes the following components:

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



Policy syntax

```

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

```

Description:

- The current policy version is 1.
- The policy can have multiple statements.
 - The effect of each statement can be either `Allow` or `Deny` .

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

- Each statement can have its own conditions.

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

- You can attach multiple policies to a RAM user. If policies that apply to a request include an `Allow` statement and a `Deny` statement, the Deny statement overrides the Allow statement.

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

Policy format check

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

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

1.11.3. RAM roles

1.11.3.1. View basic information about a RAM role

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. On the **Roles** page, click the name of the target RAM role.
5. In the basic information section, click the **User Groups and Permissions** tabs to view relevant information.

1.11.3.2. Create a RAM role

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create RAM Role**.
5. On the **Roles - Create RAM Role** page that appears, set **Role Name** and **Description**.
6. Click **Create**.

1.11.3.3. Add a permission policy

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.

2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Add Permission Policy**.
7. In the dialog box that appears, enter information about the permission policy.

For more information about how to enter the policy content, see [Permission policy structure and syntax](#).

1.11.3.4. Modify the content of a RAM permission policy

You can modify the content of a RAM permission policy as needed.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click the name of a permission policy in the **Permission Policy Name** column.
7. In the **Modify Permission Policy** dialog box that appears, modify the relevant information and click **OK**. For more information about how to modify the policy content, see [Permission policy structure and syntax](#).

1.11.3.5. Modify the name of a RAM permission policy

You can modify the name of a RAM permission policy as needed.

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **Permissions** tab. Click the name of a permission policy in the **Permission Policy Name** column.
6. In the **Modify Permission Policy** dialog box that appears, modify the permission policy name.

1.11.3.6. Add a RAM role to a user group

You can bind RAM roles to user groups as needed.

Prerequisites

You must create a user group before RAM roles can be added. If no user groups have been created, see [Add a role](#).

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **User Groups** tab.
6. Click **Add User Group**. In the dialog box that appears, select a user group.
7. Click **OK**.

1.11.3.7. Grant permissions to a RAM role

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Select Existing Permission Policy**.
7. In the dialog box that appears, select a RAM permission policy and click **OK**. If no RAM permission policies are available, see [Add a permission policy](#).

1.11.3.8. Remove permissions from a RAM role

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.

2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Remove** in the **Actions** column corresponding to the permission policy that you want to remove.

1.11.3.9. Modify a RAM role name

Administrators can modify the names of RAM roles.

Context

 **Note** The name of a preset role cannot be modified.

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, click **More** in the **Actions** column corresponding to a RAM role, and choose **Modify** from the shortcut menu to go to the **Roles** page.
5. Move the pointer over the role name and click  to enter a new role name.

1.11.3.10. Delete a RAM role

This topic describes how to delete a RAM user.

Prerequisites

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

Procedure

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

1.11.4. RAM authorization policies

1.11.4.1. Create a RAM role

You can create authorization policies and grant them to organizations as needed.

Procedure

1. **Log on to the ASCM console** as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Roles**.
4. In the upper-right corner of the page, click **Create RAM User**.

5. On the **Create RAM User** page, set **Organization and Service**.
6. Click **OK**.

1.11.4.2. View the details of a RAM role

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Roles**.
4. On the **RAM Users** page, set **Role Name**, **Service Name**, or **Organization Name**, and click **Search** in the upper-right corner. To perform another search, click **Clear**.
5. Find the target RAM role and click **Details** in the **Actions** column.

1.11.4.3. View RAM authorization policies

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

Prerequisites

A RAM authorization policy is created. For more information, see [Create a RAM role](#).

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the top navigation bar, click **Configurations**.
3. In the left-side navigation pane of the **Configurations** page, click **RAM Roles**.
4. On the **RAM Users** page, set **Role Name** or **Service Name**, and click **Search** in the upper-right corner. To perform another search, click **Clear**.
5. Find the RAM role that you want to view and click **Details** in the **Actions** column.
6. Click the **Role Policy** tab to view information about the role authorization policy. Click **Details** in the **Actions** column to view the policy details.

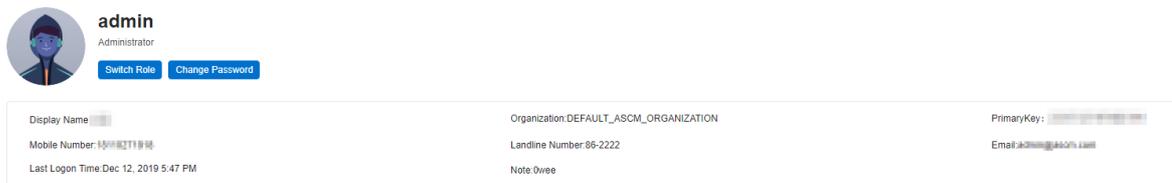
1.12. Personal information management

1.12.1. Modify personal information

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

Procedure

1. [Log on to the ASCM console](#) as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose **User Information** from the shortcut menu.



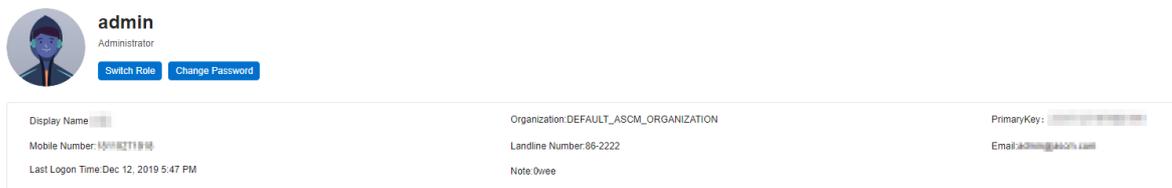
3. Click  next to the item you want to modify.
4. In the Modify User Information dialog box that appears, modify the relevant information.
5. Click OK.

1.12.2. Change your logon password

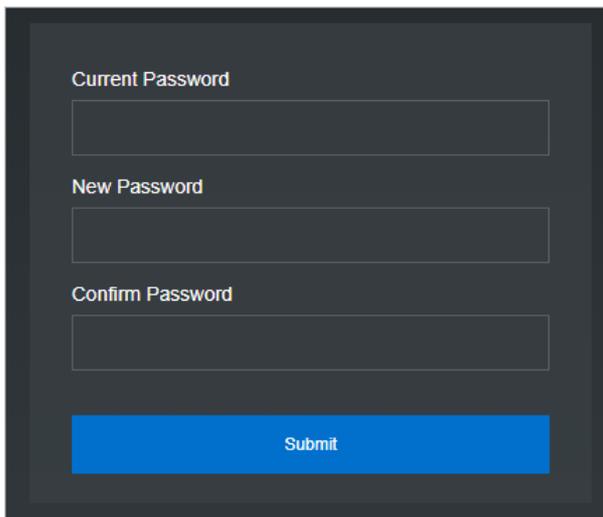
To improve security, you must change your logon password in a timely manner.

Procedure

1. Log on to the ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose User Information from the shortcut menu.



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



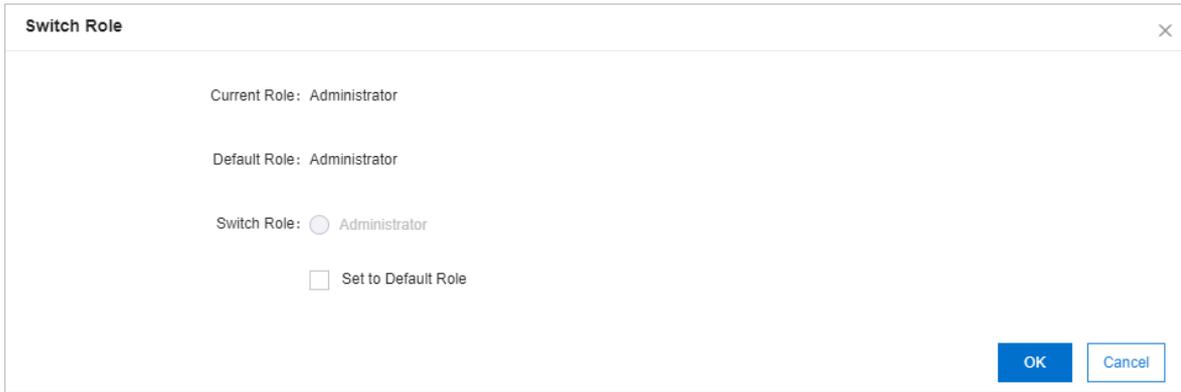
4. Click Submit.

1.12.3. Switch the current role

You can switch the scope of your current role.

Procedure

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



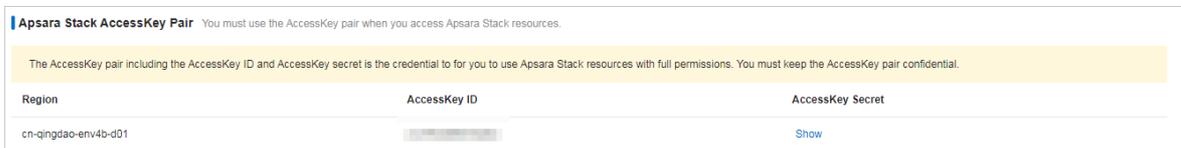
You can also switch back to the default role.

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

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

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the user profile picture and choose **User Information** from the shortcut menu.
3. In the **Apsara Stack AccessKey Pair** section, view your AccessKey pair.



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

2. Elastic Compute Service (ECS)

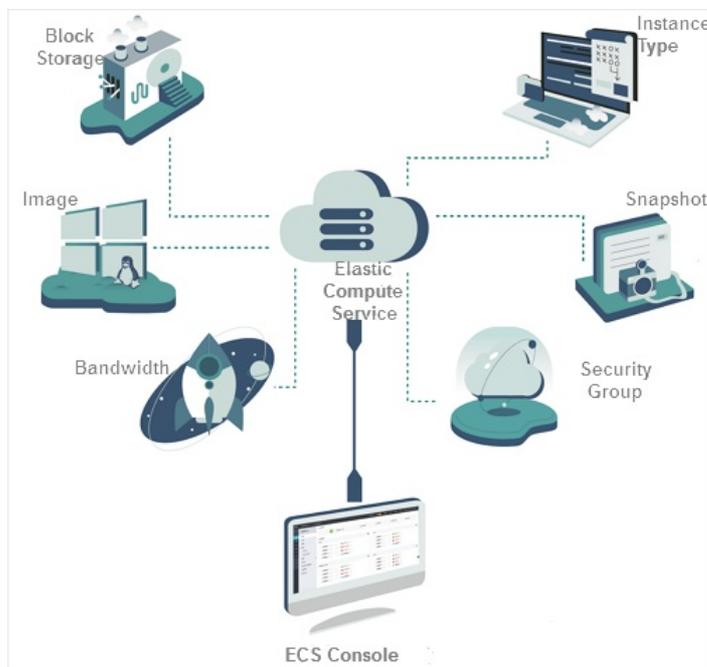
2.1. What is ECS?

2.1.1. Overview

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

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

ECS components



2.1.2. Instance types

An ECS instance is the smallest unit that can provide compute capabilities and services for your business. The compute capabilities of instances vary by instance type.

An ECS instance type defines the basic properties of ECS instances, such as CPU, clock speed, and memory. In addition to the instance type, you must also configure the Block Storage devices, image, and network type when you create an ECS instance. The following table describes instance families and lists the instance types of each instance family.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
n4	ecs.n4.small	1	2.0	N/A	0.5	50	1	1
	ecs.n4.large	2	4.0	N/A	0.5	100	1	1
	ecs.n4.xlarge	4	8.0	N/A	0.8	150	1	2
	ecs.n4.2xlarge	8	16.0	N/A	1.2	300	1	2
	ecs.n4.4xlarge	16	32.0	N/A	2.5	400	1	2
	ecs.n4.8xlarge	32	64.0	N/A	5.0	500	1	2
mn4	ecs.mn4.small	1	4.0	N/A	0.5	50	1	1
	ecs.mn4.large	2	8.0	N/A	0.5	100	1	1
	ecs.mn4.xlarge	4	16.0	N/A	0.8	150	1	2
	ecs.mn4.2xlarge	8	32.0	N/A	1.2	300	1	3
	ecs.mn4.4xlarge	16	64.0	N/A	2.5	400	1	8
	ecs.mn4.8xlarge	32	128.0	N/A	5.0	500	2	8
xn4	ecs.xn4.small	1	1.0	N/A	0.5	50	1	1
e4	ecs.e4.small	1	8.0	N/A	0.5	50	1	1
	ecs.e4.large	2	16.0	N/A	0.5	100	1	1
	ecs.e4.xlarge	4	32.0	N/A	0.8	150	1	2
	ecs.e4.2xlarge	8	64.0	N/A	1.2	300	1	3
	ecs.e4.4xlarge	16	128.0	N/A	2.5	400	1	8
	ecs.sn1ne.large	2	4.0	N/A	1.0	300	2	2

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
sn1ne	ecs.sn1ne.xlarge	4	8.0	N/A	1.5	500	2	3
	ecs.sn1ne.2xlarge	8	16.0	N/A	2.0	1,000	4	4
	ecs.sn1ne.3xlarge	12	24.0	N/A	2.5	1,300	4	6
	ecs.sn1ne.4xlarge	16	32.0	N/A	3.0	1,600	4	8
	ecs.sn1ne.6xlarge	24	48.0	N/A	4.5	2,000	6	4
	ecs.sn1ne.8xlarge	32	64.0	N/A	6.0	2,500	8	8
g6	ecs.g6.large	2	8.0	N/A	1.0	300	2	2
	ecs.g6.xlarge	4	16.0	N/A	1.5	500	4	3
	ecs.g6.2xlarge	8	32.0	N/A	2.5	800	8	4
	ecs.g6.3xlarge	12	48.0	N/A	4.0	900	8	6
	ecs.g6.4xlarge	16	64.0	N/A	5.0	1,000	8	8
	ecs.g6.6xlarge	24	96.0	N/A	7.5	1,500	12	8
	ecs.g6.8xlarge	32	128.0	N/A	10.0	2,000	16	8
g5	ecs.g5.large	2	8.0	N/A	1.0	300	2	2
	ecs.g5.xlarge	4	16.0	N/A	1.5	500	2	3
	ecs.g5.2xlarge	8	32.0	N/A	2.5	800	2	4
	ecs.g5.3xlarge	12	48.0	N/A	4.0	900	4	6
	ecs.g5.4xlarge	16	64.0	N/A	5.0	1,000	4	8
	ecs.g5.6xlarge	24	96.0	N/A	7.5	1,500	6	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
	ecs.g5.8xlarge	32	128.0	N/A	10.0	2,000	8	8
	ecs.g5.16xlarge	64	256.0	N/A	20.0	4,000	16	8
sn2ne	ecs.sn2ne.large	2	8.0	N/A	1.0	300	2	2
	ecs.sn2ne.xlarge	4	16.0	N/A	1.5	500	2	3
	ecs.sn2ne.2xlarge	8	32.0	N/A	2.0	1,000	4	4
	ecs.sn2ne.3xlarge	12	48.0	N/A	2.5	1,300	4	6
	ecs.sn2ne.4xlarge	16	64.0	N/A	3.0	1,600	4	8
	ecs.sn2ne.6xlarge	24	96.0	N/A	4.5	2,000	6	4
	ecs.sn2ne.8xlarge	32	128.0	N/A	6.0	2,500	8	8
	ecs.sn2ne.14xlarge	56	224.0	N/A	10.0	4,500	14	8
r6	ecs.r6.large	2	16.0	N/A	1.0	300	2	2
	ecs.r6.xlarge	4	32.0	N/A	1.5	500	4	3
	ecs.r6.2xlarge	8	64.0	N/A	2.5	800	8	4
	ecs.r6.3xlarge	12	96.0	N/A	4.0	900	8	6
	ecs.r6.4xlarge	16	128.0	N/A	5.0	1,000	8	8
	ecs.r6.6xlarge	24	192.0	N/A	7.5	1,500	12	8
	ecs.r6.8xlarge	32	256.0	N/A	10.0	2,000	16	8
	ecs.r5.large	2	16.0	N/A	1.0	300	2	2
	ecs.r5.xlarge	4	32.0	N/A	1.5	500	2	3

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
r5	ecs.r5.2xlarge	8	64.0	N/A	2.5	800	2	4
	ecs.r5.3xlarge	12	96.0	N/A	4.0	900	4	6
	ecs.r5.4xlarge	16	128.0	N/A	5.0	1,000	4	8
	ecs.r5.6xlarge	24	192.0	N/A	7.5	1,500	6	8
	ecs.r5.8xlarge	32	256.0	N/A	10.0	2,000	8	8
	ecs.r5.16xlarge	64	512.0	N/A	20.0	4,000	16	8
se1ne	ecs.se1ne.large	2	16.0	N/A	1.0	300	2	2
	ecs.se1ne.xlarge	4	32.0	N/A	1.5	500	2	3
	ecs.se1ne.2xlarge	8	64.0	N/A	2.0	1,000	4	4
	ecs.se1ne.3xlarge	12	96.0	N/A	2.5	1,300	4	6
	ecs.se1ne.4xlarge	16	128.0	N/A	3.0	1,600	4	8
	ecs.se1ne.6xlarge	24	192.0	N/A	4.5	2,000	6	4
	ecs.se1ne.8xlarge	32	256.0	N/A	6.0	2,500	8	8
	ecs.se1ne.14xlarge	56	480.0	N/A	10.0	4,500	14	8
se1	ecs.se1.large	2	16.0	N/A	0.5	100	1	2
	ecs.se1.xlarge	4	32.0	N/A	0.8	200	1	3
	ecs.se1.2xlarge	8	64.0	N/A	1.5	400	1	4
	ecs.se1.4xlarge	16	128.0	N/A	3.0	500	2	8
	ecs.se1.8xlarge	32	256.0	N/A	6.0	800	3	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
	ecs.se1.14xlarge	56	480.0	N/A	10.0	1,200	4	8
ebmg5s	ecs.ebmg5s.24xlarge	96	384.0	N/A	30.0	4,500	8	32
ebmg5	ecs.ebmg5.24xlarge	96	384.0	N/A	10.0	4,000	8	32
i2	ecs.i2.xlarge	4	32.0	1 × 894	1.0	500	2	3
	ecs.i2.2xlarge	8	64.0	1 × 1,788	2.0	1,000	2	4
	ecs.i2.4xlarge	16	128.0	2 × 1,788	3.0	1,500	4	8
	ecs.i2.8xlarge	32	256.0	4 × 1,788	6.0	2,000	8	8
	ecs.i2.16xlarge	64	512.0	8 × 1,788	10.0	4,000	16	8
d1	ecs.d1.2xlarge	8	32.0	4 × 5,500	3.0	300	1	4
	ecs.d1.3xlarge	12	48.0	6 × 5,500	4.0	400	1	6
	ecs.d1.4xlarge	16	64.0	8 × 5,500	6.0	600	2	8
	ecs.d1.6xlarge	24	96.0	12 × 5,500	8.0	800	2	8
	ecs.d1-c8d3.8xlarge	32	128.0	12 × 5,500	10.0	1,000	4	8
	ecs.d1.8xlarge	32	128.0	16 × 5,500	10.0	1,000	4	8
	ecs.d1-c14d3.14xlarge	56	160.0	12 × 5,500	17.0	1,800	6	8
	ecs.d1.14xlarge	56	224.0	28 × 5,500	17.0	1,800	6	8
	ecs.d2-zyy-d0.4xlarge	16	64.0	N/A	3.0	300	2	8
	ecs.d2-zyy-d0.6xlarge	24	96.0	N/A	4.0	400	2	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
d2	ecs.d2-zyy.4xlarge	16	64.0	6 × 7,500	3.0	300	4	8
	ecs.d2-zyy.6xlarge	24	96.0	12 × 7,500	4.0	400	4	8
	ecs.d2-zyy.7xlarge	56	224.0	12 × 7,300	17.0	1,800	6	8
	ecs.d2-zyy.8xlarge	32	160.0	12 × 7,300	17.0	1,800	6	8
	ecs.d2-zyy.22xlarge	88	352.0	12 × 7,300	17.0	1,800	6	8
	ecs.d2-zyy-m40.8xlarge	32	128.0	12 × 7,500	6.0	600	4	8
	ecs.d2-gab.4xlarge	16	64.0	6 × 1,150	3.0	300	4	8
	ecs.d2-gab.8xlarge	32	128.0	12 × 1,150	6.0	600	4	8
sccg5ib	ecs.sccg5ib.24xlarge	96	384.0	N/A	10.0	4,500	8	32
scch5ib	ecs.scch5ib.16xlarge	64	192.0	N/A	10.0	4,500	8	32
c6	ecs.c6.large	2	4.0	N/A	1.0	300	2	2
	ecs.c6.xlarge	4	8.0	N/A	1.5	500	4	3
	ecs.c6.2xlarge	8	16.0	N/A	2.5	800	8	4
	ecs.c6.3xlarge	12	24.0	N/A	4.0	900	8	6
	ecs.c6.4xlarge	16	32.0	N/A	5.0	1,000	8	8
	ecs.c6.6xlarge	24	48.0	N/A	7.5	1,500	12	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
	ecs.c6.8xlarge	32	64.0	N/A	10.0	2,000	16	8
sn1	ecs.sn1.medium	2	4.0	N/A	0.5	100	1	2
	ecs.sn1.large	4	8.0	N/A	0.8	200	1	3
	ecs.sn1.xlarge	8	16.0	N/A	1.5	400	1	4
	ecs.sn1.3xlarge	16	32.0	N/A	3.0	500	2	8
	ecs.sn1.7xlarge	32	64.0	N/A	6.0	800	3	8
sn2	ecs.sn2.medium	2	8.0	N/A	0.5	100	1	2
	ecs.sn2.large	4	16.0	N/A	0.8	200	1	3
	ecs.sn2.xlarge	8	32.0	N/A	1.5	400	1	4
	ecs.sn2.3xlarge	16	64.0	N/A	3.0	500	2	8
	ecs.sn2.7xlarge	32	128.0	N/A	6.0	800	3	8
	ecs.sn2.14xlarge	56	224.0	N/A	10.0	1,200	4	8

The following table describes FPGA-accelerated instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	FPGA
	ecs.f1-c8f1.2xlarge	8	60.0	N/A	3.0	400	4	2	Intel ARRIA 10 GX 1150
	ecs.f1-c8f1.4xlarge	16	120.0	N/A	5.0	1,000	4	2	2 × Intel ARRIA 10 GX 1150

f1 Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	FPGA
f1	ecs.f1-c28f1.7xlarge	28	112.0	N/A	5.0	2,000	8	2	Intel ARRIA 10 GX 1150
	ecs.f1-c28f1.14xlarge	56	224.0	N/A	10.0	2,000	14	2	2 × Intel ARRIA 10 GX 1150
f3	ecs.f3-c16f1.4xlarge	16	64.0	N/A	5.0	1,000	4	8	1 × Xilinx VU9P
	ecs.f3-c16f1.8xlarge	32	128.0	N/A	10.0	2,000	8	8	2 × Xilinx VU9P
	ecs.f3-c16f1.16xlarge	64	256.0	N/A	20.0	2,000	16	8	4 × Xilinx VU9P

The following table describes GPU-accelerated instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	GPU
gn5	ecs.gn5-c4g1.xlarge	4	30.0	440	3.0	300	1	3	1 × NVIDIA P100
	ecs.gn5-c8g1.2xlarge	8	60.0	440	3.0	400	1	4	1 × NVIDIA P100
	ecs.gn5-c4g1.2xlarge	8	60.0	880	5.0	1,000	2	4	2 × NVIDIA P100
	ecs.gn5-c8g1.4xlarge	16	120.0	880	5.0	1,000	4	8	2 × NVIDIA P100
	ecs.gn5-c28g1.7xlarge	28	112.0	440	5.0	1,000	8	8	1 × NVIDIA P100
	ecs.gn5-c8g1.8xlarge	32	240.0	1760	10.0	2,000	8	8	4 × NVIDIA P100
	ecs.gn5-c28g1.14xlarge	56	224.0	880	10.0	2,000	14	8	2 × NVIDIA P100

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	GPU
	ecs.gn5-c8g1.14xlarge	54	480.0	3520	25.0	4,000	14	8	8 × NVIDIA P100
gn4	ecs.gn4-c4g1.xlarge	4	30.0	N/A	3.0	300	1	3	1 × NVIDIA M40
	ecs.gn4-c8g1.2xlarge	8	30.0	N/A	3.0	400	1	4	1 × NVIDIA M40
	ecs.gn4.8xlarge	32	48.0	N/A	6.0	800	3	8	1 × NVIDIA M40
	ecs.gn4-c4g1.2xlarge	8	60.0	N/A	5.0	500	1	4	2 × NVIDIA M40
	ecs.gn4-c8g1.4xlarge	16	60.0	N/A	5.0	500	1	8	2 × NVIDIA M40
	ecs.gn4.14xlarge	56	96.0	N/A	10.0	1,200	4	8	2 × NVIDIA M40
ga1	ecs.ga1.xlarge	4	10.0	1 × 87	1.0	200	1	3	0.25 × AMD S7150
	ecs.ga1.2xlarge	8	20.0	1 × 175	1.5	300	1	4	0.5 × AMD S7150
	ecs.ga1.4xlarge	16	40.0	1 × 350	3.0	500	2	8	1 × AMD S7150
	ecs.ga1.8xlarge	32	80.0	1 × 700	6.0	800	3	8	2 × AMD S7150
	ecs.ga1.14xlarge	56	160.0	1 × 1,400	10.0	1,200	4	8	4 × AMD S7150
	ecs.gn5i-c2g1.large	2	8.0	N/A	1.0	100	2	2	1 × NVIDIA P4
	ecs.gn5i-c4g1.xlarge	4	16.0	N/A	1.5	200	2	3	1 × NVIDIA P4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	GPU
gn5i	ecs.gn5i-c8g1.2xlarge	8	32.0	N/A	2.0	400	4	4	1 × NVIDIA P4
	ecs.gn5i-c16g1.4xlarge	16	64.0	N/A	3.0	800	4	8	1 × NVIDIA P4
	ecs.gn5i-c16g1.8xlarge	32	128.0	N/A	6.0	1,200	8	8	2 × NVIDIA P4
	ecs.gn5i-c24g1.12xlarge	48	192.0	N/A	10.0	2,000	8	8	2 × NVIDIA P4
	ecs.gn5i-c28g1.14xlarge	56	224.0	N/A	10.0	2,000	14	8	2 × NVIDIA P4
gn5e	ecs.gn5e-c11g1.3xlarge	10	58.0	N/A	2.0	150	1	6	1 × NVIDIA P4
	ecs.gn5e-c11g1.5xlarge	22	116.0	N/A	4.0	300	1	8	2 × NVIDIA P4
	ecs.gn5e-c11g1.11xlarge	44	232.0	N/A	6.0	600	2	8	4 × NVIDIA P4
	ecs.gn5e-c11g1.22xlarge	88	464.0	N/A	10.0	1,200	4	8	8 × NVIDIA P4
	ecs.gn6i-c10g1.2xlarge	10	42.0	N/A	5.0	800	2	4	1 × T4
	ecs.gn6i-c10g1.5xlarge	20	84.0	N/A	8.0	1,000	4	6	2 × T4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	GPU
gn6i	ecs.gn6i-c10g1.10xlarge	40	168.0	N/A	15.0	2,000	8	8	4 × T4
	ecs.gn6i-c10g1.20xlarge	80	336.0	N/A	30.0	4,000	16	8	8 × T4
	ecs.gn6i-c14g1.3xlarge	14	56.0	N/A	5.0	1,000	4	6	1 × T4
	ecs.gn6i-c14g1.7xlarge	28	112.0	N/A	10.0	2,000	8	8	2 × T4
	ecs.gn6i-c14g1.14xlarge	56	224.0	N/A	20.0	4,000	12	8	4 × T4
	ecs.gn6i-c20g1.5xlarge	20	80.0	N/A	10.0	1,500	4	6	1 × T4
	ecs.gn6i-c20g1.10xlarge	40	160.0	N/A	20.0	3,000	8	8	2 × T4
gn6v	ecs.gn6v-c8g1.2xlarge	8	32.0	N/A	2.5	800	4	4	1 × NVIDIA V100
	ecs.gn6v-c8g1.8xlarge	32	128.0	N/A	10.0	2,000	8	8	4 × NVIDIA V100
	ecs.gn6v-c8g1.16xlarge	64	256.0	N/A	20.0	2,500	16	8	8 × NVIDIA V100
sccgn6p	ecs.sccgn6p.24xlarge	96	768.0	N/A	30.0	4,500	8	32	8 × NVIDIA V100

The following table describes shared instance families that support IPv6.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	IPv6 support	NIC queues	ENIs (including one primary ENI)
n4v2	ecs.n4v2.small	1	2.0	N/A	0.5	50	Yes	1	1
	ecs.n4v2.large	2	4.0	N/A	0.5	100	Yes	1	1
	ecs.n4v2.xlarge	4	8.0	N/A	0.8	150	Yes	1	2
	ecs.n4v2.2xlarge	8	16.0	N/A	1.2	300	Yes	1	2
	ecs.n4v2.4xlarge	16	32.0	N/A	2.5	400	Yes	1	2
	ecs.n4v2.8xlarge	32	64.0	N/A	5.0	500	Yes	1	2
mn4v2	ecs.mn4v2.small	1	4.0	N/A	0.5	50	Yes	1	1
	ecs.mn4v2.large	2	8.0	N/A	0.5	100	Yes	1	1
	ecs.mn4v2.xlarge	4	16.0	N/A	0.8	150	Yes	1	2
	ecs.mn4v2.2xlarge	8	32.0	N/A	1.2	300	Yes	1	2
	ecs.mn4v2.4xlarge	16	64.0	N/A	2.5	400	Yes	1	2
	ecs.mn4v2.8xlarge	32	128.0	N/A	5.0	500	Yes	2	8
xn4v2	ecs.xn4v2.small	1	1.0	N/A	0.5	50	Yes	1	1
e4v2	ecs.e4v2.small	1	8.0	N/A	0.5	50	Yes	1	1
	ecs.e4v2.large	2	16.0	N/A	0.5	100	Yes	1	1
	ecs.e4v2.xlarge	4	32.0	N/A	0.8	150	Yes	1	2
	ecs.e4v2.2xlarge	8	64.0	N/A	1.2	300	Yes	1	3
	ecs.e4v2.4xlarge	16	128.0	N/A	2.5	400	Yes	1	8

The following table describes burstable instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Baseline CPU computing performance	CPU credits per hour	Max CPU credit balance	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
t5	ecs.t5- lc2m1.nano	1	0.5	20%	12	288	N/A	0.1	40	1	1
	ecs.t5- lc1m1.small	1	1.0	20%	12	288	N/A	0.2	60	1	1
	ecs.t5- lc1m2.small	1	2.0	20%	12	288	N/A	0.2	60	1	1
	ecs.t5- lc1m2.large	2	4.0	20%	24	576	N/A	0.4	100	1	1
	ecs.t5- lc1m4.large	2	8.0	20%	24	576	N/A	0.4	100	1	1
	ecs.t5- c1m1.large	2	2.0	25%	30	720	N/A	0.5	100	1	1
	ecs.t5- c1m2.large	2	4.0	25%	30	720	N/A	0.5	100	1	1
	ecs.t5- c1m4.large	2	8.0	25%	30	720	N/A	0.5	100	1	1
	ecs.t5- c1m1.xlarge	4	4.0	25%	60	1,440	N/A	0.8	200	1	2
	ecs.t5- c1m2.xlarge	4	8.0	25%	60	1,440	N/A	0.8	200	1	2
	ecs.t5- c1m4.xlarge	4	16.0	25%	60	1,440	N/A	0.8	200	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Baseline CPU computing performance	CPU credits per hour	Max CPU credit balance	Local storage (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)
	ecs.t5-c1m1.2xlarge	8	8.0	25%	120	2,880	N/A	1.2	400	1	2
	ecs.t5-c1m2.2xlarge	8	16.0	25%	120	2,880	N/A	1.2	400	1	2
	ecs.t5-c1m4.2xlarge	8	32.0	25%	120	2,880	N/A	1.2	400	1	2
	ecs.t5-c1m1.4xlarge	16	16.0	25%	240	5,760	N/A	1.2	600	1	2
	ecs.t5-c1m2.4xlarge	16	32.0	25%	240	5,760	N/A	1.2	600	1	2

The following table describes the ecs.anyshare custom instance type.

Instance family	vCPUs (x)	Memory (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	Disk bandwidth (Gbit/s)
Custom instance type ecs.anyshare	$0 < x \leq 2$	1 to 16	0.5	100	1	2	0.5
	$2 < x \leq 4$	2 to 32	0.8	$100 + (x - 2)/0.2$	1	2	0.8
	$4 < x \leq 8$	4 to 64	$0.8 + (x - 5)/4$	$200 + (x - 4)/0.2$	1	3	$0.8 + (x - 5)/4$
	$8 < x \leq 12$	8 to 96	$1.5 + (x - 8)/8$	$400 + (x - 8)/0.8$	2	3	$1.5 + (x - 8)/4$
	$12 < x \leq 16$	12 to 128	$2 + (x - 12)/4$	$450 + (x - 12)/0.8$	3	4	1
	$16 < x \leq 24$	16 to 196	$3 + (x - 16)/8$	$500 + (x - 16)/0.8$	3	5	$1 + (x - 16)/8$

Instance family	vCPUs (x)	Memory (GiB)	Bandwidth (Gbit/s)	Packet forwarding rate (Kpps)	NIC queues	ENIs (including one primary ENI)	Disk bandwidth (Gbit/s)
	$24 < x \leq 32$	24 to 256	$4 + (x - 24)/8$	$600 + (x - 24)/0.4$	4	6	$2 + (x - 24)/8$
	$x > 32$	32 to 352	$\min(5 + (x - 32)/8, 10)$	$\min(800 + (x - 32)/0.8, 1200)$	4	8	$\min(3 + (x - 32)/8, 8)$

The following instance types are applicable only in environments that are upgraded from Apsara Stack V2 to V3.

Instance family	Instance type	vCPUs	Memory (GiB)
n1	ecs.n1.tiny	1	1.0
	ecs.n1.small	1	2.0
	ecs.n1.medium	2	4.0
	ecs.n1.large	4	8.0
	ecs.n1.xlarge	8	16.0
	ecs.n1.3xlarge	16	32.0
	ecs.n1.7xlarge	32	64.0
n2	ecs.n2.small	1	4.0
	ecs.n2.medium	2	8.0
	ecs.n2.large	4	16.0
	ecs.n2.xlarge	8	32.0
	ecs.n2.3xlarge	16	64.0
	ecs.n2.7xlarge	32	128.0
e3	ecs.e3.small	1	8.0
	ecs.e3.medium	2	16.0
	ecs.e3.large	4	32.0
	ecs.e3.xlarge	8	64.0
	ecs.e3.3xlarge	16	128.0
c1	ecs.c1.small	8	8.0
	ecs.c1.large	8	16.0
c2	ecs.c2.medium	16	16.0
	ecs.c2.large	16	32.0

Instance family	Instance type	vCPUs	Memory (GiB)
	ecs.c2.xlarge	16	64.0
m1	ecs.m1.medium	4	16.0
	ecs.m1.xlarge	8	32.0
m2	ecs.m2.medium	4	32.0
s1	ecs.s1.small	1	2.0
	ecs.s1.medium	1	4.0
	ecs.s1.large	1	8.0
s2	ecs.s2.small	2	2.0
	ecs.s2.large	2	4.0
	ecs.s2.xlarge	2	8.0
	ecs.s2.2xlarge	2	16.0
s3	ecs.s3.medium	4	4.0
	ecs.s3.large	4	8.0
t1	ecs.t1.small	1	1.0

2.1.3. Instance lifecycle

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

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

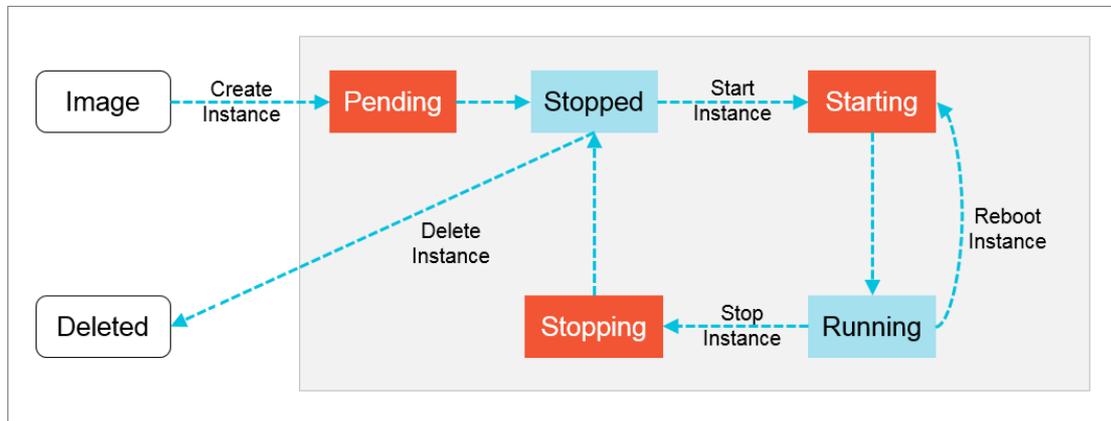
Instance states

State	State attribute	Description	State in an API response
Instance being created	Intermediate	The instance is being created and waiting to be started. If an instance remains in the Instance being created state for an extended period of time, an exception has occurred.	Pending
Starting	Intermediate	When you start or restart an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Starting state for an extended period of time, an exception has occurred.	Starting
Running	Stable	While an instance is in the Running state, the instance can function normally and can accommodate your business needs.	Running

State	State attribute	Description	State in an API response
Stopping	Intermediate	When you stop an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Stopped state. If an instance remains in the Stopping state for an extended period of time, an exception has occurred.	Stopping
Stopped	Stable	An instance enters this state when it is stopped. An instance in the Stopped state cannot provide external services.	Stopped
Reinitializing	Intermediate	When you re-initialize the system disk or a data disk of an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Reinitializing state for an extended period of time, an exception has occurred.	Stopped
Changing system disk	Intermediate	When you replace the system disk of an instance by using the ECS console or calling an API operation, the instance enters the Changing system disk state before it enters the Running state. If an instance remains in the Changing system disk state for an extended period of time, an exception has occurred.	Stopped

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

Transitions between instance states in API responses



2.2. Instructions

2.2.1. Restrictions

Learn about restrictions before performing operations on ECS instances.

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

2.2.2. Suggestions

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

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

2.2.3. Limits

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

General limits

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

Instance family ga1

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

- Ubuntu 16.04 with an AMD GPU driver pre-installed
- Windows Server 2016 English version with an AMD GPU driver pre-installed
- Windows Server 2008 R2 English version with an AMD GPU driver pre-installed

Note:

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

 **Note** This operation causes data loss.

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

Instance families gn4, gn5i, and gn5

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

2.2.4. Notice for Windows users

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

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

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

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

2.2.5. Notice for Linux users

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

- Do not modify content of the default /etc/issue files under a Linux instance. Otherwise, the custom image created from the instance cannot be recognized, and instances created based on the image cannot start as expected.
- Do not arbitrarily modify the permissions of each directory in the partition where the root directory is located, especially permissions of /etc, /sbin, /bin, /boot, /dev, /usr, and /lib directories. Improper modification of permissions can cause errors.
- Do not rename, delete, or disable Linux root accounts.
- Do not compile or perform any other operations on the Linux kernel.
- We do not recommend the use of Swap for partitioning.
- Do not enable the NetWorkManager service. This service conflicts with the internal network service of the system, causing network errors.

2.2.6. Notice on defense against DDoS attacks

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

2.3. Quick start

2.3.1. Overview

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

Perform the following procedure:

1. **Create a security group**

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

2. **Create an instance**

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

3. **Connect to an instance**

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

2.3.2. Log on to the ECS console

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

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.

2.3.3. Create a security group

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

Prerequisites

A Virtual Private Cloud (VPC) has been created. For more information, see *VPC User Guide*.

Context

Instances that belong to the same account and are in the same region and in the same security group can communicate with each other over the internal network. If instances that belong to the same account in the same region are in different security groups, you can implement internal network communication by authorizing mutual access between two security groups.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **New Security Group**.
5. Configure the parameters of the security group.

Type	Parameter	Required	Description
------	-----------	----------	-------------

Type	Parameter	Required	Description
Region	Organization	Yes	The organization to which the security group belongs. Make sure that the security group and the VPC belong to the same organization.
	Resource Set	Yes	The resource set to which the security group belongs. Make sure that the security group and the VPC belong to the same resource set.
	Region	Yes	The region to which the security group belongs. Make sure that the security group and the VPC belong to the same region.
	Zone	Yes	The ID of the zone where the security group resides.
Basic Settings	VPC	Yes	The VPC to which the security group belongs.
	Security Group Name	No	The name must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It cannot start with http:// or https://.
	Description	No	The description of the security group. We recommend that you provide an informational description to simplify future management operations. The name must be 2 to 256 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and commas (,). It cannot start with http:// or https://.

6. Click **Submit**.

2.3.4. Create an instance

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

Prerequisites

- A VPC and a VSwitch are created. For more information, see *Create a VPC and Create a VSwitch* in Apsara Stack VPC User Guide.
- If you want to assign an IPv6 address to the instance that you want to create, make sure that the VPC and VSwitch are associated with IPv6 CIDR blocks. For more information, see *Enable an IPv6 CIDR block for a VPC and Enable an IPv6 CIDR block for a VSwitch* in Apsara Stack VPC User Guide.
- A security group is available. If no security group is available, create a security group. For more information, see [Create a security group](#).

Context

Some limits apply when you create GPU-accelerated instances. For more information, see [Limits](#).

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. Click **Create Instance**.
4. Configure parameters listed in the following tables to create an instance.
 - i. Configure the basic settings of the instance.

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

- ii. Select a region and zone for the instance.

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

iii. Configure the network of the instance.

Parameter	Required	Description
Network Type	Yes	Select the type of the network in which to create the instance. VPC is available.
VPC	Yes	Select a VPC in which to create the instance.
VSwitch	Yes	Select a VSwitch to be connected to the instance.
Private IP	No	Specify a private IP address for the instance. The private IP address must be within the CIDR block of the VSwitch. If you do not specify a private IP address, the system will automatically allocate a private IP address to the instance.

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

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

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

Parameter	Required	Description
Instance Family	Yes	Select an instance family for the instance. After you select an instance family, you must select an instance type.
Instance Type	Yes	Select an instance type. Windows Server images require specific CPU and memory combinations. For more information, see Limits in <i>ECS Product Introduction</i> .

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

Parameter	Required	Description
Image Type	Yes	Select an image type. Valid values: Public Image and Custom Image .

Parameter	Required	Description
Public Image	Subject to the image type	<p>Select a public image for the instance. Public images provided by Alibaba Cloud are licensed, secure, and stable. Public images include Windows Server images and major Linux images.</p> <p>This parameter must be specified when you set Image Type to Public Image.</p> <p>When you use an image that supports DHCPv6 to create an instance, an IPv6 address is automatically assigned to the instance. The created instance can use this IPv6 address to communicate over the internal network. When you use an image that does not support DHCPv6 to create an instance, you must manually assign an IPv6 address to the instance. The following images support DHCPv6:</p> <ul style="list-style-type: none"> ▪ Linux images: <ul style="list-style-type: none"> ▪ CentOS 7.6 IPv6 64-bit ▪ CentOS 6.10 64-bit ▪ SUSE Linux Enterprise Server 12 SP4 64-bit ▪ Windows Server images <p> Note To use an IPv6 address to communicate over the Internet, you must also enable public bandwidth for the IPv6 address. For more information, see <i>Enable Internet bandwidth for an IPv6 address in Apsara Stack V PC User Guide</i>.</p>
Custom Image	Subject to the image type	<p>Select a custom image for the instance. Custom images are created from instances or snapshots, or imported from your local device.</p> <p>This parameter must be specified when you set Image Type to Custom Image.</p>

viii. Configure the storage settings for the instance.

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

ix. Configure the logon password settings for the instance.

Parameter	Required	Description
Set Password	Yes	Specify when to set the password. Valid values: Now and Later . If Later is selected, you can use the password reset feature to set a password at a later time. For more information, see . <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note The password is used to log on to the instance, not to the VNC. </div>
Logon Password	No	Set the password to be used to log on to the instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include <code>()`~!@# \$ % ^ & * - _ + = { } [] : ; ' < > , . ? /</code>
Confirm Password	No	Re-enter the password.

- x. (Optional) Select a deployment set in which to create the instance.
- xi. (Optional) Enter a name for the instance. The name must be 2 to 128 characters in length and start with a letter. It can contain periods (.), underscores (_), colons (:), and hyphens (-).
If you do not specify a name, the system will assign an instance name at random.
- xii. (Optional) In the **User Data** field, enter the user data to be automatically run upon instance startup. Windows supports **Batch** and **PowerShell** scripts. Before you perform Base64 encoding, make sure that the content to be encoded includes `[bat]` or `[powershell]` as the first line. Linux supports shell scripts.
- xiii. Enter the number of instances that you want to create. The number must be an integer ranging from 1 to 100.

5. Click **Submit**.

Result

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

2.3.5. Connect to an instance

2.3.5.1. Instance connecting overview

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

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

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

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

2.3.5.2. Connect to a Linux instance by using SSH commands in

Linux or Mac OS X

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

Prerequisites

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

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

Procedure

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

```
ssh root@instance IP
```

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

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

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

Prerequisites

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

Procedure

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

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

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

2.3.5.4. Connect to a Windows instance by using RDP

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

Prerequisites

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

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

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

Procedure

1. Activate the Remote Desktop Connection feature by using any of the following methods:
 - Click **Start**, enter `mstsc` in the search box, and click `mstsc` in the search result.
 - Press **Windows Key + R**. In the Run dialog box that appears, enter `mstsc` and click **OK**.
2. In the Remote Desktop Connection dialog box, enter the EIP of the instance and click **Show Options**.
3. Enter the username. The default username is `administrator`.
4. (Optional) If you do not want to enter the password upon subsequent logons, select **Allow me to save credentials**.
5. Click **Connect**.

6. In the **Windows Security** dialog box that appears, enter the password for the account and click **OK**.

Result

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

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

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

2.3.5.5. Connect to an ECS instance by using the VNC

You can access your instance by using the VNC in the ECS console when other SSH clients such as PuTTY, Xshell, and SecureCRT do not work properly.

Prerequisites

- The instance is in the **Running** state.
- The root certificate is imported to your web browser. For more information, see [Install a certificate](#).
- If you log on to an instance for the first time after the instance is created, make sure that you set a new VNC password. For more information, see [VNC Password](#).

Context

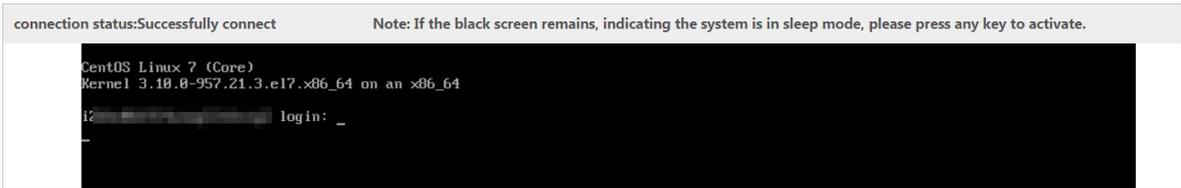
The VNC password is used to log on to the VNC of the ECS console, and the instance password is used to log on to an instance.

You can use the VNC to connect to an instance to solve issues shown in the following table.

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

Procedure

1. **Log on to the ECS console**.
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the ECS instance, and click **Remote Connection** in the **Actions** column.
5. Enter the VNC password, and then click **OK**. After the connection is successful, the logon page is displayed, as shown in the following figure.

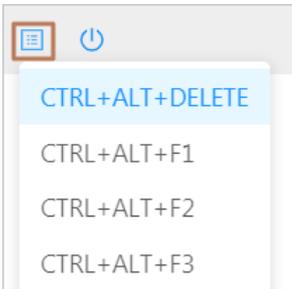


6. Enter the username and password.

- For Linux instances: Enter the username *root* and the logon password.

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

- For Windows instances: To use key combinations such as Ctrl + Alt + Delete, click the corresponding key combination in the upper-right corner of the VNC page.



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

2.4. Instances

2.4.1. Create an instance

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

Prerequisites

- A VPC and a VSwitch are created. For more information, see *Create a VPC and Create a VSwitch* in Apsara Stack VPC User Guide.
- If you want to assign an IPv6 address to the instance that you want to create, make sure that the VPC and VSwitch are associated with IPv6 CIDR blocks. For more information, see *Enable an IPv6 CIDR block for a VPC and Enable an IPv6 CIDR block for a VSwitch* in Apsara Stack VPC User Guide.
- A security group is available. If no security group is available, create a security group. For more information, see [Create a security group](#).

Context

Some limits apply when you create GPU-accelerated instances. For more information, see [Limits](#).

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. Click **Create Instance**.
4. Configure parameters listed in the following tables to create an instance.

i. Configure the basic settings of the instance.

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

ii. Select a region and zone for the instance.

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

iii. Configure the network of the instance.

Parameter	Required	Description
Network Type	Yes	Select the type of the network in which to create the instance. VPC is available.
VPC	Yes	Select a VPC in which to create the instance.
VSwitch	Yes	Select a VSwitch to be connected to the instance.
Private IP	No	Specify a private IP address for the instance. The private IP address must be within the CIDR block of the VSwitch. If you do not specify a private IP address, the system will automatically allocate a private IP address to the instance.

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

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

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

Parameter	Required	Description
Instance Family	Yes	Select an instance family for the instance. After you select an instance family, you must select an instance type.
Instance Type	Yes	Select an instance type. Windows Server images require specific CPU and memory combinations. For more information, see Limits in <i>ECS Product Introduction</i> .

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

Parameter	Required	Description
Image Type	Yes	Select an image type. Valid values: Public Image and Custom Image .
Public Image	Subject to the image type	<p>Select a public image for the instance. Public images provided by Alibaba Cloud are licensed, secure, and stable. Public images include Windows Server images and major Linux images.</p> <p>This parameter must be specified when you set Image Type to Public Image.</p> <p>When you use an image that supports DHCPv6 to create an instance, an IPv6 address is automatically assigned to the instance. The created instance can use this IPv6 address to communicate over the internal network. When you use an image that does not support DHCPv6 to create an instance, you must manually assign an IPv6 address to the instance. The following images support DHCPv6:</p> <ul style="list-style-type: none"> ▪ Linux images: <ul style="list-style-type: none"> ▪ CentOS 7.6 IPv6 64-bit ▪ CentOS 6.10 64-bit ▪ SUSE Linux Enterprise Server 12 SP4 64-bit ▪ Windows Server images <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note To use an IPv6 address to communicate over the Internet, you must also enable public bandwidth for the IPv6 address. For more information, see <i>Enable Internet bandwidth for an IPv6 address</i> in <i>Apsara Stack V PC User Guide</i>.</p> </div>
Custom Image	Subject to the image type	<p>Select a custom image for the instance. Custom images are created from instances or snapshots, or imported from your local device.</p> <p>This parameter must be specified when you set Image Type to Custom Image.</p>

viii. Configure the storage settings for the instance.

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

ix. Configure the logon password settings for the instance.

Parameter	Required	Description
Set Password	Yes	Specify when to set the password. Valid values: Now and Later . If Later is selected, you can use the password reset feature to set a password at a later time. For more information, see . <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> ? Note The password is used to log on to the instance, not to the VNC. </div>
Logon Password	No	Set the password to be used to log on to the instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include () ` ~ ! @ # \$ % ^ & * - _ + = { } [] ; ' < > , . ? /
Confirm Password	No	Re-enter the password.

- x. (Optional) Select a deployment set in which to create the instance.
- xi. (Optional) Enter a name for the instance. The name must be 2 to 128 characters in length and start with a letter. It can contain periods (.), underscores (_), colons (:), and hyphens (-).
If you do not specify a name, the system will assign an instance name at random.
- xii. (Optional) In the **User Data** field, enter the user data to be automatically run upon instance startup. Windows supports **Batch** and **PowerShell** scripts. Before you perform Base64 encoding, make sure that the content to be encoded includes `[bat]` or `[powershell]` as the first line. Linux supports shell scripts.
- xiii. Enter the number of instances that you want to create. The number must be an integer ranging from 1 to 100.

5. Click **Submit**.

Result

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

2.4.2. Connect to an instance

2.4.2.1. Instance connecting overview

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

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

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

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

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

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

Prerequisites

Create a security group and an instance.

Procedure

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

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

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

Prerequisites

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

Procedure

1. Download and install PuTTY for Windows.
2. Start the PuTTY client and complete the following settings:
 - **Host Name (or IP Address):** Enter the EIP of the instance to be connected.
 - **Port:** Select the default port 22.
 - **Connection Type:** Select SSH.

- **Saved Session:** Enter the name of the session. Click **Save**. After the settings are saved, PuTTY remembers the name and IP address of the instance. This eliminates the need to enter them every time you connect to the instance.
3. Click **Open** to connect to the instance. When you connect to the instance for the first time, PuTTY displays security alerts. Click **Yes** to proceed.
 4. Enter the username `root` and press **Enter**.
 5. Enter the password for the instance and press **Enter**.

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

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

2.4.2.4. Connect to a Windows instance by using RDC

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

Prerequisites

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

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

Procedure

1. Use one of the following methods to enable RDC:
 - Click **Start**, enter `mstsc` in the search box, and click `mstsc` in the search result.
 - Press the Windows logo key+R. In the Run dialog box that appears, enter `mstsc` and click **OK**.
2. In the **Remote Desktop Connection** dialog box, enter the Elastic IP address of the instance and click **Show Options**.
3. Enter the username. The default username is `administrator`.
4. (Optional)If you do not want to enter the password upon subsequent logons, select **Allow me to save credentials**.
5. Click **Connect**.
6. In the **Windows Security** dialog box that appears, enter the password corresponding to the username you entered and click **OK**.

Result

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

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

1. [Connect to an ECS instance by using the VNC](#).
2. Choose **Start > Control Panel**.
3. Click **System and Security**.
4. Click **Check for updates** in the **Windows Updates** section.

5. If updates are available, click **Install updates**.
6. Restart the instance.

2.4.2.5. Install the certificate for VNC in Windows

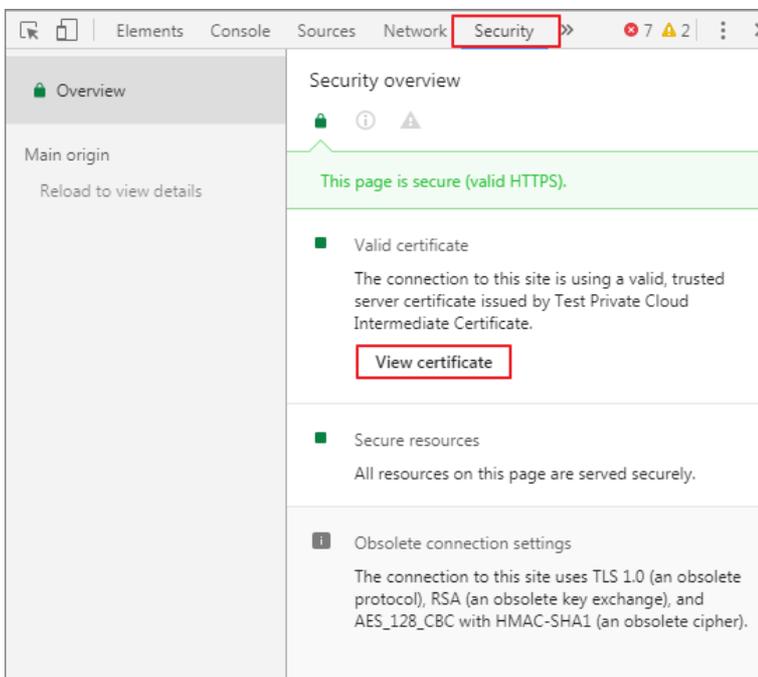
Before you log on to VNC, you must export the relative certificate from the site such as the Apsara Stack Cloud Management (ASCM) console and install the certificate in your local browser.

Context

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

Procedure

1. Export the certificate from the ASCM console.
 - i. Log on to the ASCM console. Press the **F12** key or **Fn+F12** to view and select the certificate. For example, in the Chrome browser, press the **F12** key to open Chrome DevTools.



- ii. In the **Certificate** dialog box, click the **Certificate Path** tab, select the root certificate, and then click **View Certificate**.
 - iii. In the **Certificate** dialog box, click the **Details** tab, and click **Copy to File**.
 - iv. In the **Certificate Export Wizard** dialog box, click **Next**.
 - v. Select **DER encoded binary X.509 (.CER)** as the format and then click **Next**.
 - vi. Click **Browse**, select where to store the certificate, enter a file name, and then click **Save**.
 - vii. Click **Next**.
 - viii. Click **Finish**.
 - ix. Click **OK**.
2. Install the certificate in your local browser.
 - i. Double-click the certificate.
 - ii. In the **Certificate** dialog box, click **Install Certificate**.
 - iii. In the **Certificate Import Wizard** dialog box, click **Next**.

- iv. Select **Place all certificates in the following store** and click **Browse**.
 - v. In the **Select Certificate Store** dialog box, select **Trusted Root Certificate Authority** and then click **OK**.
 - vi. In the **Certificate Import Wizard** dialog box, click **Next**.
 - vii. Click **Finish**.
 - viii. If a security warning message is displayed, click **Yes**.
3. Restart your browser and log on to the ASCM console. If no security warning message is displayed in the left part of the address bar, the certificate is installed.



2.4.2.6. Connect to an ECS instance by using the VNC

You can access your instance by using the VNC in the ECS console when other SSH clients such as PuTTY, Xshell, and SecureCRT do not work properly.

Prerequisites

- The instance is in the **Running** state.
- The root certificate is imported to your web browser. For more information, see [Install a certificate](#).
- If you log on to an instance for the first time after the instance is created, make sure that you set a new VNC password. For more information, see [Set a VNC password](#).

Context

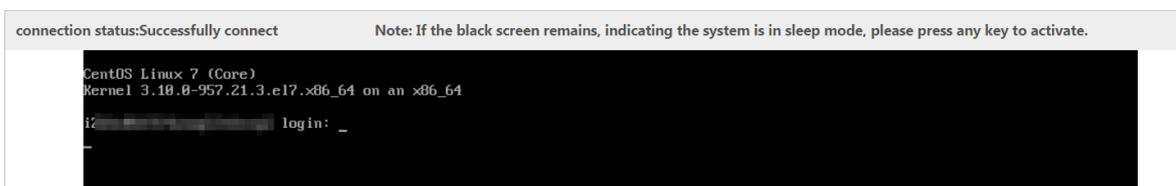
The VNC password is used to log on to the VNC of the ECS console, and the instance password is used to log on to an instance.

You can use the VNC to connect to an instance to solve issues shown in the following table.

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the ECS instance, and click **Remote Connection** in the **Actions** column.
5. Enter the VNC password, and then click **OK**. After the connection is successful, the logon page is displayed, as shown in the following figure.

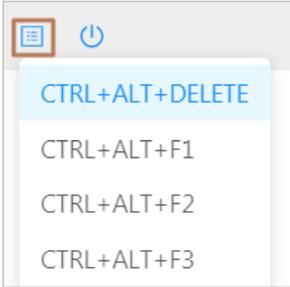


6. Enter the username and password.

- For Linux instances: Enter the username *root* and the logon password.

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

- For Windows instances: To use key combinations such as Ctrl + Alt + Delete, click the corresponding key combination in the upper-right corner of the VNC page.



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

2.4.3. View instances

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region. The created instances that match the specified criteria are displayed.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

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

Filter option	Description
Instance Name	Enter an instance name to search for the instance.
Instance ID	Enter an instance ID to search for the instance.
IP Address	Enter the IP address of an instance to search for the instance.
VPC ID	Enter a VPC ID to search for the instances that belong to the VPC.
Image ID	Enter an image ID to search for the instances that use the image.

Filter option	Description
Status	Select an instance status to search for the instances in that status. Valid values: <ul style="list-style-type: none"> Running Stopped Starting Stopping
Security Group ID	Enter a security group ID to search for the instances that belong to the security group.
Operating System	Enter the name of operating system to search for the instances that use the operating system.

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

2.4.4. Modify an instance

You can modify the name, description, and custom data of an existing instance.

Procedure

- Log on to the ECS console.
- In the left-side navigation pane, click **Instances**.
- In the top navigation bar, select an organization, a resource set, and a region.
- Find the instance and choose **More > Modify** from the Actions column.
- Modify the name, description, and custom data for the instance. The name must be 2 to 128 characters in length. The description must be 2 to 256 characters in length. The custom data must be 2 to 999 characters in length.
- Click **OK**.

2.4.5. Stop an instance

You can stop an instance that is not in use. Stopping an instance will interrupt the services that are running on it. Exercise caution when you stop an instance.

Prerequisites

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

Procedure

- Log on to the ECS console.
- In the left-side navigation pane, click **Instances**.
- In the top navigation bar, select an organization, a resource set, and a region.
- Use one of the following methods to stop the instance:
 - To stop a single instance, find the instance and choose **More > Instance Status > Stop** in the Actions column.
 - To stop one or more instances at a time, select the instances and click **Stop** in the lower-left corner of the Instances page.

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

Result

In the **Status** column, the instance state changes from **Running** to **Stopping**. After the instance is stopped, its state changes to **Stopped**.

2.4.6. Start an instance

You can start a stopped instance.

Prerequisites

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

Procedure

1. **Log on to the ECS console.**
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Use one of the following methods to start the instance:
 - To start a single instance, find the instance and choose **More > Instance Status > Start** in the **Actions** column.
 - To start one or more instances at a time, select the instances and click **Start** in the lower-left corner of the **Instances** page.
5. In the message that appears, click **OK**.

Result

In the **Status** column, the instance state changes from **Stopped** to **Starting**. After the instance is started, its state changes to **Running**.

2.4.7. Restart an instance

You must restart an instance after you change its logon password or install system updates. Restarting an instance will stop the instance and interrupt the services that are running on it for a period of time. Exercise caution when you restart an instance.

Prerequisites

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

Procedure

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

2.4.8. Delete an instance

You can delete an instance that is no longer needed to release its resources. Deleted instances cannot be recovered. We recommend that you back up data before you delete an instance. If a data disk is released along with the instance, the data on the disk cannot be recovered.

Prerequisites

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Select the instance and click **Delete** in the lower-left corner of the Instances page.
5. In the message that appears, click **OK**.

2.4.9. Change the instance type of an instance

You can change the instance type of an instance to suit your business needs. This eliminates the need to create a new instance.

Prerequisites

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the instance and click **Upgrade/Downgrade** in the **Actions** column.
5. On the **Change Specifications** page that appears, select a new instance type and click **Submit**. The instance types available for selection are displayed on the **Change Specifications** page.
6. Restart the instance by using the console or calling an API operation for the new instance type to take effect. For more information, see [Start an instance](#) or `StartInstance` in *ECS Developer Guide*.

2.4.10. Change an instance logon password

If you did not set a logon password when creating an instance or forgot the password, you can reset the password.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the instance and select one of the following methods to access the instance details page.
 - In the **Instance ID/Name** column, click the instance ID.
 - Click **Manage** in the **Actions** column.
 - In the **Actions** column, choose **More > Show Details**.
5. Click **Change Password**.
6. Enter and confirm the new password, and then click **OK**. The password must be 8 to 30 characters in length

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

7. Restart the instance in the console or by calling an API operation to make the new password take effect. For more information, see [Restart an instance](#) or *the RebootInstance section* in ECS Developer Guide.

2.4.11. Change the VNC password

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the instance and select one of the following methods to access the instance details page.
 - In the **Instance ID/Name** column, click the instance ID.
 - Click **Manage** in the **Actions** column.
 - In the **Actions** column, choose **More > Show Details**.
5. Click **Change VNC Password**.
6. Enter and confirm the new password, and click **OK**. The password must be 6 characters in length and can contain digits and uppercase and lowercase letters. It does not support special characters.
7. Restart the instance in the console or by calling an API operation to make the new password take effect. For more information, see [Restart an instance](#) or *the RebootInstance section* in ECS Developer Guide.

2.4.12. Add an ECS instance to a security group

You can add a created instance to a security group and use security group rules to control network access to the instance.

Context

A security group acts as a virtual firewall and is used to provide security isolation. A security group controls access to ECS instances.

Instances that belong to the same account and are in the same region and in the same security group can communicate with each other over the internal network. If instances that belong to the same account in the same region are in different security groups, you can implement internal network communication by authorizing mutual access between two security groups.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the security group and click **Manage Instances** in the **Actions** column.
5. Click **Add Instance**.
6. Select the instance and click **OK**. An instance can be added to five security groups. After an instance is added, the security group rules automatically apply to the instance.

2.4.13. Customize instance data

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

Context

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

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

Usage instructions

• Limits

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

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

Supported operating systems

Windows	Linux
<ul style="list-style-type: none"> ▪ Windows Server 2016 64-bit ▪ Windows Server 2012 64-bit ▪ Windows Server 2008 64-bit 	<ul style="list-style-type: none"> ▪ CentOS ▪ Ubuntu ▪ SUSE Linux Enterprise ▪ OpenSUSE ▪ Debian ▪ Aliyun Linux

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

 **Note** Only English characters are allowed.

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

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

- For Linux instances, the script format must meet the requirements described in [Types of Linux instance customization scripts](#).
- For Windows instances, the script can only start with `[bat]` or `[powershell]`.
- After starting an instance, run a command to view the following information:
 - Execution result of the instance customization script
 - Data imported to instances
- **Console:** You can modify the custom instance data in the console. Whether the modified instance customization script needs to be re-executed depends on the script type. For example, if the `bootcmd` script in Cloud Config is modified for Linux instances, the script is automatically executed each time instances are restarted.
- **OpenAPI:** You can also use OpenAPI to customize instance data. For more information, see [CreateInstance](#) and [ModifyInstanceAttribute](#) in *ECS Developer Guide*.

Linux instance data customization scripts

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

Description of Linux instance data customization scripts

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

Types of Linux instance customization scripts

• User-Data Script

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

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

• Cloud Config Data

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

```
#cloud-config
apt:
primary:
- arches: [default]
uri: http://us.archive.ubuntu.com/ubuntu/
```

• Include

- Description: The configuration content can be saved in a text file and imported into cloud-init as a URL.
- Format: The first line must be `#include`.
- Limit: The script size (including the first line) cannot exceed 16 KB before the data is Base64 encoded.
- Frequency: The script execution frequency depends on the script type in the text file.
- Example:

```
#include
http://ecs-image-test.oss-cn-hangzhou.aliyuncs.com/userdata/cloudconfig
```

• GZIP format

- Description: Cloud-init limits the size of customization scripts to 16 KB. You can compress and import the script file into the customization script if the file size exceeds 16 KB.
- Format: The `.gz` file is imported into the customization script as a URL in `#include`.
- Frequency: The script execution frequency depends on the script content contained in the GZIP file.

- Example:

```
#include
http://ecs-image-test.oss-cn-hangzhou.aliyuncs.com/userdata/config.gz
```

View the custom data of a Linux instance

Run the following command in the instance:

```
curl http://100.100.100.200/latest/user-data
```

Windows instance customization scripts

Windows instance customization scripts independently developed by Alibaba Cloud can be used to initialize Windows instances.

There are two types of Windows instance customization scripts:

- Batch processing program: starts with `[bat]` and serves as the first line. The script size must be smaller than 16 KB before the data is Base64 encoded.
- PowerShell script: starts with `[powershell]` and serves as the first line. The script size must be smaller than 16 KB before the data is Base64 encoded.

View the custom data of a Windows instance

Run the following PowerShell command in the instance:

```
Invoke-RestMethod http://100.100.100.200/latest/user-data/
```

2.4.14. Modify a private IP address

Each instance is assigned a private NIC and bound with a private IP address. You can modify the private IP address of the instance. The private IP address you use must be within the CIDR block of the VSwitch to which the instance belongs and cannot be used by another instance.

Prerequisites

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the instance and choose **More > Change Private IP Address** from the **Actions** column.
5. Enter a new private IP address and click **OK**. The private IP address you can use must be within the CIDR block of the VSwitch to which the instance belongs and cannot be used by another instance or for a specific purpose.

For example, if the CIDR block of the VSwitch is 192.168.1.0/24, you can use an IP address in the range of 192.168.1.1 to 192.168.1.254. The first address 192.168.1.0 identifies the subnet itself, and the last address 192.168.1.255 identifies the broadcast address. Both addresses are reserved and cannot be used.

2.4.15. Install the CUDA and GPU drivers for a Linux instance

You must install a GPU driver on GPU instances to use the GPU. If the image you use does not contain a pre-installed GPU driver, you must manually install the CUDA and GPU drivers for the instance.

Prerequisites

If your instance cannot connect to the Internet, the installation file cannot be downloaded. You can install an FTP client on the instance to transfer the installation file to the instance.

Context

When installing NVIDIA drivers, you must install the kernel package that contains the kernel header file before you install the CUDA and GPU drivers on the instance.

Procedure

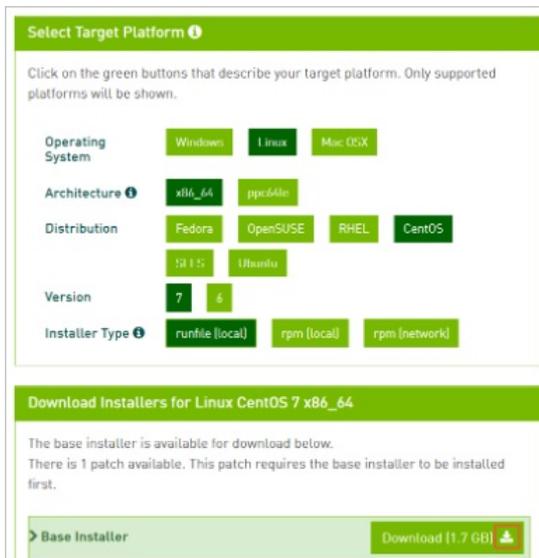
1. Install the kernel package.
 - i. Run the `uname -r` command to view the current kernel version. A similar output is displayed:
 - CentOS: `3.10.0-862.14.4.el7.x86_64`
 - Ubuntu: `4.4.0-117-generic`
 - ii. Copy the kernel package of the corresponding version to the instance and install the package.
 - CentOS: Copy the RPM package of the `kernel-devel` component and run the `rpm -ivh 3.10.0-862.14.4.el7.x86_64.rpm` command to install the package. `3.10.0-862.14.4.el7.x86_64.rpm` is used as an example. Replace it with the actual package name.
 - Ubuntu: Copy the DEB package of the `linux-headers` component and run the `dpkg -i 4.4.0-117-generic.deb` command to install the package. `4.4.0-117-generic.deb` is used as an example. Replace it with the actual package name.
2. Download the CUDA Toolkit.
 - i. Access the [official CUDA download page](#). Choose the version based on the GPU application requirements for CUDA. This example uses [CUDA Toolkit 9.2](#).

Download the CUDA Toolkit

The screenshot shows the 'Latest Release' section with 'CUDA Toolkit 10.0 (Sept 2018)'. Below it is the 'Archived Releases' section with a list of versions: 'CUDA Toolkit 9.2 (May 2018), Online Documentation', 'CUDA Toolkit 9.1 (Dec 2017), Online Documentation', 'CUDA Toolkit 9.0 (Sept 2017), Online Documentation', 'CUDA Toolkit 8.0 GA2 (Feb 2017), Online Documentation', 'CUDA Toolkit 8.0 GA1 (Sept 2016), Online Documentation', 'CUDA Toolkit 7.5 (Sept 2015)', 'CUDA Toolkit 7.0 (March 2015)', 'CUDA Toolkit 6.5 (August 2014)', and 'CUDA Toolkit 6.0 (April 2014)'. The 'CUDA Toolkit 9.2' link is highlighted with a red rectangular box.

- ii. Choose a platform based on your operating system. Select **Installer Type** to **runfile (local)** and click **Download**. NVIDIA drivers are already included in the CUDA Toolkit.

Download the drivers



3. Copy the downloaded `cuda_9.2.148_396.37_linux.run` file to the instance. `cuda_9.2.148_396.37_linux.run` is used as an example. Replace it with the actual file name.
4. Run the `sudo sh ./cuda_9.2.148_396.37_linux.run --silent --verbose --driver --toolkit --samples` command to install the CUDA driver. `cuda_9.2.148_396.37_linux.run` is used as an example. Replace it with the actual file name.

The installation takes about 10 to 20 minutes. When `Driver: Installed` is displayed, the installation is successful.

View the CUDA installation result

```

= Summary =
-----
Driver: Installed
Toolkit: Installed in /usr/local/cuda-9.2
Samples: Installed in /home/lb164654, but missing recommended libraries

Please make sure that
- PATH includes /usr/local/cuda-9.2/bin
- LD_LIBRARY_PATH includes /usr/local/cuda-9.2/lib64, or, add /usr/local/cuda-9.2/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-9.2/bin
To uninstall the NVIDIA Driver, run nvidia-uninstall

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-9.2/doc/pdf for detailed information on setting up CUDA.

Logfile is /tmp/cuda_install_19765.log

```

5. Run the `nvidia-smi` command to view the GPU driver status. If the output displays the details of the GPU driver, the driver is running properly.

View the GPU driver status

```

$ nvidia-smi
Mon Oct 15 19:05:00 2018

+-----+
| NVIDIA-SMI 396.37                Driver Version: 396.37          |
+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|   0   Tesla P4             Off   | 00000000:00:08.0 Off  |   0         0         |
| N/A   28C    P0             23W / 75W |  0MiB / 7611MiB |   0%      Default   |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                         GPU Memory          |
| GPU       PID    Type    Process name          Usage          |
+-----+-----+-----+-----+-----+
| No running processes found         |
+-----+

```

What's next

If you want to run the OpenGL program, you must first purchase the licenses and install the GRID drivers. For information about the installation procedure, see the official NVIDIA documentation.

2.4.16. Install the CUDA and GPU drivers for a Windows instance

You must install a GPU driver on GPU instances to use the GPU. If the image you use does not contain a pre-installed GPU driver, you must manually install the CUDA and GPU drivers for the instance.

Prerequisites

- If your instance cannot connect to the Internet, the installation file cannot be downloaded. You can install an FTP client on the instance to transfer the installation file to the instance.
- To compile CUDA programs, first install a Windows compiling environment, such as Visual Studio 2015. If you do not need to compile CUDA programs, ignore it.

Procedure

1. Download the CUDA Toolkit.
 - i. Access the [official CUDA download page](#). Choose the version based on the GPU application requirements for CUDA. This example uses [CUDA Toolkit 9.2](#).
 - ii. Choose a platform based on your operating system. Set **Installer Type** to **exe (local)** and click **Download**. NVIDIA drivers are already included in the CUDA Toolkit.
2. Copy the downloaded `cuda_9.2.148_windows.exe` file to the instance. `cuda_9.2.148_windows.exe` is used as an example. Replace it with the actual file name.
3. Double-click `cuda_9.2.148_windows.exe` and follow the installation wizard to install the CUDA driver. `cuda_9.2.148_windows.exe` is used as an example. Replace it with the actual file name. The installation takes about 10 to 20 minutes. When `Installed: - Nsight Monitor and HUD Launcher` is displayed, the driver is installed.
4. Press **Win + R** and enter `devmgmt.msc`. The NVIDIA device is displayed in **Display Adapter**.
5. Press **Win + R**, enter `cmd`, and run the `"C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi"` command. If the output displays the details of the GPU driver, the driver is running properly.

What's next

If you want to run the OpenGL and DirectX programs, you must first purchase the licenses and install the GRID drivers. For information about the installation procedure, see the official NVIDIA documentation.

2.5. Disks

2.5.1. Create a disk

You can create an independent data disk and attach it to an ECS instance to increase the storage space of the instance. This topic describes how to create an empty data disk. You cannot create independent system disks.

Context

We recommend that you plan the number and size of data disks before you create them. The following limits apply to data disks:

- A maximum of 16 data disks can be attached to an instance. Disks and Shared Block Storage devices share this quota.
- Each Shared Block Storage device can be attached to up to four ECS instances at the same time.
- Each ultra disk, shared ultra disk, standard SSD, or shared SSD can have a maximum capacity of 32 TiB.
- Disks cannot be combined in ECS. They are independent of each other. You cannot combine multiple disks into one by formatting them.

We recommend that you do not use Logical Volume Manager (LVM) to create logical volumes across multiple disks, because a snapshot can only back up data of a single disk. If you create a logical volume across several disks, data discrepancies will occur when you restore these disks.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. Click **Create Disk**.
4. Configure parameters listed in the following table to create a disk.

Section	Parameter	Required	Description
Region	Organization	Yes	Select an organization in which to create the disk.
	Resource Set	Yes	Select a resource set in which to create the disk.
	Region	Yes	Select a region in which to create the disk.
	Zone	Yes	Select a zone in which to create the disk.
	Name	Yes	Enter a name for the disk. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).

Section	Parameter	Required	Description
Basic Settings	Specifications	Yes	<p>Select a disk category and specify the disk size. Valid disk categories:</p> <ul style="list-style-type: none"> ◦ SSD Disk ◦ Ultra Disk ◦ Shared SSD: shared SSD ◦ Shared Ultra Disk: shared ultra disk <p>The disk size must range from 20 GiB to 32768 GiB.</p>
	Encrypt	No	Specify whether to encrypt the disk.
	Encryption Algorithm	No	<p>Select an encryption algorithm. This parameter is required when you set Encrypt to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◦ AES256: AES256 encryption algorithm. ◦ SM4: Chinese encryption algorithm SM4.
	Encryption Key	No	<p>Select an encryption key. This parameter is required when you set Encrypt to Yes.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note If no key is available, create a key in KMS.</p> </div>

Section	Parameter	Required	Description
	Use Snapshot	No	<p>Specify whether to create the disk from a snapshot. If you select Yes, you must specify a snapshot. The size of the created disk depends on the size of the specified snapshot.</p> <ul style="list-style-type: none"> ◦ If the disk size that you specify is greater than the snapshot size, the disk will be created with the size you specified. ◦ If the disk size that you specify is smaller than the snapshot size, the disk will be created with the snapshot size.

5. Click **Submit**.

Result

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

What's next

After the disk is created, you must attach the disk to an instance and partition and format the disk. For more information, see the following topics:

-
- [Format a data disk for a Linux instance](#)
- [Format a data disk of a Windows instance](#)

2.5.2. View disks

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region. The created disks that match the specified criteria are displayed.
4. Select a filter option from the drop-down list, enter the relevant information in the search box, and click **search**.

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

Filter option	Description
Disk Name	Enter a disk name to search for the disk.
Disk ID	Enter a disk ID to search for the disk.

Filter option	Description
Instance ID	Enter an instance ID to search for the disks that are attached to the instance.
Disk Status	<p>Select a disk status to search for disks in that status. Valid values:</p> <ul style="list-style-type: none"> ◦ Running ◦ Pending ◦ Attaching ◦ Detaching ◦ Creating ◦ Deleting ◦ Deleted <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin: 5px 0;"> <p>? Note Deleted disks are no longer displayed in the disk list.</p> </div> <ul style="list-style-type: none"> ◦ Initializing ◦ All Statuses
Disk Properties	<p>Select a disk type to search for disks of that type. Valid values:</p> <ul style="list-style-type: none"> ◦ All ◦ System Disk ◦ Data Disk
Policy ID	Enter the ID of an automatic snapshot policy to search for the disks that use the policy.
Encryption Key ID	Enter the ID of an encryption key to search for the disks that are encrypted with the key.

5. In the **Disk ID/Name** column, click a disk ID to go to the Disk Details page of the disk. The properties and mount information of the disk are displayed on the Disk Details page.

2.5.3. Roll back a disk

If you have created snapshots for a disk, you can use a snapshot to roll back the disk to the state it was when the snapshot was taken. Rolling back a disk is irreversible. Once the disk is rolled back, the disk data before the rollback time cannot be restored. Exercise caution when you perform this operation.

Prerequisites

- Snapshots have been created for the disk.
- The disk is not released.
- The instance where the target disk resides must be in the **Stopped** state.

Procedure

1. **Log on to the ECS console.**
2. In the left-side navigation pane, choose **Snapshots and Images > Snapshots**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Select a filtering option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

You can select multiple filtering options to narrow down the search results.

Filtering option	Description
Snapshot Name	Enter a snapshot name to search for the snapshot.
Snapshot ID	Enter a snapshot ID to search for the snapshot.
Instance ID	Enter an instance ID to search for the snapshots related to the instance.
Disk ID	Enter a disk ID to search for the snapshots related to the disk.
Snapshot Type	Select a snapshot type to search for the snapshots of that type. Options include: <ul style="list-style-type: none"> ◦ All ◦ User Snapshots: manual snapshots. ◦ Automatic snapshots: automatic snapshots.
Creation Time	Enter a creation time to search for the snapshots that were created at that time.

5. Find the snapshot and click **Restore** in the **Actions** column.
6. Click **OK**.

2.5.4. Modify the disk properties

You can modify the properties of a created disk, such as changing the settings of the Release Disk with Instance and Release Automatic Snapshots with Disk options.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the disk and choose **More > Modify Disk Properties** from the **Actions** column.
5. Modify the release mode.
 - **Release Disk with Instance:** When this option is selected, the disk is released together when the instance it is attached to is deleted. When this option is not selected, the disk changes to the **Pending** state when the instance it is attached to is deleted.
 - **Release Automatic Snapshots with Disk:** When this option is selected, the automatic snapshots created for the disk is released together when the disk is deleted. When this option is not selected, the automatic snapshots are retained.
6. Click **OK**.

2.5.5. Modify the disk description

You can modify the name and description of a created disk.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the disk and choose **More > Modify Disk Description** from the **Actions** column.

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

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

6. Click OK.

2.5.6. Attach a disk

You can attach a disk that is created separately to an ECS instance as a data disk. The disk and the instance must be in the same region and the same zone.

Prerequisites

The disk is in the **Pending** state.

Context

- You do not need to attach data disks that are created at the same time as an instance.
- A disk can only be attached to an instance that is in the same zone and region as the disk.
- You can attach a disk to a single ECS instance at a time.
- A Shared Block Storage device can be attached to a maximum of four ECS instances at the same time.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the disk and choose **More > Attach** from the **Actions** column.
5. Specify the destination instance and configure the release mode as needed.
 - If you select **Release Disk with Instance**, the disk is released together when the instance it is attached to is deleted.
 - If you do not select **Release Disk with Instance**, the disk changes to the **Pending** state when the instance it is attached to is deleted.
6. Click OK.

2.5.7. Partition and format disks

2.5.7.1. Format a data disk for a Linux instance

Data disks created separately are not partitioned or formatted. This topic describes how to partition and format a data disk of a Linux instance.

Prerequisites

The disk has been attached to the instance.

Procedure

1. [Connect to the instance](#).
2. Run the `fdisk -l` command to view all data disks attached to the ECS instance. If `/dev/vdb` is not displayed in the command output, the ECS instance does not have a data disk. Check whether the data disk is attached to the instance.

```
[root@iZ*****eZ ~]# fdisk -l
Disk /dev/vda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00078f9c

Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          1         5222   41940992   83  Linux
Disk /dev/vdb: 21.5 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

3. Create partitions for the data disk.
 - i. Run the `fdisk /dev/sdb` command.
 - ii. Enter `n` to create a new partition.
 - iii. Enter `p` to set the partition as the primary partition.
 - iv. Enter a partition number and press the Enter key. In this example, `7` is entered to create Partition 1.
 - v. Enter the number of the first available sector. This example uses the default value. This is done by pressing the Enter key. You can also enter a value from 1 to 41610 and press the Enter key.
 - vi. Enter the number of the last sector. This example uses the default value. This is done by pressing the Enter key. You can also enter a value from 1 to 11748 and press the Enter key.
 - vii. (Optional)Optional. To create multiple partitions, repeat steps b through f until all four primary partitions are created.

viii. Run the `wq` command to start partitioning.

```
[root@iZ*****eZ ~]# fdisk /dev/vdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x01ac58fe.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-41610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-41610, default 41610):
Using default value 41610

Command (m for help): wq
The partition table has been altered!
```

4. Run the `fdisk -l` command to view the partitions. If `/dev/vdb1` is displayed in the command output, new partition `vdb1` is created.

```
[root@iZ*****eZ ~]# fdisk -l
Disk /dev/vda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00078f9c

Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          1         5222    41940992   83  Linux
Disk /dev/vdb: 21.5 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x01ac58fe

Device Boot      Start         End      Blocks   Id  System
/dev/vdb1          1         41610    20971408+   83  Linux
```

5. Format the new partition. In this example, the new partition is formatted as ext3 after you run the `mkfs.ext3 /dev/vdb1` command. The time required for formatting depends on the disk size. You can also format the new partition to another file system. For example, you can run the `mkfs.ext4 /dev/vdb1` command to format the partition as ext4.

Compared with ext2, ext3 only adds the log function. Compared with ext3, ext4 improves on some important data structures. ext4 provides better performance and reliability, and more functions.

```
[root@iZ*****leZ ~]# mkfs.ext3 /dev/vdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5242852 blocks
262142 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

6. Run the `echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab` command to write the information of the new partition to the `/etc/fstab` file. You can run the `cat /etc/fstab` command to view the new partition information. Ubuntu 12.04 does not support barriers. To write the information of the new partition into the `/etc/fstab` file, you must run the `echo '/dev/vdb1 /mnt ext3 barrier=0 0 0' >> /etc/fstab` command.

In this example, the partition information is added to the ext3 file system. You can also modify the ext3 parameter to add the partition information to another type of file system.

To attach the data disk to a specific folder, for example, to store web pages, modify the `/mnt` part of the preceding command.

```
[root@iZ*****eZ ~]# echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab
[root@iZbp19cdhgdj0aw5r2izleZ ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Aug 14 21:16:42 2014
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=94e4e384-0ace-437f-bc96-057dd64f**** / ext4 defaults,barrier=0 1 1
tmpfs      /dev/shm      tmpfs defaults    0 0
devpts     /dev/pts      devpts gid=5,mode=620 0 0
sysfs      /sys          sysfs defaults    0 0
proc       /proc         proc  defaults    0 0
/dev/vdb1 /mnt ext3 defaults 0 0
```

7. Mount the new partitions. Run the `mount -a` command to mount all the partitions listed in `/etc/fstab` and run the `df -h` command to view the result. If the following information is displayed, the new partitions are mounted and available for use.

```
[root@iZ*****eZ ~]# mount -a
[root@iZ*****eZ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       40G  5.6G  32G  15% /
tmpfs           499M   0 499M   0% /dev/shm
/dev/vdb1       20G  173M  19G   1% /mnt
```

2.5.7.2. Format a data disk of a Windows instance

Data disks created separately are not partitioned or formatted. This topic describes how to partition and format a data disk of a Windows instance. This example uses Windows Server 2008.

Prerequisites

The disk has been attached to an instance.

Procedure

1. In the lower-left corner of the screen, click the **Server Manager** icon.
2. In the left-side navigation pane of the Server Manager window, choose **Storage > Disk Management**.
3. Right-click an empty partition and select **New Simple Volume** from the shortcut menu. If the disk status is **Offline**, change it to **Online**.
4. Click **Next**.
5. Set the size of the simple volume, which is the partition size. Then click **Next**. The default value is the maximum value of the disk space. You can specify the partition size as needed.
6. Specify the drive letter and then click **Next**.
7. Specify the formatting options and then click **Next**. We recommend that you format the partition with the default settings provided by the wizard.
8. When the wizard prompts that the partition has been completed, click **Finish** to close the wizard.

2.5.8. Expand a disk

You can expand system or data disks online. After a disk is expanded, you do not need to restart the instance to which the disk is attached for the new disk capacity to take effect.

Prerequisites

- To avoid data loss, we recommend that you create a snapshot to back up disk data before you expand a disk. For more information, see [Create a snapshot](#).
- No snapshot is being created for the disk to be expanded.
- The disk or the instance to which the disk is attached meets the following requirements:
 - If the disk is a system disk, the instance is in the **Running** state.
 - If the disk is a data disk, one of the following requirements is met:
 - The disk is in the **Pending** state.
 - If the disk is attached to an instance, the instance is in the **Running** state.
 - If the disk is a Shared Block Storage device, it is in the **Pending** state.

Context

The following limits apply when you expand a disk.

Limit	Description
Disk category	<ul style="list-style-type: none"> • Ultra disks and standard SSDs can be expanded. • Shared SSDs and shared ultra disks can be expanded.
Operating system	The system disks of Windows Server 2003 instances cannot be expanded.
Partitioning mode	You cannot expand a data disk that uses the MBR partitioning scheme to more than 2 TiB. To expand this kind of disk to more than 2 TiB, we recommend that you create and attach a new data disk with the desired size. Use the GPT partitioning scheme to partition the new data disk and then copy data from the original data disk to the new data disk.
File system	For Windows instances, you can expand only disks that use NTFS file systems.
Maximum capacity	<ul style="list-style-type: none"> • Ultra disk and standard SSD: 32,768 GiB • Shared SSD and shared ultra disk: 32,768 GiB
Operations	<ul style="list-style-type: none"> • When you expand disks, only the capacity of the disks is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate storage space on a disk after the disk is expanded. • You cannot shrink an expanded disk by any means, such as by rolling it back.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. Find the target disk and choose **More > Expand Disk** in the **Actions** column.
4. In the Expand Disk dialog box that appears, specify a new capacity for the disk. The new capacity must be greater than the current capacity.
5. Click **OK**.

Result

When you expand disks, only the capacity of the disks is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate storage space on a disk after the disk is expanded.

2.5.9. Reinitialize a disk

You can reinitialize a disk to restore it to its initial state.

Prerequisites

- The disk is in the **Running** state.
- The instance is in the **Stopped** state.
- After a disk is reinitialized, its data is lost and cannot be recovered. Exercise caution when you perform this operation. We recommend that you back up the data of the disk or create snapshots before you reinitialize the disk. For more information, see [Create a snapshot](#).

Context

The result of disk reinitialization depends on the disk type and how the disk is created.

- **System disk:**
 - The disk is restored to the initial state of the image used by the disk.
 - If the original image is deleted, the disk cannot be reinitialized.
- **Data disk:**
 - If the disk is empty when created, the disk is restored to an empty disk.
 - If the disk is created from a snapshot, the disk is restored to a disk with only the data of the source snapshot.
 - If the disk is created from a snapshot and the snapshot is deleted, the disk cannot be reinitialized.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the data disk and click **Reinitialize** in the **Actions** column.
5. Perform the following operations based on the disk type.
 - For a system disk, enter and confirm a new logon password, select the **Start Instance After Reinitializing Disk** option as needed, and then click **OK**.
 The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include `()'~!@#$%^&*-_+=|{}[]:;<>,./`
 - For a data disk, click **OK**.

Result

When the disk is being reinitialized, the disk enters the **Initializing** state. After the reinitialization, it changes to the **Running** state.

2.5.10. Detach a data disk

You can detach a data disk, not a system disk.

Prerequisites

- For a Windows instance, you must bring the data disk offline in Disk Management.

 **Note** To guarantee data integrity, we recommend that you stop read/write operations on the data disk when you detach the disk. Otherwise, data may be lost.

- For a Linux instance, you must connect to the instance and unmount the partitions on the disk.

 **Note** If you have configured the `/etc/fstab` file to automatically mount the disk partitions upon instance startup, you must delete the mounting information from the `/etc/fstab` file before you detach the disk. Otherwise, you cannot connect to the instance after the instance is restarted.

- The data disk to be detached is in the **Running** state.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the data disk and choose **More > Detach** from the **Actions** column.
5. Click **OK**.

2.5.11. Release a data disk

You can release a data disk that is no longer needed. The released data disk cannot be recovered. Exercise caution when you release a data disk.

Prerequisites

The data disk is in the **Pending** state. If the data disk is attached to an instance, detach the disk from the instance first.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the data disk and choose **More > Release** from the **Actions** column.
5. Click **OK**.

2.6. Images

2.6.1. Create a custom image

You can create a custom image and use it to create identical instances or replace the system disks of existing instances. This allows you to have multiple instances with the same operating system and data environment.

Create a custom image from a snapshot

You can create a custom image from a system disk snapshot to fully load the operating system and data environment of the snapshot to the image. Before you perform this operation, make sure that a snapshot of system disks is used. You cannot create a custom image from snapshots of data disks.

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Snapshots**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the snapshot and click **Create Custom Image** in the **Actions** column.
5. Enter the name and description of the image, and then click **OK**. The name must be 2 to 128 characters in

length and can contain periods (.), underscores (_), hyphens (-), and colons (:). It cannot start with a special character or digit.

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

Create a custom image from an instance

You can create a custom image from an instance to completely replicate the data of all disks of the instance, including the system disk and data disks.

 **Note** To avoid data security risks, delete sensitive data before you create a custom image.

When you create a custom image from an instance, a snapshot is generated for each disk in the instance, and all the snapshots constitute a complete custom image.

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Instances**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the instance and choose **More > Create Custom Image** from the **Actions** column.
5. Enter the name and description of the custom image, and then click **OK**. The name must be 2 to 128 characters in length and can contain periods (.), underscores (_), hyphens (-), and colons (:). It cannot start with a special character or digit.

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

2.6.2. View images

You can view the list of created images.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The created images that match the specified criteria are displayed.
4. Select the tab based on the type of images you want to view. You can select the **Custom Images** or **Public Images** tab.
5. Select a filtering option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

You can select multiple filtering options to narrow down the search results.

Filtering option	Description
Image Name	Enter an image name to search for the image.
Image ID	Enter an image ID to search for the image.
Snapshot ID	Enter a snapshot ID to search for the images associated with the snapshot. This option is not available for public images.

2.6.3. View instances related to an image

You can view the instances that use the specified image.

Procedure

1. [Log on to the ECS console](#).

2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Select a tab based on the type of the image. You can select the **Custom Images** or **Public Images** tab.
5. Find the target image and click **Related Instances** in the **Actions** column.

Result

The **Instances** page appears, showing the instances that use the image. You can perform operations on these instances, such as updating the image.

2.6.4. Modify the description of a custom image

You can modify the description of a created custom image.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target custom image and click **Modify Description** in the **Actions** column.
5. In the dialog box that appears, modify the image description in the **Basic Settings** field. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.
6. Click **OK**.

2.6.5. Share custom images

You can share a custom image that you create to organizations that you manage to create multiple identical ECS instances in a short time.

Context

Only custom images can be shared. Shared images are not counted towards the image quota assigned to the organization.

The organization can use the shared image to create instances or replace system disks of existing instances.

You can delete shared images. After a shared image is deleted, the image is no longer visible to the organization to which the image was shared. The system disk of instances created from the shared image can no longer be reinitialized.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the image and click **Share Image** in the **Actions** column.
5. Select the organization to which you want to share the image and click **OK**. An image can be shared only to the organizations that the image owner manages.

2.6.6. Import custom images

2.6.6.1. Limits on importing custom images

This topic describes the limits on importing images. You must understand the limits to ensure image availability and improve import efficiency.

The following limits apply when you import custom images:

- [Limits on importing custom images in Linux](#)

- [Limits on importing custom images in Windows](#)

Limits on importing custom images in Linux

When you import custom images in Linux, note the following limits:

- Multiple network interfaces are not supported.
- IPv6 addresses are not supported.
- The password must be 8 to 30 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- The firewall is disabled, and port 22 is opened.
- The Linux system disk size ranges from 40 GiB to 500 GiB.
- DHCP must be enabled in the image.
- SELinux is disabled.
- The Kernel-based Virtual Machine (KVM) driver must be installed.
- We recommend that you install cloud-init to configure the hostname and NTP and yum sources.

Limits

Item	Standard operating system image	Non-standard operating system image
Description	<p>The supported standard 32-bit and 64-bit operating systems include:</p> <ul style="list-style-type: none"> • CentOS • Ubuntu • SUSE • openSUSE • Red Hat • Debian • CoreOS • Aliyun Linux <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note Support for standard operating systems may be subject to version changes. You can access the ECS console to check the latest supported operating systems.</p> </div>	<p>Non-standard operating systems include:</p> <ul style="list-style-type: none"> • Operating systems that are not supported by Alibaba Cloud. • Standard operating systems that do not meet the requirements of critical system configuration files, basic system environments, and applications. <p>If you want to use non-standard operating system images, you must select Others Linux when importing images. If you import non-standard operating system images, Alibaba Cloud does perform any processing on the instances created from these images. After you create an instance, you must connect to the instance by clicking Connect to VPN in the ECS console. You can then configure the IP address, route, and password for the instance.</p>

Item	Standard operating system image	Non-standard operating system image
Critical system configuration files	<ul style="list-style-type: none"> Do not modify <code>/etc/issue*</code> . Otherwise, the version of the operating system cannot be identified, which leads to system creation failure. Do not modify <code>/boot/grub/menu.lst</code> . Otherwise, the system may fail to start. Do not modify <code>/etc/fstab</code> . Otherwise, partitions cannot be loaded, which leads to system startup failure. Do not modify <code>/etc/shadow</code> to read-only. Otherwise, the password file cannot be modified, which leads to system creation failure. Do not modify <code>/etc/selinux/config</code> to enable SELinux. Otherwise, the system may fail to start. 	Requirements for standard operating system images are not met.
Requirements for the basic system environment	<ul style="list-style-type: none"> Do not adjust the system disk partitions. Only disks with a single root partition are supported. Make sure that the system disk has sufficient storage space. Do not modify critical system files, such as <code>/sbin</code> , <code>/bin</code> , and <code>/lib*</code> . Before importing an image, confirm the integrity of the file system. Only ext3 and ext4 file systems are supported. 	
Applications	Do not install <code>qemu-ga</code> on a custom image. Otherwise, some of the services that Alibaba Cloud needs may be unavailable.	
Image file formats	Only images in the RAW, VHD, or qcow2 format can be imported. If you want to import images in other formats, use a tool to convert the format before importing the image. We recommend that you import images in the VHD format, which has a smaller transmission footprint.	
Image file size	We recommend that you configure the disk size for importing images based on the virtual disk size (not the image file size). The disk size for importing images must be at least 40 GiB.	

Limits on importing custom images in Windows

When you import custom images in Windows, note the following limits:

- The password must be 8 to 30 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- Imported Windows images do not provide the Windows activation service.
- The firewall must be disabled. Otherwise, remote logon is unavailable. Port 3389 must be opened.
- The Windows system disk size ranges from 40 GiB to 500 GiB.

Limits

Item	Description
Operating system versions	<p>Alibaba Cloud supports importing the following versions of 32-bit and 64-bit operating system images:</p> <ul style="list-style-type: none"> • Microsoft Windows Server 2016 • Microsoft Windows Server 2012, including: <ul style="list-style-type: none"> ◦ Microsoft Windows Server 2012 R2 (Standard Edition) ◦ Microsoft Windows Server 2012 (Standard Edition and Datacenter Edition) • Microsoft Windows Server 2008, including: <ul style="list-style-type: none"> ◦ Microsoft Windows Server 2008 R2 (Standard Edition, Datacenter Edition, and Enterprise Edition) ◦ Microsoft Windows Server 2008 (Standard Edition, Datacenter Edition, and Enterprise Edition) • Microsoft Windows Server 2003, including: <ul style="list-style-type: none"> ◦ Microsoft Windows Server 2003 R2 (Standard Edition, Datacenter Edition, and Enterprise Edition) ◦ Microsoft Windows Server 2003 (Standard Edition, Datacenter Edition, and Enterprise Edition) or later, including Service Pack 1 (SP1) • Microsoft Windows 7, including: <ul style="list-style-type: none"> ◦ Microsoft Windows 7 (Professional Edition) ◦ Microsoft Windows 7 (Enterprise Edition) <p> Note Support for standard operating systems may be subject to version changes. You can access the ECS console to check the latest supported operating systems.</p>
Requirements for the basic system environment	<ul style="list-style-type: none"> • Multi-partition system disks are supported. • Make sure that the system disk has sufficient storage space. • Do not modify critical system files. • Before importing an image, confirm the integrity of the file system. • The NTFS file system with the MBR partition type is supported.
Applications	<p>Do not install qemu-ga on an imported image. If it is installed, some of the services that Alibaba Cloud needs may be unavailable.</p>

Item	Description
Image file formats	<ul style="list-style-type: none"> • RAW • VHD • qcow2 <p>We recommend that you configure the system disk size for importing images based on the virtual disk size (not the image file size). The system disk size for importing images must range from 40 GiB to 500 GiB.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note We recommend that you import images in the VHD format, which has a smaller transmission footprint.</p> </div>

2.6.6.2. Convert the image file format

You can only import image files in the RAW, VHD, and qcow2 formats to ECS. If you want to import images in other formats, you must convert the image into a supported format. This topic describes how to convert the image format in Windows and Linux.

Context

You can use the `qemu-img` tool to convert an image from VMDK, VDI, VHDX, qcow1, or QED to RAW, VHD, or qcow2, or implement conversion between RAW, VHD, and qcow2.

 **Note** We recommend that you use the qcow2 format if your application environment supports this format.

Windows

1. Download `qemu`. Visit [QEMU Binaries for Windows \(64 bit\)](#) to download the `qemu` tool. Select a `qemu` version based on your operating system.
2. Install `qemu`. The installation path in this example is `C:\Program Files\qemu`.
3. Configure the environment variables for `qemu`.
 - i. Choose **Start > Computer**, right-click **Computer**, and choose **Properties** from the shortcut menu.
 - ii. In the left-side navigation pane, click **Advanced System Settings**.
 - iii. In the **System Properties** dialog box that appears, click the **Advanced** tab and then click **Environment Variables**.
 - iv. In the **Environment Variables** dialog box that appears, find the **Path** variable from the **System variables** section.
 - If the **Path** variable exists, click **Edit**.
 - If the **Path** variable does not exist, click **New**.
 - v. Add a system variable value.
 - In the **Edit System Variable** dialog box that appears, add `C:\Program Files\qemu` to the **Variable value** field, separate different variable values with semicolons (;), and then click **OK**.
 - In the **New System Variable** dialog box that appears, enter `Path` in the **Variable name** field, enter `C:\Program Files\qemu` in the **Variable value** field, and then click **OK**.
4. Open **Command Prompt** in Windows and run the `qemu-img --help` command. If a successful response is displayed, the tool is installed.
5. In the **Command Prompt** window, run the `cd [Directory of the source image file]` command to switch to a new file directory, for example, `cd D:\ConvertImage`.

- In the Command Prompt window, run the `qemu-img convert -f raw -O qcow2 centos.raw centos.qcow2` command to convert the image file format. The parameters are described as follows:
 - The `-f` parameter is followed by the source image format.
 - The `-O` parameter (case-sensitive) is followed by the destination image format, source file name, and destination file name.

After the conversion is complete, the destination file appears in the directory of the source image file.

Linux

- Install the `qemu-img` tool.
 - For Ubuntu, run the `apt install qemu-img` command.
 - For CentOS, run the `yum install qemu-img` command.
- Run the `qemu-img convert -f raw -O qcow2 centos.raw centos.qcow2` command to convert the image file format.

The parameters are described as follows:

 - The `-f` parameter is followed by the source image format.
 - The `-O` parameter (case-sensitive) is followed by the destination image format, source file name, and destination file name.

2.6.6.3. Import a custom image

After you upload a local image to an OSS bucket, you can import the image to the ECS environment as a custom image.

Prerequisites

- An image that meets the limits and requirements for image import has been made. The image must be in the RAW, VHD, or qcow2 format. For more information, see [Limits on importing custom images](#) and [Convert the image file format](#).
- You have been authorized to import images. For more information, see *the RAM authorization section* in ASCM Console User Guide.
- A local image has been uploaded to a bucket by using the OSS console or calling an OSS API operation. For more information, see *the Upload objects section* in *OSS User Guide* or *the PutObject section* in *OSS Developer Guide*.

 **Note** Make sure that the bucket is in the same region as the custom image you want to create.

Procedure

- [Log on to the ECS console](#).
- In the left-side navigation pane, choose **Snapshots and Images > Images**.
- In the top navigation bar, select an organization, a resource set, and a region.
- Click **Import Image**.
- Configure the parameters of the image.

Parameter	Required	Description
Region	Yes	The region of the custom image to be imported.
Organization	Yes	The organization to which the custom image belongs.

Parameter	Required	Description
Resource Set	Yes	The resource set of the custom image.
OSS Bucket Name	Yes	The name of the OSS bucket where the image to be imported resides.
OSS Object Name	Yes	The endpoint of the OSS object where the image is stored. For information about how to obtain the endpoint of an OSS object, see <i>the Obtain object UR Ls section</i> in OSS User Guide.
Image Name	Yes	The name of the custom image. The name must be 2 to 128 characters in length. It must start with a letter and can contain letters, periods (.), underscores (_), and hyphens (-).
Operating System	Yes	Linux and Windows are available.
System Disk	Yes	The size of the system disk of the ECS instance. Unit: GiB.
System Architecture	Yes	x86_64 and i386 are available.
Platform	Yes	<p>Linux:</p> <ul style="list-style-type: none"> ◦ CentOS ◦ Ubuntu ◦ SUSE ◦ OpenSUSE ◦ Debian ◦ CoreOS ◦ Aliyun ◦ Others Linux ◦ Customized Linux <p>Windows:</p> <ul style="list-style-type: none"> ◦ Windows Server 2003 ◦ Windows Server 2008 ◦ Windows Server 2012
Image Format	Yes	The format of the custom image. RAW, VHD, and qcow2 are available.
Description	No	The description of the custom image.

6. Click **OK**.

Result

You can go to the Images page to view the progress of custom image creation. For more information, see . When 100% is displayed in the Progress column of the Images page, the image is created.

2.6.7. Export a custom image

You can export a custom image to an OSS bucket and then download it to your local device.

Prerequisites

- OSS is activated and an OSS bucket is created. For more information, see the *"Create buckets" section* in OSS

User Guide.

- You are authorized to export images. For more information, see *the "RAM" chapter* in ASCM Console User Guide.

Context

You can export custom images to the RAW, VHD, or qcow2 format. After a custom image is exported to an OSS bucket, you can download the image to your local device. For more information, see *the "Obtain object URLs" section* in OSS User Guide.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target custom image and click **Export Image** in the **Actions** column.
5. Select a bucket for **OssBucket**. Enter a prefix in the **OSS Prefix** field. Then click **OK**. The **OSS Prefix** field is optional. The prefix must be 1 to 30 characters in length and can contain digits and letters.

2.6.8. Delete a custom image

You can delete a custom image that is no longer needed. However, public images cannot be deleted.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Images**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Select one of the following methods to delete custom images.
 - To delete one image, find the image and click **Delete Image** in the **Actions** column.
 - To delete multiple images, select images and click **Delete** at the bottom of the image list.
5. Click **OK**.

2.7. Snapshots

2.7.1. Create a snapshot

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

Prerequisites

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

Context

You can create up to 64 snapshots for each disk.

Snapshots can be used in the following scenarios:

- Restore a disk from one of its snapshots.
- Create a custom image from a system disk snapshot.
 - For more information, see [. Data disk snapshots cannot be used to create custom images.](#)
- Create a new data disk from a data disk snapshot.

To create a data disk from a snapshot, set Use Snapshot to Yes and specify a snapshot on the Create Disk page. For more information, see [Create a disk](#). When you re-initialize a data disk created from a snapshot, the disk is restored to the status of the snapshot.

Note the following considerations when you create a snapshot:

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

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target disk and click **Create Snapshot** in the **Actions** column.
5. Enter a snapshot name and description and then click **OK**.

 **Note** The names of manual snapshots cannot start with auto because auto is a prefix reserved for automatic snapshots.

You can go to the Snapshots page to check the progress of snapshot creation. For more information, see [View snapshots](#). When 100% is displayed in the Progress column, the snapshot is created.

2.7.2. View snapshots

You can view the list of created snapshots.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Snapshots**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The created snapshots that match the specified criteria are displayed.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

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

Filter option	Description
Snapshot Name	Enter a snapshot name to search for the snapshot.
Snapshot ID	Enter a snapshot ID to search for the snapshot.
Instance ID	Enter an instance ID to search for the snapshots related to the instance.
Disk ID	Enter a disk ID to search for the snapshots related to the disk.
Snapshot Type	Select a snapshot type to search for the snapshots of that type. Valid values: <ul style="list-style-type: none"> ○ All ○ User Snapshots: manual snapshots ○ Automatic Snapshots: automatic snapshots

Filter option	Description
Creation Time	Enter a time to search for the snapshots that were created at that time.

2.7.3. Delete a snapshot

You can delete a snapshot that is no longer needed. After the snapshot is deleted, it cannot be recovered. You cannot delete system disk snapshots that have been used to create custom images.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Snapshots**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Use one of the following methods to delete the snapshot:
 - To delete a single snapshot, find the snapshot and click **Delete** in the **Actions** column.
 - To delete one or more snapshots at a time, select the snapshots and click **Delete** in the lower-left corner of the Snapshots page.
5. In the message that appears, click **OK**.

2.8. Automatic snapshot policies

2.8.1. Create an automatic snapshot policy

Automatic snapshot policies can apply to both system disks and data disks and can be used to create periodical snapshots for the disks. Using automatic snapshot policies can improve data availability and operation error tolerance.

Context

Automatic snapshot policies can effectively eliminate the following risks associated with manual snapshot creation:

- When applications such as personal websites or databases deployed on an ECS instance encounter attacks or system vulnerabilities, you may not be able to manually create snapshots. In this case, you can use the latest automatic snapshot to roll back the affected disks to restore your data and reduce losses.
- You can also specify an automatic snapshot policy to create snapshots before regular system maintenance tasks. This eliminates the need to manually create snapshots and ensures that snapshots are always created before maintenance.

You can create up to 64 snapshots for each disk. If the maximum number of snapshots for a disk is reached when a new snapshot is being created, the system deletes the oldest automatic snapshot.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Automatic Snapshot Policies**.
3. Click **Create Policy**.
4. Configure the properties of the automatic snapshot policy.

Parameter	Required	Description
Region	Yes	The ID of the region to which the automatic snapshot policy applies.

Parameter	Required	Description
Organization	Yes	The organization to which the automatic snapshot policy applies.
Policy Name	Yes	The name of the automatic snapshot policy. The name must be 2 to 128 characters in length and cannot start with a special character or digit. It can contain periods (.), underscores (_), hyphens (-), and colons (:).
Creation Time	Yes	<p>The time when a snapshot is automatically created. You can select any hour from 00:00 to 23:00.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin: 5px 0;"> <p> Note The default time zone for the snapshot policy is UTC+8. You can change the time zone based on your business requirements.</p> </div> <p>The creation of an automatic snapshot is canceled if the scheduled time for creating the snapshot is reached but the previous automatic snapshot is still being created. This may occur if the disk stores a large volume of data. For example, you can specify a policy for the system to create automatic snapshots at the following points in time: 00:00, 01:00, and 02:00. When the system starts creating a snapshot at 00:00, it takes 70 minutes for the system to complete the snapshot task. Therefore, the system does not create another snapshot at 01:00. Instead, after the system completes the snapshot task at 01:10, the system creates the next snapshot at 02:00.</p>
Frequency	Yes	The day when a snapshot is created. You can select multiple values. The day ranges from Monday to Sunday.
Retention Policy	No	<p>The retention policy of the automatic snapshot. The default value of the retention time is 30 days. You can configure the following parameters:</p> <ul style="list-style-type: none"> ◦ Keep for: Specify the number of days during which the snapshots can be retained. Valid values: 1 to 65536. ◦ Always keep the snapshots until the number of snapshots reaches the upper limit: Select this option to retain the snapshots until the maximum number of snapshots is reached.

5. Click OK.

What's next

After the automatic snapshot policy is created, you need to apply it to a disk to automatically create snapshots for the disk. For more information, see .

2.8.2. View automatic snapshot policies

You can view the list of automatic snapshot policies.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Automatic Snapshot Policies**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The created automatic snapshot policies that match the specified criteria are displayed.
4. View the list of automatic snapshot policies.

2.8.3. Modify an automatic snapshot policy

You can modify the properties of an automatic snapshot policy, such as the name, creation time, frequency, and retention policy.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Automatic Snapshot Policies**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the automatic snapshot policy and click **Modify Policy** in the **Actions** column.
5. Modify the properties of the policy. Changes made to the retention duration do not take effect on existing snapshots, but take effect only on newly created snapshots.
6. Click **OK**.

2.8.4. Configure an automatic snapshot policy

After you apply an automatic snapshot policy to a disk, snapshots will be created automatically for the disk based on the policy settings. You can cancel an applied automatic snapshot policy at any time.

Context

We recommend that you configure the automatic snapshot policy to create automatic snapshots during off-peak hours. You can also manually create a snapshot for the disk that already has an automatic snapshot policy applied. When an automatic snapshot is being created, you must wait until the snapshot is complete before you can create a manual snapshot. The automatic snapshot is named in the following format: `auto_yyyyMMdd_1`, such as `auto_20140418_1`.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Storage > Disks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the disk and click **Configure Automatic Snapshot Policy** in the **Actions** column.
5. Select a procedure based on the operation you want to perform on the policy.
 - To apply an automatic snapshot policy, turn on **Automatic Snapshot Policy**, select a policy, and then click **OK**.
 - To cancel an automatic snapshot policy, turn off **Automatic Snapshot Policy** and click **OK**.

2.8.5. Configure an automatic snapshot policy for multiple disks

After you apply an automatic snapshot policy to a disk, snapshots will be created automatically for the disk based on the policy settings. You can cancel an applied automatic snapshot policy at any time.

Context

We recommend that you configure the automatic snapshot policy to create automatic snapshots during off-peak hours. You can also manually create a snapshot for the disk that already has an automatic snapshot policy applied. When an automatic snapshot is being created, you must wait until the snapshot is complete before you can create a manual snapshot. The automatic snapshot is named in the following format: `auto_yyyyMMdd_1`, such as `auto_20140418_1`.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Automatic Snapshot Policies**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the automatic snapshot policy and click **Apply Policy** in the **Actions** column.
5. Select a tab based on the operation you want to perform on the disks.
 - To apply the automatic snapshot policy, click the **Disks Without Policy Applied** tab, select one or more disks, and click **Apply Policy** at the bottom of the disk list.
 - To cancel the automatic snapshot policy, click the **Disks With Policy Applied** tab, select one or more disks, and click **Disable Automatic Snapshot Policy** at the bottom of the disk list.

2.8.6. Delete an automatic snapshot policy

You can delete an automatic snapshot policy that is no longer needed. After you delete the automatic snapshot policy, the policy is automatically canceled for disks that have it applied.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Snapshots and Images > Automatic Snapshot Policies**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the automatic snapshot policy and click **Delete Policy** in the **Actions** column.
5. In the message that appears, click **OK**.

2.9. Security groups

2.9.1. Create a security group

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

Prerequisites

A Virtual Private Cloud (VPC) has been created. For more information, see *VPC User Guide*.

Context

Instances that belong to the same account and are in the same region and in the same security group can communicate with each other over the internal network. If instances that belong to the same account in the same region are in different security groups, you can implement internal network communication by authorizing mutual access between two security groups.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.

4. Click **New Security Group**.

5. Configure the parameters of the security group.

Type	Parameter	Required	Description
Region	Organization	Yes	The organization to which the security group belongs. Make sure that the security group and the VPC belong to the same organization.
	Resource Set	Yes	The resource set to which the security group belongs. Make sure that the security group and the VPC belong to the same resource set.
	Region	Yes	The region to which the security group belongs. Make sure that the security group and the VPC belong to the same region.
	Zone	Yes	The ID of the zone where the security group resides.
Basic Settings	VPC	Yes	The VPC to which the security group belongs.
	Security Group Name	No	The name must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). It cannot start with http:// or https://.
	Description	No	The description of the security group. We recommend that you provide an informational description to simplify future management operations. The name must be 2 to 256 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and commas (,). It cannot start with http:// or https://.

6. Click **Submit**.

2.9.2. View security groups

You can view the list of security groups that you create.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region. The created security groups that match the specified criteria are displayed.
4. Select a filtering option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

You can select multiple filtering options to narrow down the search results.

Filtering option	Description
Security Group ID	Enter a security group ID to search for the security group.
Security Group Name	Enter a security group name to search for the security group.
VPC ID	Enter a VPC ID to search for the security groups that belong to the VPC.

2.9.3. Modify a security group

You can modify the name and description of a created security group.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the security group and click **Modify** in the **Actions** column.
5. Modify the name and description of the security group.
6. Click **OK**.

2.9.4. Add a security group rule

You can use security group rules to control access to and from the ECS instances in a security group over the Internet and the internal network.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Rules** in the **Actions** column.
5. Click **Create Rule**.
6. Configure parameters listed in the following table to create a security group rule.

Parameter	Required	Description
ENI Type	Yes	Valid value: Internal Network ENI . In VPCs, you cannot find public NICs in ECS instances and can add only internal security group rules. However, the added security group rules apply to both the Internet and internal network.
Direction	Yes	<ul style="list-style-type: none"> ◦ Outbound: access from the ECS instances in the current security group to other ECS instances on the internal network or resources on the Internet. ◦ Inbound: access from other ECS instances on the internal network or resources on the Internet to the ECS instances in the current security group.
Action	Yes	<ul style="list-style-type: none"> ◦ Allow: allows access requests on the specified port or ports. ◦ Deny: discards data packets and returns no messages. <p>If two security group rules are different only in the Action parameter, the Deny rule takes effect while the Allow rule is ignored.</p>
Protocol	Yes	<ul style="list-style-type: none"> ◦ All: all protocols. This value can be used in total trust scenarios. ◦ TCP: can be used to allow or deny traffic on one or several successive ports. ◦ UDP: can be used to allow or deny traffic on one or several successive ports. ◦ ICMP: can be used when the <code>ping</code> command is used to test the status of the network connection between instances. ◦ ICMPv6: can be used when the <code>ping6</code> command is used to test the status of the network connection between instances. ◦ GRE: can be used for VPN.
Port Range	Yes	<p>The port range depends on the protocol type.</p> <ul style="list-style-type: none"> ◦ When you set Protocol to All, -1/-1 is displayed, indicating all ports. You cannot specify a port range in this case. ◦ When you set Protocol to TCP, you can use the <start port number>/<end port number> format to specify a port range. Valid port numbers: 1 to 65535. Set the start port number and end port number to the same value to specify a single port. For example, use 22/22 to specify port 22. ◦ When you set Protocol to UDP, you can use the <start port number>/<end port number> format to specify a port range. Valid port numbers: 1 to 65535. Set the start port number and end port number to the same value to specify a single port. For example, use 3389/3389 to specify port 3389. ◦ When you set Protocol to ICMP, -1/-1 is displayed, indicating all ports. You cannot specify a port range in this case. ◦ When you set Protocol to ICMPv6, -1/-1 is displayed, indicating all ports. You cannot specify a port range in this case. ◦ When you set Protocol to GRE, -1/-1 is displayed, indicating all ports. You cannot specify a port range in this case.
Priority	Yes	The priority of the rule. Valid values: 1 to 100. The default value is 1, indicating the highest priority.

Parameter	Required	Description
Authorization Type	Yes	<ul style="list-style-type: none"> ◦ IPv4 Addresses: IPv4 addresses or IPv4 CIDR blocks. ◦ IPv6 Addresses: IPv6 addresses or IPv6 CIDR blocks. ◦ Security Groups: another security group. This authorization type takes effect only on the internal network.
Authorization Object	Yes	<p>Authorization objects depend on the authorization type.</p> <p>When you set Authorization Type to IPv4 Addresses:</p> <ul style="list-style-type: none"> ◦ Enter an IPv4 address or IPv4 CIDR block. Example: <i>12.1.1.1</i> or <i>13.1.1.1/25</i>. ◦ You can enter up to 10 authorization objects at a time. Separate multiple objects with commas (,). ◦ If you specify <i>0.0.0.0/0</i>, all IPv4 addresses will be allowed or denied based on the Action parameter. Exercise caution when you specify <i>0.0.0.0/0</i>. <p>When you set Authorization Type to IPv6 Addresses:</p> <ul style="list-style-type: none"> ◦ Enter an IPv6 address or IPv6 CIDR block. Example: <i>2001:0db8::1428:****</i> or <i>2001:0db8::1428:****/128</i>. ◦ You can enter up to 10 authorization objects at a time. Separate multiple objects with commas (,). ◦ If you specify <i>::/0</i>, all IPv6 addresses will be allowed or denied based on the Action parameter. Exercise caution when you specify <i>::/0</i>. <p>When you set Authorization Type to Security Groups, select a security group ID. If the current security group is of the VPC type, the selected security group must be in the same VPC as the current security group.</p>
Description	No	<p>The description of the security group rule. We recommend that you provide an informational description to simplify future management operations. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code>.</p>

7. Click OK.

2.9.5. Clone a security group rule

You can clone a security group rule to quickly create a similar rule.

Procedure

1. [Log on to the ECS console.](#)
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Find the target security group rule and click **Clone** in the **Actions** column.
7. In the **Clone Security Group Rule** dialog box, modify the attributes of the security group rule. For more information about the attributes of security group rules, see [Add a security group rule.](#)
8. Click OK.

2.9.6. Modify a security group rule

You can modify improper rules in a security group to ensure the security of ECS instances in the security group.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Find the target security group rule and click **Modify** in the **Actions** column.
7. In the **Modify Security Group Rule** dialog box, modify the attributes of the security group rule. For more information about the attributes of security group rules, see [Add a security group rule](#).
8. Click **OK**.

2.9.7. Export security group rules

You can export security group rules of a security group to a local device for backup.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Click **Export** in the upper-right corner to download and save the rules to a local device.

2.9.8. Import security group rules

You can import a local backup file of security group rules into a security group to quickly create or restore security group rules.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Rules** in the **Actions** column.
5. On the Rules page that appears, click the **Inbound** or **Outbound** tab.
6. Click **Import** in the upper-right corner.
7. In the **Import Rule** dialog box, click **Choose File**.
8. Select the target local backup file of security group rules and click **Open**. Then, click **OK**. The local backup file must be in the CSV format. You can download a template file from the **Import Rule** dialog box.

2.9.9. Add an instance

You can add an existing instance to a security group in the same region. After the instance is added, the security group rules of the security group automatically apply to the instance.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.

4. Find the security group and click **Manage Instances** in the **Actions** column.
5. Click **Add Instance**.
6. Select an instance and click **OK**.

2.9.10. Remove instances from a security group

You can remove instances from a security group, but each of the instances must always belong to at least one security group.

Prerequisites

The instances to be removed are added to two or more security groups.

Context

After an ECS instance is removed from a security group, the instance will be isolated from the other ECS instances in the security group. We recommend that you perform a full test in advance to ensure that services can run properly after you remove the instance.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target security group and click **Manage Instances** in the **Actions** column.
5. On the **Instances** page that appears, select one or more instances and click **Remove** in the lower-left corner.
6. Click **OK**.

2.9.11. Delete a security group

You can delete a security group that is no longer needed.

Prerequisites

No instances exist in the security group.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > Security Groups**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Use one of the following methods to delete the security group:
 - To delete a single security group, find the security group and click **Delete** in the **Actions** column.
 - To delete one or more security groups at a time, select the security groups and click **Delete** in the lower-left corner of the **Security Groups** page.
5. In the message that appears, click **OK**.

2.10. Elastic Network Interfaces

2.10.1. Create an ENI

You can bind elastic network interfaces (ENIs) to instances to create high-availability clusters and implement fine-grained network management. You can also unbind an ENI from an instance and then bind the ENI to another instance to implement a low-cost failover solution.

Prerequisites

- A virtual private cloud (VPC) and a VSwitch are created. For more information, see [Create a VPC](#) and [Create a VSwitch](#) in *Apsara Stack VPC User Guide* .
- A security group is available in the VPC. If no security group is available in the VPC, create a security group. For more information, see [Create a Security Group](#) .

Context

ENIs are classified into primary and secondary ENIs.

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

ENIs created separately are secondary ENIs. You can bind secondary ENIs to or unbind them from instances. This topic describes how to create a secondary ENI.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **Create ENI**.
5. Configure parameters listed in the following table to create an ENI.

Section	Parameter	Required	Description
Region	Organization	Yes	The organization in which to create the ENI.
	Resource Set	Yes	The resource set in which to create the ENI.
	Region	Yes	The region in which to create the ENI.
	Zone	Yes	The zone in which to create the ENI.
	VPC	Yes	The VPC in which to create the ENI. The secondary ENI can be bound only to an instance in the same VPC. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note After the ENI is created, you cannot change its VPC. </div>

Section	Parameter	Required	Description
Basic Settings	VSwitch	Yes	<p>The VSwitch to be associated with the ENI. The secondary ENI can be bound only to an instance in the same VPC. Select a VSwitch that is in the same zone as the instance to which the ENI will be bound. The VSwitch of the ENI can be different from that of the instance.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note After an ENI is created, you cannot change its VSwitch.</p> </div>
	Security Group	Yes	<p>The security group in which to create the ENI within the specified VPC. The rules of the security group automatically apply to the ENI.</p>
	ENI Name	Yes	<p>The name of the ENI. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).</p>
	Description	No	<p>The description of the ENI. We recommend that you provide an informational description to simplify future management operations. The description must be 2 to 256 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).</p>

Section	Parameter	Required	Description
	Primary Private IP	No	The primary private IPv4 address of the ENI. The IPv4 address must be within the CIDR block of the specified VSwitch. If you do not specify a primary private IP address, the system automatically assigns a private IP address to the ENI.

6. Click **Submit**.

Result

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

2.10.2. View ENIs

You can view the list of created ENIs.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The created ENIs that match the specified criteria are displayed.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and click **Search**.

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

Filter option	Description
ENI Name	Enter an ENI name to search for the ENI.
ENI ID	Enter an ENI ID to search for the ENI.
VSwitch ID	Enter a VSwitch ID to search for the ENIs that are associated with the VSwitch.
Security Group ID	Enter a security group ID to search for the ENIs that belong to the security group.
Instance ID	Enter an instance ID to search for the ENIs that are bound to the instance.

2.10.3. Modify a secondary ENI

You can modify the attributes of a secondary elastic network interface (ENI), including the name, security group, and description.

Prerequisites

The secondary ENI is in the **Available** state.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the secondary ENI and click **Modify** in the **Actions** column.
5. In the Modify ENI dialog box that appears, modify the name, security group, and description of the ENI.
6. Click **OK**.

2.10.4. Bind a secondary ENI to an instance

You can bind a secondary elastic network interface (ENI) to an instance. After the ENI is bound to the instance, the instance can process the traffic on the ENI.

Prerequisites

- The secondary ENI is in the **Available** state.
- The instance to which you want to bind the secondary ENI is in the **Running** or **Stopped** state.
- The instance and the secondary ENI belong to the same VPC.
- The VSwitch with which the secondary ENI is associated is in the same zone as the VSwitch to which the instance is connected. An ENI can be bound only to an instance in the same zone. The VSwitches of the ENI and of the instance can be different but must be in the same zone.

Context

The following limits apply when you bind an ENI to an instance:

- You can manually bind only secondary ENIs. Primary ENIs share the same lifecycle as instances and cannot be manually bound.
- An ENI can only be bound to a single ECS instance. However, an ECS instance can be bound with multiple ENIs. The maximum number of ENIs that can be bound to an instance depends on the instance type. For more information about the number of ENIs that can be bound to an instance of each instance type, see *Instance families in ECS Product Introduction*.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the target secondary ENI and click **Bind** in the **Actions** column.
5. In the Bind dialog box that appears, select an instance and click **OK**.

Result

In the **Status/Creation Time** column, the status of the secondary ENI changes to **Bound**.

2.10.5. Unbind a secondary ENI from an instance

You can unbind a secondary elastic network interface (ENI) from an instance. After the secondary ENI is unbound from the instance, the instance no longer processes the traffic on the ENI.

Prerequisites

- The secondary ENI is in the **Bound** state.
- The instance is in the **Running** or **Stopped** state.

Context

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

Procedure

1. [Log on to the ECS console.](#)
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the secondary ENI and click **Unbind** in the **Actions** column.
5. Click **OK**.

Result

In the **Status/Creation Time** column, the status of the secondary ENI changes to **Available**.

2.10.6. Delete a secondary ENI

You can delete a secondary elastic network interface (ENI) that is no longer needed.

Prerequisites

The secondary ENI is in the **Available** state.

Context

You can delete only secondary ENIs. Primary ENIs share the same lifecycle as instances and cannot be deleted.

Procedure

1. [Log on to the ECS console.](#)
2. In the left-side navigation pane, choose **Networks and Security > ENIs**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the secondary ENI and click **Delete** in the **Actions** column.
5. Click **OK**.

2.11. Deployment sets

2.11.1. Create a deployment set

You can use a deployment set to distribute or aggregate instances involved in your business. You can select hosts, racks, or network switches to improve service availability or network performance based on your needs.

Procedure

1. [Log on to the ECS console.](#)
2. In the left-side navigation pane, click **Deployment Sets**.
3. Click **Create Deployment Set**.
4. Configure the parameters of the deployment set.

Category	Parameter	Required	Description
Region	Organization	Yes	The organization to which the deployment set belongs.
	Resource Set	Yes	The resource set to which the deployment set belongs.
	Region	Yes	The region where the deployment set is located.
	Zone	Yes	The zone where the deployment set is located.

Category	Parameter	Required	Description
Basic Settings	Deployment Domain	Yes	This parameter setting determines the Deployment Target options. Valid values: <ul style="list-style-type: none"> ◦ Default: When Default is selected, the deployment target options are Host, Rack, and Network Switch. ◦ Switch: When Switch is selected, the deployment target options are Host and Rack.
	Deployment Target	Yes	The basic unit that can be scheduled when you deploy instances. <ul style="list-style-type: none"> ◦ Host: Instances are distributed or aggregated at the host level. ◦ Rack: Instances are distributed or aggregated at the rack level. ◦ VSwitch: Instances are distributed or aggregated at the VSwitch level.
	Deployment Policy	Yes	The dispersion policies are used to improve service availability to avoid business impact when a host, rack, or switch fails. The aggregation policies are used to improve network performance to minimize the access latency between instances. Options are: <ul style="list-style-type: none"> ◦ Loose Dispersion ◦ Strict Dispersion ◦ Loose Aggregation ◦ Strict Aggregation
	Deployment Set Name	No	The name of the deployment set. The name must be 2 to 128 characters in length. It must start with a letter but cannot start with http:// or https://. It can contain digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).
	Description	No	The description of the deployment set. We recommend that you provide an informational description to simplify future management operations. The description must be 2 to 256 characters in length. It must start with a letter but cannot start with http:// or https://. It can contain digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,).

5. Click **Submit**.

2.11.2. View deployment sets

You can view the list of created deployment sets.

Procedure

1. **Log on to the ECS console.**
2. In the left-side navigation pane, click **Deployment Sets**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The created deployment sets that match the specified criteria are displayed.
4. Select a filter option from the drop-down list, enter the relevant information in the search bar, and then

click **Search**.

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

Filter option	Description
Deployment Set Name	Enter a deployment set name to search for the deployment set.
Deployment Set ID	Enter a deployment set ID to search for the deployment set.
Resource Set	Enter a resource set name to search for the deployment sets that belong to the resource set.

2.11.3. Modify a deployment set

You can modify the name and description of a deployment set.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Deployment Sets**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the deployment set and click **Modify** in the **Actions** column.
5. In the Change Deployment Set dialog box, change the name of the deployment set.
6. Click **OK**.

2.11.4. Delete a deployment set

You can delete a deployment set that is no longer needed.

Prerequisites

No instances exist in the deployment set.

Procedure

1. [Log on to the ECS console](#).
2. In the left-side navigation pane, click **Deployment Sets**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the deployment set and click **Delete** in the **Actions** column.
5. Click **OK**.

2.12. Install FTP software

2.12.1. Overview

File Transfer Protocol (FTP) transfers files between a client and a server by establishing two TCP connections. One is the command link for transferring commands between a client and a server. The other is the data link used to upload or download data. Before uploading files to an instance, you must build an FTP site for the instance.

2.12.2. Install and configure vsftpd in CentOS

This topic describes how to install and configure vsftpd in CentOS to transfer files.

Procedure

1. Install vsftpd.

```
yum install vsftpd -y
```

2. Add an FTP account and a directory.

- i. Check the location of the *nologin* file, which is usually under the */usr/sbin* or */sbin* directory.
- ii. Create an FTP account. Run the following commands to create the */alidata/www/wwwroot* directory and specify this directory as the home directory of the account *pwftp*. You can also customize the account name and directory.

```
mkdir -p /alidata/www/wwwroot  
useradd -d /alidata/www/wwwroot -s /sbin/nologin pwftp
```

iii. Modify the account password.

```
passwd pwftp
```

iv. Modify the permissions on the specified directory.

```
chown -R pwftp.pwftp /alidata/www/wwwroot
```

3. Configure vsftpd.

i. Open the vsftpd configuration file.

```
vi /etc/vsftpd/vsftpd.conf
```

ii. Change the value of `anonymous_enable` from `YES` to `NO`.

iii. Delete the comment delimiter (`#`) from the following configuration lines:

```
local_enable=YES  
write_enable=YES  
chroot_local_user=YES
```

iv. Press the Esc key to exit the edit mode, and enter `:wq` to save the modifications and exit.

4. Modify the shell configuration.

i. Open the shell configuration file.

```
vi /etc/shells
```

ii. If the file does not contain */usr/sbin/nologin* or */sbin/nologin*, add it to the file.

5. Start vsftpd and perform a logon test.

i. Start vsftpd.

```
service vsftpd start
```

ii. Use the account *pwftp* to perform an FTP logon test. This example uses the directory */alidata/www/wwwroot*.

2.12.3. Install vsftpd in Ubuntu or Debian

This topic describes how to install and configure vsftpd in an instance running Ubuntu or Debian to transfer files.

Procedure

1. Update the software source.

```
apt-get update
```

2. Install vsftpd.

```
apt-get install vsftpd -y
```

3. Add an FTP account and a directory.

- i. Check the location of the *nologin* file, which is typically under the */usr/sbin* or */sbin* directory.
- ii. Create an FTP account. Run the following commands to create the */alidata/www/wwwroot* directory and specify this directory as the home directory of the account *pwftp*. You can also customize the account name and directory.

```
mkdir -p /alidata/www/wwwroot
useradd -d /alidata/www/wwwroot -s /sbin/nologin pwftp
```

iii. Modify the account password.

```
passwd pwftp
```

iv. Modify the permissions on the specified directory.

```
chown -R pwftp.pwftp /alidata/www/wwwroot
```

4. Configure vsftpd.

i. Open the vsftpd configuration file.

```
vi /etc/vsftpd.conf
```

- ii. Change the value of `anonymous_enable` from `YES` to `NO`.
- iii. Delete the comment delimiter (`#`) from the following configuration lines:

```
local_enable=YES
write_enable=YES
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
```

- iv. Press the Esc key to exit the edit mode, and enter `:wq` to save the modifications and exit.
- v. Open the */etc/vsftpd.chroot_list* file and add the FTP account name to the file. Save the modifications and exit. You can follow steps a to d to open and save the file.

5. Modify shell configurations.

i. Open the shell configuration file.

```
vi /etc/shells
```

- ii. If the file does not contain */usr/sbin/nologin* or */sbin/nologin*, add it to the file.

6. Start vsftpd and perform a logon test.

i. Start vsftpd.

```
service vsftpd restart
```

- ii. Use the account *pwftp* to perform an FTP logon test. This example uses the directory */alidata/www/wwwroot*.

2.12.4. Build an FTP site in Windows Server 2008

This topic describes how to build an FTP site on an instance running Windows Server 2008.

Prerequisites

You have added the Web Server (IIS) role and installed FTP on an instance.

Procedure

1. **Connect to an instance.**
2. Choose **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
3. Right-click the server name and select **Add FTP Site** from the shortcut menu.
4. Enter an FTP site name and a physical path, and then click **Next**.
5. Set **IP Address** to **All Unassigned** and **SSL** to **No SSL**, and then click **Next**.
6. Set **Authentication** to **Basic**, **Authorization** to **All Users**, and **Permissions** to **Read and Write**, and click **Finish**.

Result

Then you can use the administrator account and password to upload and download files through FTP. Make sure that the following conditions are met:

- The port for the FTP site is not in use by other applications, and Windows firewall is not blocking the port.
- The security group of the instance contains a security group rule that allows inbound access to the FTP port.

2.12.5. Build an FTP site in Windows Server 2012

This topic describes how to build an FTP site on an instance running Windows Server 2012.

Prerequisites

You have added the Web Server (IIS) role and installed FTP on an instance.

Procedure

1. **Connect to an instance.**
2. Click the **Server Manager** icon.
3. In the left-side navigation pane, click **IIS**.
4. In the **Server** area, right-click the server name and select **Internet Information Services (IIS) Manager** from the shortcut menu.
5. Right-click the server name and select **Add FTP Site** from the shortcut menu.
6. Enter an FTP site name and a physical path, and then click **Next**.
7. Set **IP Address** to **All Unassigned** and **SSL** to **No SSL**, and then click **Next**.
8. Set **Authentication** to **Basic**, **Authorization** to **All Users**, and **Permissions** to **Read and Write**, and click **Finish**.

Result

Then you can use the administrator account and password to upload and download files through FTP. Make sure that the following conditions are met:

- The port for the FTP site is not in use by other applications, and Windows firewall is not blocking the port.
- The security group of the instance contains a security group rule that allows inbound access to the FTP port.

3. Container Service for Kubernetes

3.1. What is Container Service?

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

Container Service simplifies the deployment and scaling operations on Kubernetes clusters. Integrated with services such as virtualization, storage, network, and security, Container Service aims to provide the optimal cloud environment for Kubernetes containerized applications. Alibaba Cloud is a Kubernetes Certified Service Provider (KCSP). As one of the first services to participate in the Certified Kubernetes Conformance Program, Container Service provides you with professional support and services.

3.2. Planning and preparation

Before you start using Container Service, you need to create cloud resources such as VPC networks, VSwitches, disks, and OSS buckets based on your application requirements.

Before you create a Kubernetes cluster, make the following preparations:

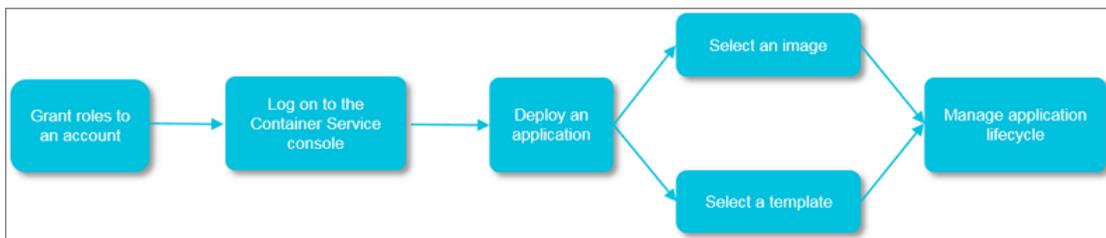
- **Create a VPC network (optional)**
To create a cluster in an existing VPC network, you must create the VPC network and VSwitches in advance.
- **Create a volume (optional)**
To create a stateful application with network storage, you must create disks or OSS buckets in advance.

3.3. Quick start

3.3.1. Flowchart

You can take the following steps to use Container Service.

The flowchart is as follows:



Step 1: Authorize the default role

Authorize the Container Service default role to perform operations on the resources under the target organization.

Step 2: Log on to the Container Service console

Log on to the Container Service console. For more information, see [Log on to the Container Service console](#).

Step 3: Create a cluster

Set the network environment and the number of nodes, and configure node details.

Step 4: Deploy an application by using an image or orchestration template

You can use an existing image or orchestration template, or create a new image or orchestration template.

To create an application that consists of services based on different images, use an orchestration template.

Step 5: Manage the application lifecycle

3.3.2. Log on to the Container Service for Kubernetes console

You can perform the following steps to log on to the Container Service for Kubernetes console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Elastic Computing > Container Service for Kubernetes**.
5. Select the target organization and region.
6. Click **ACK** to go to the Container Service for Kubernetes console.

3.3.3. Log on to the Container Registry console

You can perform the following steps to log on to the Container Registry console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Elastic Computing > ACK-CR**.
5. Select the target organization and region.
6. Click **CR** to go to the Container Registry console.

3.3.4. Create a Kubernetes cluster

To create a Kubernetes cluster, you need to set a series of parameters. For more information, see [Cluster parameters](#).

Procedure

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Clusters > Clusters**. On the Clusters page, click **Create Kubernetes Cluster** in the upper-right corner.
3. Configure basic parameters.

Cluster parameters

Parameter	Description
Cluster Name	<p>The name of the cluster. The name must be 1 to 63 characters in length and can contain digits, Chinese characters, letters, and hyphens (-).</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Note The cluster name must be unique under one account.</p> </div>
Region	The region where the cluster is deployed.
Zone	The zone where the cluster belongs.

Parameter	Description
VPC	<p>You can select a VPC network from the drop-down list.</p> <ul style="list-style-type: none"> ◦ If the VPC network that you select already has a NAT gateway, Container Service will use this NAT gateway. ◦ Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear the Configure SNAT for VPC check box. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Note In this case, you need to manually create a NAT gateway or configure SNAT rules to enable Internet access to the VPC network. Otherwise, cluster creation will fail.</p> </div>
Node Type	Currently, only pay-as-you-go nodes are supported.
Master Configuration	<p>Select the instance types and system disk type.</p> <ul style="list-style-type: none"> ◦ Instance Type: You can select multiple instance types. For more information, see the <i>Instance types</i> chapter of ECS User Guide. ◦ System Disk: SSD and ultra disks are supported. ◦ Quantity: You can add three master nodes.
Worker Instance	You can choose to create instances or add existing instances.
Worker Configuration	<p>If you choose Create Instance, you need to set the following parameters:</p> <ul style="list-style-type: none"> ◦ Instance Type: You can select multiple instance types. For more information, see the <i>Instance types</i> chapter of ECS User Guide. ◦ Quantity: Set the number of worker nodes. ◦ System Disk: SSD and ultra disks are supported. ◦ Attach Data Disk: SSD, ultra, and basic disks are supported.
Docker Version and Kubernetes Version	Supported Docker and Kubernetes versions are displayed. Select based on your needs.
Password	<p>Set the node logon password.</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Note The password must be 8 to 30 characters in length and contain three of the following four types of characters: uppercase letters, lowercase letters, digits, and special characters.</p> </div>
Confirm Password	Enter the logon password again.

Parameter	Description
Network Plug-in	Flannel and Terway are supported. Flannel is enabled by default.
Pod CIDR Block and Service CIDR (Optional)	For more information, see the <i>Network planning</i> chapter of VPC User Guide. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note These options are available when you select an existing VPC network.</p> </div>
Configure SNAT	Optional. If you clear this check box, you need to create a NAT gateway or configure SNAT rules to enable Internet access to the VPC network.
Public SLB	If you select this check box, a public SLB instance is created and the 6443 port used by the API server is enabled on master nodes. You can use kubeconfig to connect to the cluster through the Internet. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note If you clear the Expose API Server with Public SLB check box, you cannot access the cluster API server through the Internet.</p> </div>
SSH logon	<ul style="list-style-type: none"> ◦ If you select this check box, you can use SSH to access the cluster. ◦ If you clear this check box, you cannot use SSH or kubectl to access the cluster.
Log Service	If you enable Log Service, select an existing project or create a new one. The Log Service plug-in is automatically installed in the cluster.
Delete Protection	If you select this check box, the cluster cannot be deleted through the console or API operations.
RDS Whitelist	Add the IP addresses of nodes to the RDS whitelist.
Node Protection	This check box is selected by default to prevent nodes from being deleted through the console or APIs.
Labels	Attach labels to the nodes.
Advanced Options	<ul style="list-style-type: none"> ◦ Pods on Each Node: The maximum number of pods that can run on a single node. ◦ Cluster Domain: Default is cluster.local. Custom domains are supported. ◦ Cluster CA: Set whether to enable custom cluster CA.

4. Click **Create Cluster** in the upper-right corner.
5. On the **Confirm** page, click **OK** to start the deployment.

Result

You can find the newly created cluster on the Clusters page.

3.3.5. Create an application from an orchestration template

Container Service provides orchestration templates that you can use to create applications quickly. You can also modify the templates based on YAML syntax to customize applications.

Prerequisites

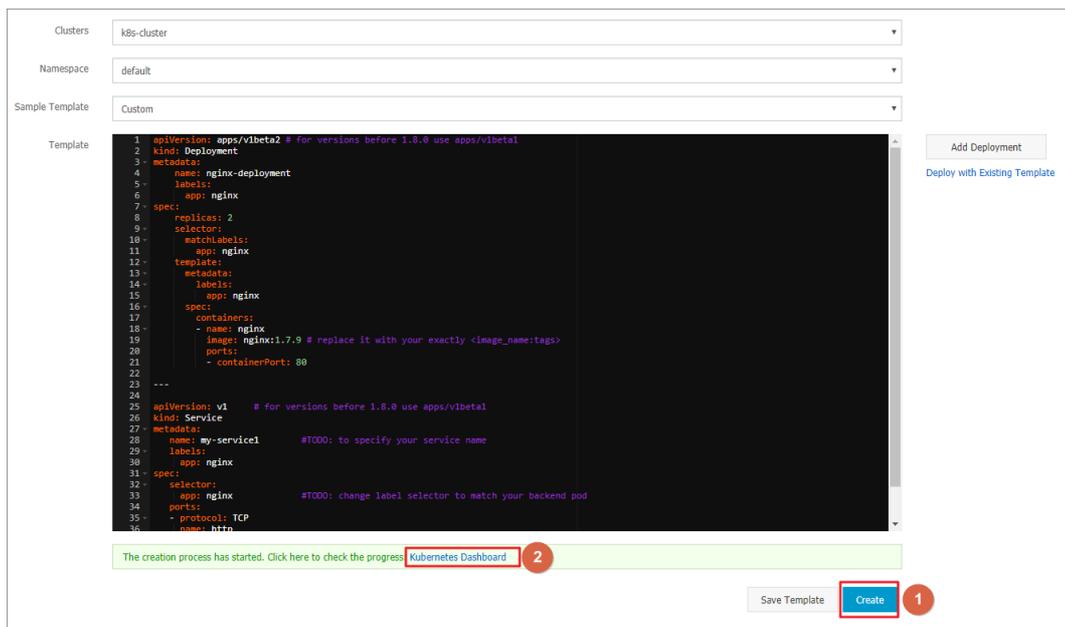
You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

Context

The following example demonstrates how to create an NGINX application consisting of a deployment and a service. The service is associated with a pod created by the deployment.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. In the upper-right corner, click **Create from Template**.
4. Set the parameters and click **Create**.
 - **Cluster:** Select the cluster where the resource objects are to be deployed.
 - **Namespace:** Select the namespace to which the resource objects belong. The default namespace is default. Except for underlying computing resources such as nodes and PVs, most resources are scoped to namespaces.
 - **Sample Template:** Container Service provides YAML templates of various resource types to help you deploy resource objects quickly. You can also create a custom template based on YAML syntax to describe the resource that you want to define.
 - **Add Deployment:** This feature allows you to quickly define a YAML template.
 - **Use Existing Template:** You can import an existing template to the configuration page.



Based on an orchestration template provided by Container Service, the following sample template creates a deployment of an NGINX application.

 **Note** Container Service supports YAML syntax. You can use the `---` symbol to separate multiple resource objects. This enables you to create multiple resource objects in a single template.

```

apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80

---

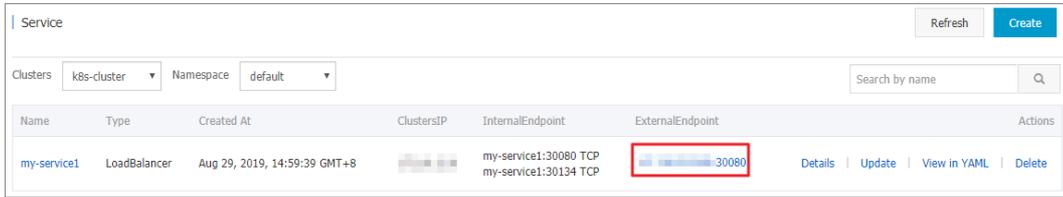
apiVersion: v1 # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
  name: my-service1 #TODO: to specify your service name
  labels:
    app: nginx
spec:
  selector:
    app: nginx #TODO: change label selector to match your backend pod
  ports:
  - protocol: TCP
    name: http
    port: 30080 #TODO: choose an unique port on each node to avoid port conflict
    targetPort: 80
  type: LoadBalancer ## This example changes the service type from NodePort to LoadBalancer.

```

5. Click **Create**. A message appears indicating the deployment status.

In the left-side navigation pane, choose **Ingresses and Load Balancing** > **Services** to view the newly created service.

- On Kubernetes Dashboard, verify that a my-service1 service is running and its external endpoint is displayed. Click the address in the External Endpoint column.



- You can visit the NGINX welcome page in the browser.



What's next

You can also choose **Ingresses and Load Balancing > Services** in the left-side navigation pane to view the NGINX service.

3.4. Kubernetes clusters

3.4.1. Authorizations

3.4.1.1. Assign RBAC permissions to a RAM user

This topic describes how to assign role-based access control (RBAC) permissions to Resource Access Management (RAM) users. By default, RBAC policies are enabled for Kubernetes 1.6 and later versions. RBAC policies are important for you to improve the management of clusters. You can use RBAC policies to specify the types of operations that are allowed for specific users based on their roles in an organization.

Procedure

- Log on to the Container Service console.
- In the left-side navigation pane, choose **Clusters > Authorizations** to go to the Authorizations page.
- Choose **Select RAM User > RAM Users**, select the required RAM user, and then click **Modify Permissions**.

Note Before you authorize a RAM user, make sure that the RAM user has been granted the RBAC administrator permissions or the cluster-admin role for the specified cluster.

- On the **Configure Role-Based Access Control (RBAC)** wizard page, click the plus sign (+) to add permissions on one or all clusters and namespaces, select a predefined role in the **Permission** field, or click the minus sign (-) for a target role to delete the role, and then click **Next Step**.

Note You can add permissions of a predefined role and one or more custom roles on a specified cluster or namespace.

The following table defines the permissions of predefined roles on one or all clusters and namespaces.

Roles and permissions

Role	RBAC-based permission on one or all clusters
Administrator	Granted the read and write permissions on resources in all namespaces.
O&M Engineer	Granted the read and write permissions on resources in all namespaces and the read-only permissions on nodes, persistent volumes (PVs), namespaces, and quotas.
Developer	Granted the read and write permissions on resources in one or all namespaces.
Restricted User	Granted the read-only permissions on resources in one or all namespaces.
Custom	Different cluster roles have different permissions. Before you authorize a RAM user, make sure that you are aware of all resource access permissions of the selected cluster roles. This avoids unnecessary permissions granted to the RAM user.

After the authorization, you can use the specified RAM user to log on to the Container Service for Kubernetes console and manage the service. For more information, see [Log on to the Container Service console](#).

Custom permissions

Container Service for Kubernetes supports predefined roles to grant different permissions. These predefined roles include: administrator, O&M engineer, developer, and restricted user. These predefined roles meet most of your requirements when you manage the service in the console. If you want to customize permissions on clusters, you can use custom roles.

Container Service for Kubernetes provides multiple custom roles.

 **Note** The cluster-admin role has permissions of a super administrator of clusters and has permissions on all resources.

You can log on to the master node of the specified cluster and run the following command to view the details of custom permissions.

```
# kubectl get clusterrole
```

```
# kubectl get clusterrole
NAME                                AGE
admin                                13d
alibaba-log-controller              13d
alicloud-disk-controller-runner    13d
cluster-admin                       13d
cs:admin                            13d
edit                                13d
flannel                             13d
kube-state-metrics                 22h
node-exporter                      22h
prometheus-k8s                    22h
prometheus-operator                22h
system:aggregate-to-admin          13d
....
system:volume-scheduler            13d
view                                13d
```

In this example, the cluster-admin role is used. On the command line, run the following command to view the permission details:

```
# kubectl get clusterrole cluster-admin -o yaml
```

 **Notice** After a RAM user is granted the permissions of cluster-admin, for the specified cluster, the RAM user has the same permissions as your Alibaba Cloud account and has all permissions on all resources in the cluster. Proceed with caution.

```
# kubectl get clusterrole cluster-admin -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  creationTimestamp: 2018-10-12T08:31:15Z
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: cluster-admin
  resourceVersion: "57"
  selfLink: /apis/rbac.authorization.k8s.io/v1/clusterroles/cluster-admin
  uid: 2f29f9c5-cdf9-11e8-84bf-00163e0b2f97
rules:
- apiGroups:
  - "*"
  resources:
  - "*"
  verbs:
  - "*"
- nonResourceURLs:
  - "*"
  verbs:
  - "*"
```

3.4.2. Clusters

3.4.2.1. Create a Kubernetes cluster

To create a Kubernetes cluster, you need to set a series of parameters. For more information, see [Cluster parameters](#).

Procedure

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Clusters > Clusters**. On the Clusters page, click **Create Kubernetes Cluster** in the upper-right corner.
3. Configure basic parameters.

Cluster parameters

Parameter	Description
Cluster Name	<p>The name of the cluster. The name must be 1 to 63 characters in length and can contain digits, Chinese characters, letters, and hyphens (-).</p> <p> Note The cluster name must be unique under one account.</p>
Region	The region where the cluster is deployed.
Zone	The zone where the cluster belongs.
VPC	<p>You can select a VPC network from the drop-down list.</p> <ul style="list-style-type: none"> ◦ If the VPC network that you select already has a NAT gateway, Container Service will use this NAT gateway. ◦ Otherwise, the system automatically creates a NAT gateway. If you do not want the system to create a NAT gateway, clear the Configure SNAT for VPC check box. <p> Note In this case, you need to manually create a NAT gateway or configure SNAT rules to enable Internet access to the VPC network. Otherwise, cluster creation will fail.</p>
Node Type	Currently, only pay-as-you-go nodes are supported.
Master Configuration	<p>Select the instance types and system disk type.</p> <ul style="list-style-type: none"> ◦ Instance Type: You can select multiple instance types. For more information, see the <i>Instance types</i> chapter of ECS User Guide. ◦ System Disk: SSD and ultra disks are supported. ◦ Quantity: You can add three master nodes.
Worker Instance	You can choose to create instances or add existing instances.
Worker Configuration	<p>If you choose Create Instance, you need to set the following parameters:</p> <ul style="list-style-type: none"> ◦ Instance Type: You can select multiple instance types. For more information, see the <i>Instance types</i> chapter of ECS User Guide. ◦ Quantity: Set the number of worker nodes. ◦ System Disk: SSD and ultra disks are supported. ◦ Attach Data Disk: SSD, ultra, and basic disks are supported.
Docker Version and Kubernetes Version	Supported Docker and Kubernetes versions are displayed. Select based on your needs.

Parameter	Description
Password	<p>Set the node logon password.</p> <p>Note The password must be 8 to 30 characters in length and contain three of the following four types of characters: uppercase letters, lowercase letters, digits, and special characters.</p>
Confirm Password	Enter the logon password again.
Network Plug-in	Flannel and Terway are supported. Flannel is enabled by default.
Pod CIDR Block and Service CIDR (Optional)	<p>For more information, see the <i>Network planning</i> chapter of VPC User Guide.</p> <p>Note These options are available when you select an existing VPC network.</p>
Configure SNAT	Optional. If you clear this check box, you need to create a NAT gateway or configure SNAT rules to enable Internet access to the VPC network.
Public SLB	<p>If you select this check box, a public SLB instance is created and the 6443 port used by the API server is enabled on master nodes. You can use kubeconfig to connect to the cluster through the Internet.</p> <p>Note If you clear the Expose API Server with Public SLB check box, you cannot access the cluster API server through the Internet.</p>
SSH logon	<ul style="list-style-type: none"> If you select this check box, you can use SSH to access the cluster. If you clear this check box, you cannot use SSH or kubectl to access the cluster.
Log Service	If you enable Log Service, select an existing project or create a new one. The Log Service plug-in is automatically installed in the cluster.
Delete Protection	If you select this check box, the cluster cannot be deleted through the console or API operations.
RDS Whitelist	Add the IP addresses of nodes to the RDS whitelist.
Node Protection	This check box is selected by default to prevent nodes from being deleted through the console or APIs.
Labels	Attach labels to the nodes.

Parameter	Description
Advanced Options	<ul style="list-style-type: none"> Pods on Each Node: The maximum number of pods that can run on a single node. Cluster Domain: Default is cluster.local. Custom domains are supported. Cluster CA: Set whether to enable custom cluster CA.

4. Click **Create Cluster** in the upper-right corner.
5. On the **Confirm** page, click **OK** to start the deployment.

Result

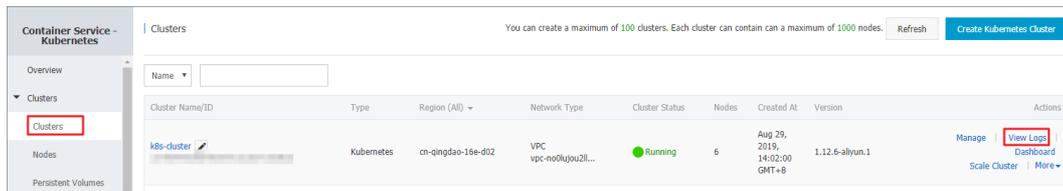
You can find the newly created cluster on the **Clusters** page.

3.4.2.2. View cluster logs

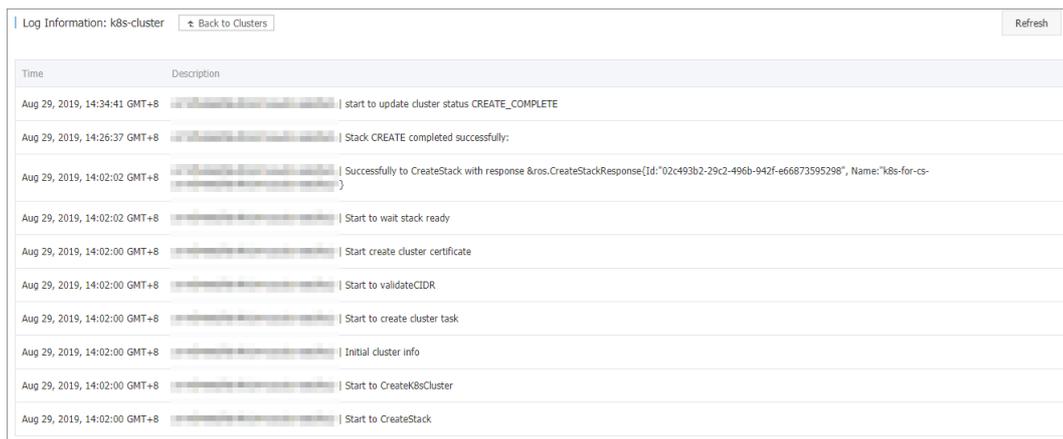
You can view operation logs through the console.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters**. The **Clusters** page appears.
3. Find the target cluster and click **View Logs** in the **Actions** column.



You can view operations performed on the cluster.



3.4.2.3. Connect to a cluster through kubectl

You can use the Kubernetes command line tool, **kubectl**, to connect to a Kubernetes cluster from a local computer.

Procedure

1. Download the latest kubectl client from the [Kubernetes change log page](#).

2. Install and set up the kubectl client.

For more information, see [Install and set up kubectl](#).

3. Configure the cluster credentials.

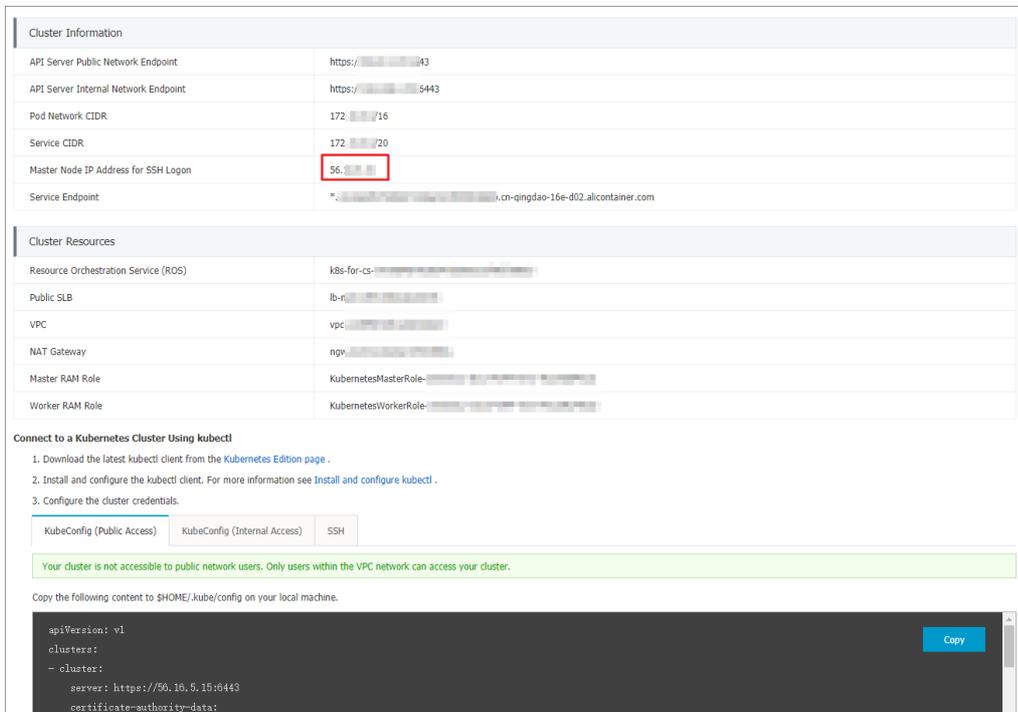
You can use the `scp` command to securely copy the master node configuration file from the `/etc/kubernetes/kube.conf` directory of the master VM and paste it to the `$HOME/.kube/config` directory of the local computer, where the `kubectl` credentials are expected to be stored.

```
mkdir $HOME/.kube
scp root@<master-public-ip>:/etc/kubernetes/kube.conf $HOME/.kube/config
```

You can find `master-public-ip` on the cluster details page.

- i. [Log on to the Container Service console](#).
- ii. In the left-side navigation pane, click **Clusters**. The Clusters page appears.
- iii. Find the target cluster and click **Manage** in the Actions column.

In the **Cluster Information** section, you can find the master node IP address.



3.4.2.4. Connect to a master node by using SSH

You can access a master node in a cluster by using a Secure Shell (SSH) client.

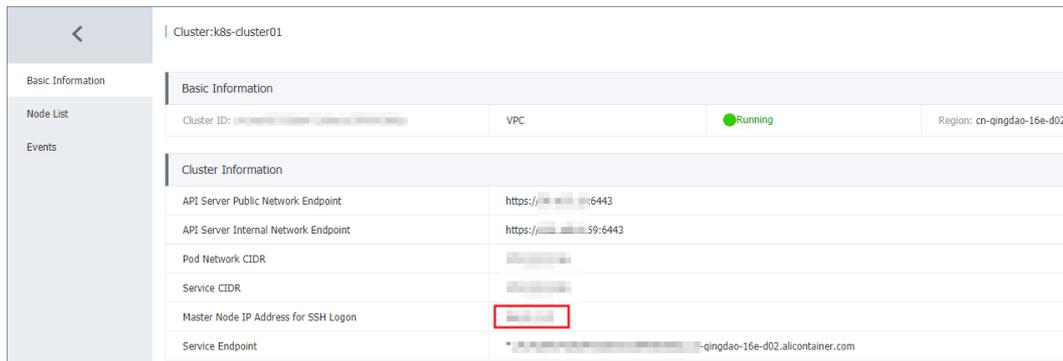
Prerequisites

- A Kubernetes cluster is created and **Use SSH to Access the Cluster from the Internet** is selected for the cluster. For more information, see [Create a Kubernetes cluster](#).
- The SSH client can connect to the network where the cluster is deployed.

Procedure

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Clusters > Clusters** to go to the Clusters page. Find the cluster that you want to manage, and click **Manage** in the Actions column for the cluster.

- The Basic Information page appears. In the Cluster Information section, you can find the IP address that is displayed in the Master Node IP Address for SSH Logon field.



- Use SSH to connect to the cluster from an SSH client that has access to the cluster network.
 - If you have a leased line that connects to the cluster network over the Internet, you can use tools such as PuTTY to create an SSH connection.
 - If you have an Elastic Compute Service (ECS) instance that is connected to the Virtual Private Cloud (VPC) network of the cluster, run the following command to create an SSH connection:

```
ssh root@ssh_ip #ssh_ip specifies the IP address of the master node for SSH connection.
```

3.4.2.5. Expand a cluster

You can adjust the number of worker nodes in a cluster based on business needs in the console.

Context

Currently, you cannot adjust the number of master nodes in a cluster.

Procedure

- Log on to the Container Service console.
- In the left-side navigation pane, choose Clusters > Clusters. The Clusters page appears.
- Select the target cluster and click Expand in the Actions column.
- On the Expand page, specify the number of worker nodes.
 - In this example, increase the number of worker nodes from zero to two.
- Configure worker nodes. Instance Type: You can select multiple instance types.
- Configure node logon settings.
 - Password: Set the logon password.
 - Confirm Password: Enter the logon password again.
- Click Submit.

What's next

In the left-side navigation pane, choose Clusters > Nodes. On the Nodes page, verify that the number of worker nodes is now changed to two.

3.4.2.6. Renew a certificate

This topic describes how to renew a Kubernetes certificate in the console.

Prerequisites

You have created a Kubernetes cluster and its certificate is about to expire.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Clusters. The Clusters page appears.
3. Select the target cluster and click Update Certificate. The Update Certificate page appears.

Note The Update Certificate button will be displayed two months before your cluster certificate expires.

4. Click Update and the Confirm page appears.
5. Click Confirm.

Result

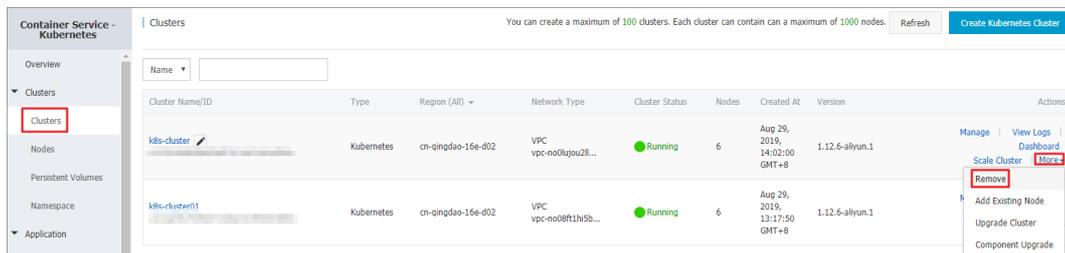
- On the Update Certificate page, the following message appears: The certificate has been updated.
- On the Clusters page, the Update Certificate button has disappeared.

3.4.2.7. Delete a cluster

You can delete clusters in the Container Service console.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Clusters. The Clusters page appears.
3. Find the target cluster and choose More > Delete in the Actions column.



What's next

Resource Orchestration Service (ROS) has no permission to delete resources that were manually added under ROS-created resources. For example, if you manually add a VSwitch under a ROS-created VPC instance, ROS cannot delete the VPC instance and therefore the cluster cannot be deleted.

Container Service allows you to force delete clusters. If your first attempt to delete a cluster fails, you can forcibly delete the cluster and ROS stack. However, you still need to manually release the resources that were manually added in the first place.

An error message appears when an attempt to delete a cluster fails.

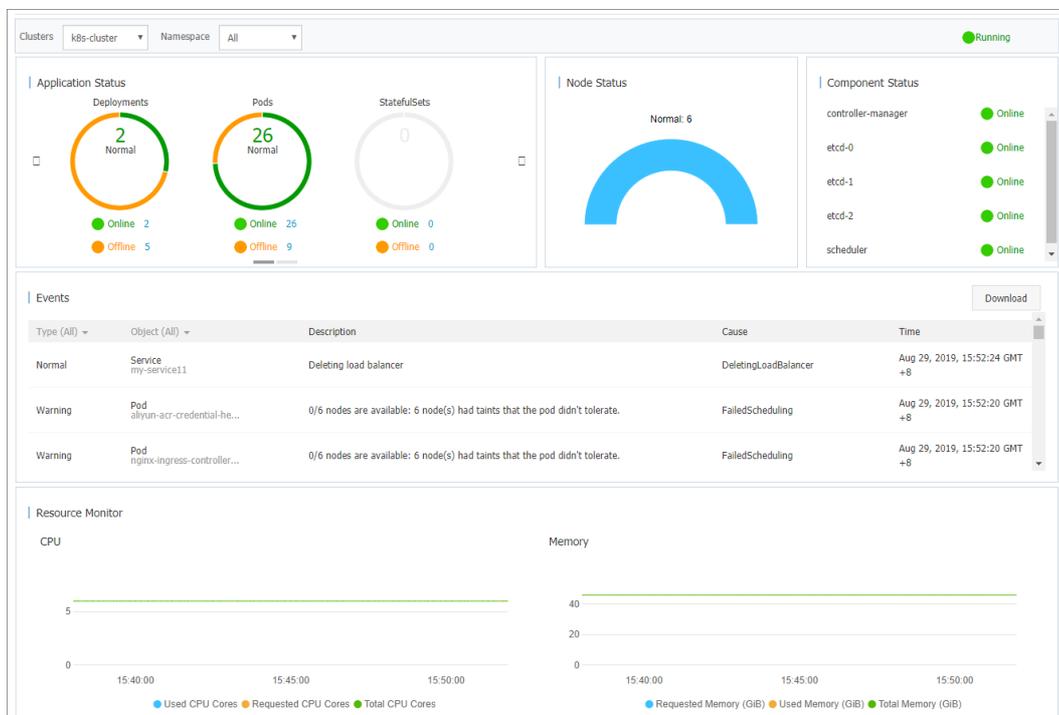
Select the cluster that you failed to delete and choose More > Delete in the Actions column. In the dialog box that appears, you can view the resources that were manually added. Select the Force Delete check box and click OK to delete the cluster and ROS resource stack.

3.4.2.8. View cluster overview

The Container Service for Kubernetes console provides a cluster overview page. This page displays the information such as application status, component status, and resource monitoring status. This allows you to check the health status of your cluster at your convenience.

Procedure

1. [Log on to the Container Service for Kubernetes console.](#)
2. In the left-side navigation pane, click **Overview**. The Overview page appears.
3. Select the target cluster and namespace. You can view the application status, component status, and resource monitoring charts.
 - **Application Status:** displays the statuses of the deployments, pods, and replica sets that are running in the cluster. Green sections indicate a normal state and yellow sections indicate an exception state.
 - **Node Status:** displays the statuses of the nodes in the cluster.
 - **Component Status:** Components are deployed in the kube-system namespace. Core components are used, such as the scheduler, controller-manager, and etcd.
 - **Events:** displays events such as warnings and errors. If no events are displayed, the cluster is running in the normal state.
 - **Monitoring:** displays CPU and memory monitoring charts. CPU usage is measured in cores or millicores and accurate to three decimal places. A millicore is one thousandth of a core. Memory usage is measured in GiB and accurate to three decimal places. For more information, see [Meaning of CPU](#) and [Meaning of memory](#).



3.4.3. Nodes

3.4.3.1. Add an existing node

You can add an existing ECS instance to a cluster. Currently, you can only add worker nodes to clusters.

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Log on to the Container Service console.](#)
- You have created an ECS instance. Ensure that the region, zone, department, project, security group, VPC network, and operating system settings of the ECS instance are the same as those of the cluster.

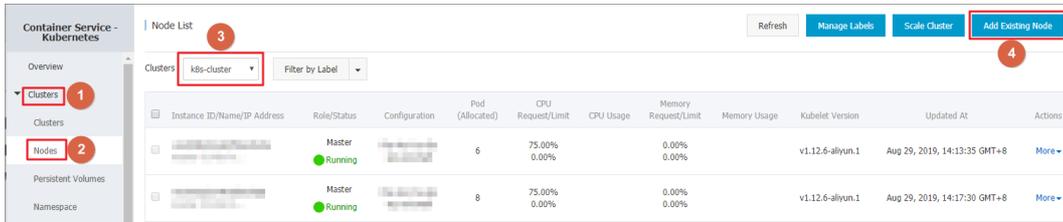
Context

- By default, a cluster can contain up to 50 nodes. To add more nodes to a cluster, submit a ticket.
- The ECS instance must be in the same region and VPC network as the cluster.

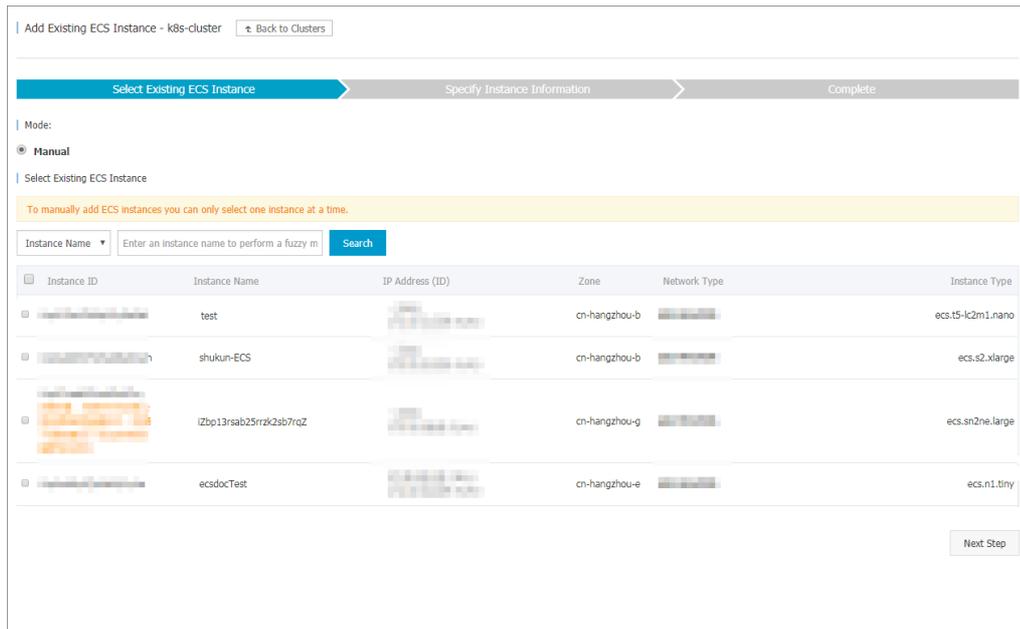
- The ECS instance must belong to the same account as the cluster.
- The ECS instance must be running the CentOS operating system.

Procedure

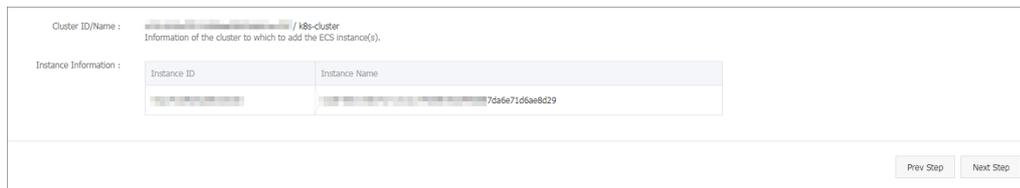
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Nodes to go to the Nodes page.
3. Select the target cluster and click Add Existing Node in the upper-right corner.



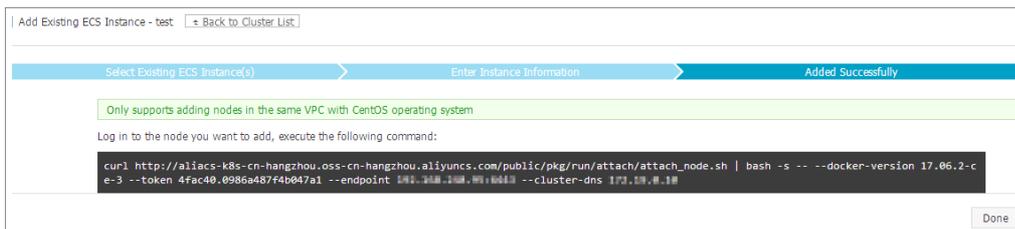
4. On the page that appears, you can manually add existing ECS instances to the cluster. To manually add an ECS instance, you need to obtain the installation command and log on to the ECS instance to run the command. You can only add one ECS instance at a time.
 - i. Select the ECS instance that you want to add and click Next Step. You can add only one ECS instance at a time.



- ii. Confirm the instance information and click Next Step.

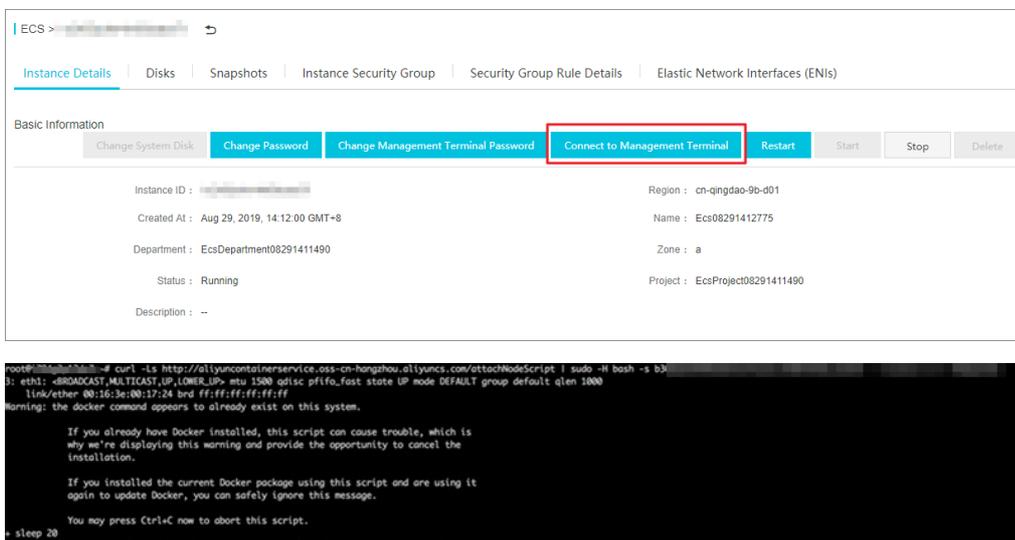


iii. Copy the command.



iv. Click Complete.

- v. Go to the ASCM console. In the top navigation bar, choose **Products > Elastic Compute Service**. On the **Instances** page, select the department and region of the cluster, and then find the target ECS instance.
- vi. Click the instance name to go to the Instance Details tab. Click **Connect to VNC**. In the dialog box that appears, enter the VNC password and then click **OK**. After you log on to the instance, paste the copied command and click **OK** to execute the script.



- vii. When the script execution succeeds, the ECS instance is added to the cluster. You can go to the Clusters page and click the cluster ID to view nodes in the cluster. Check whether the ECS instance has been added to the cluster.

3.4.3.2. View nodes

You can view the nodes in a cluster through commands, the console, or the Kubernetes Dashboard.

Through commands

Note To view the nodes in a cluster through commands, you need to **Connect to a Kubernetes cluster through kubectl**.

Use kubectl to connect to a cluster and run the following command to view the nodes in the cluster.

```
kubectl get nodes
```

A sample output is as follows:

```
$ kubectl get nodes
```

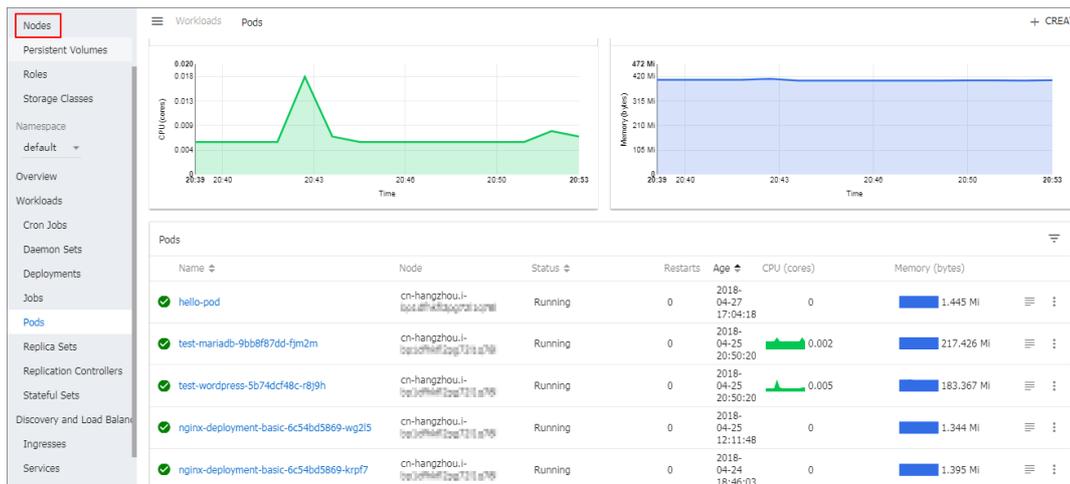
NAME	STATUS	AGE	VERSION
iz2ze2n6ep53tch701yh9zz	Ready	19m	v1.6.1-2+ed9e3d33a07093
iz2zeaf762wibijx39e5az	Ready	7m	v1.6.1-2+ed9e3d33a07093
iz2zeaf762wibijx39e5bz	Ready	7m	v1.6.1-2+ed9e3d33a07093
iz2zef4dnn9nos8elyr32kz	Ready	14m	v1.6.1-2+ed9e3d33a07093
iz2zeitvvo8enoreufstkmz	Ready	11m	v1.6.1-2+ed9e3d33a07093

Through the Container Service console

1. [Log on to the Container Service console.](#)
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the target cluster to view the nodes in the cluster.

Through the Kubernetes Dashboard

1. [Log on to the Container Service console.](#)
2. In the left-side navigation pane, click **Clusters** to go to the Clusters page.
3. Select the target cluster and click **Dashboard** in the Actions column to go to the Kubernetes Dashboard.
4. In the left-side navigation pane, click **Nodes**. On the page that appears, you can view all nodes in the cluster.



3.4.3.3. Manage node labels

You can manage node labels through the console. You can add a label to multiple nodes at the same time, filter nodes by label, and remove labels.

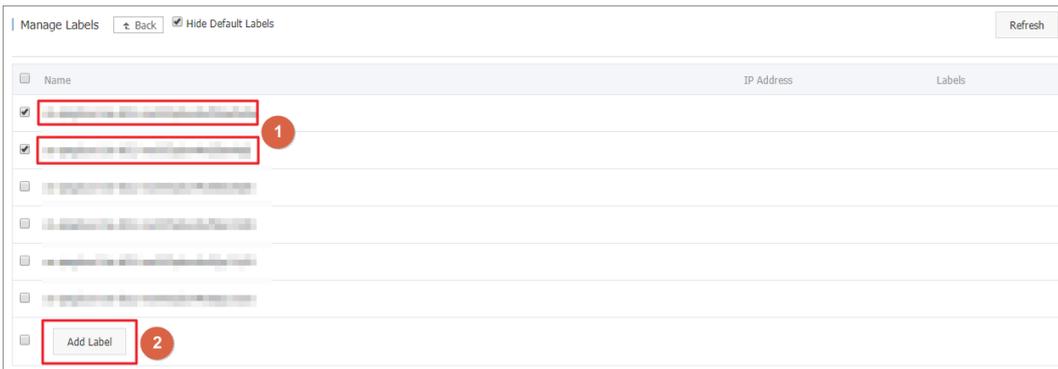
For more information about how to use labels to schedule nodes, see [Set node scheduling.](#)

Prerequisites

You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster.](#)

Add a label to multiple nodes

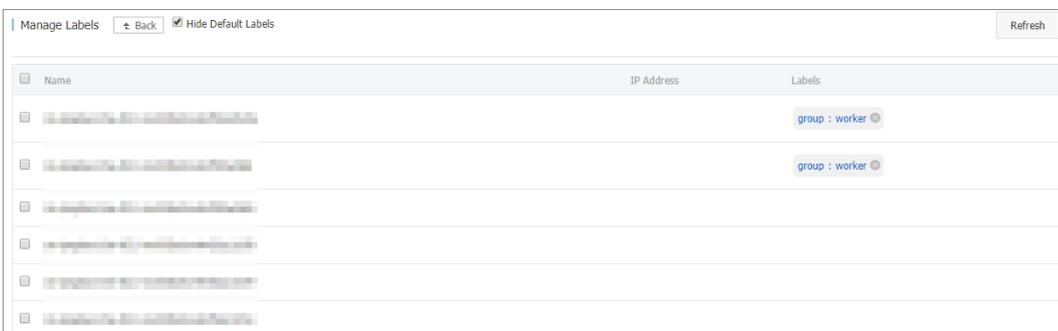
1. [Log on to the Container Service console.](#)
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the Nodes page.
3. Select the target cluster and click **Manage Labels** in the upper-right corner.
4. Select multiple nodes and then click **Add Label**.



5. In the dialog box that appears, enter the label name and value, and then click **OK**.

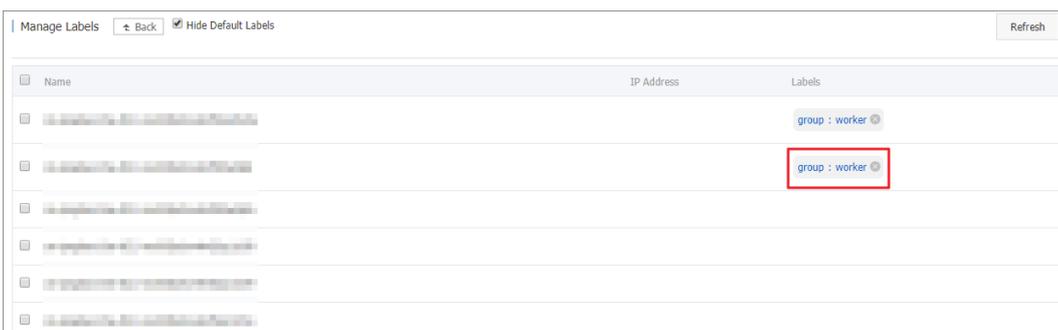


On the Manage Labels page, the selected nodes now have the same label.



Remove a label

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Nodes to go to the Nodes page.
3. Select the target cluster and click Manage Labels in the upper-right corner.
4. Select a node and click the cross sign at the end of a label, for example, `group:worker` .

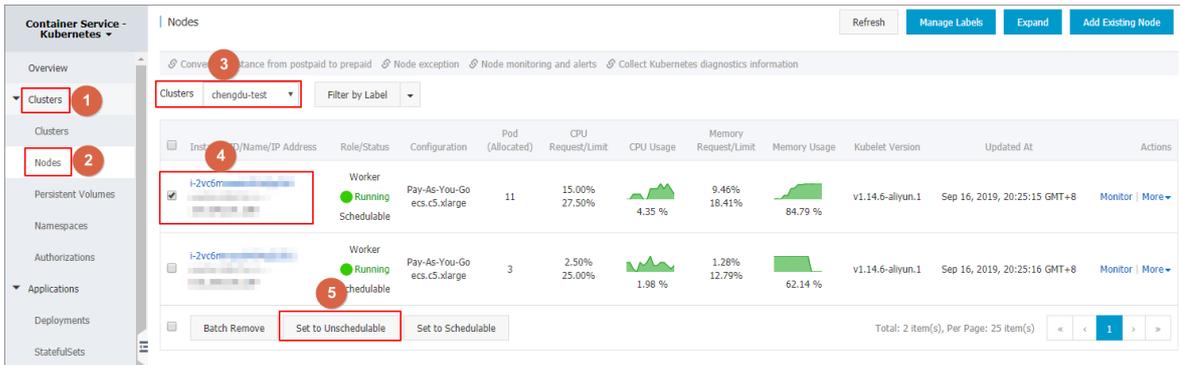


Click **Confirm** to remove the label.

3.4.3.4. Set node schedulability

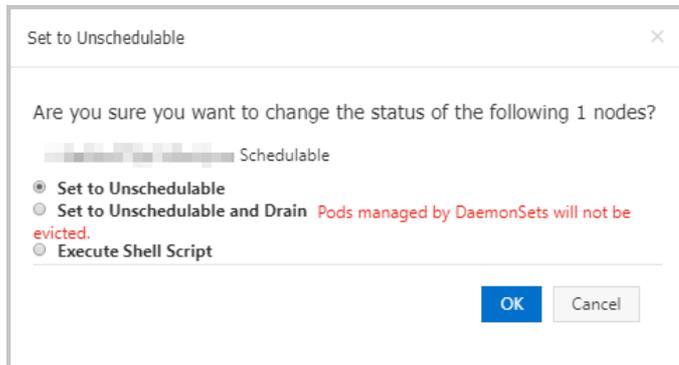
Procedure

1. Log on to the [Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the **Nodes** page.
3. Select the target cluster and select one or more nodes. Click **Set to Unscheduleable**.



4. In the dialog box that appears, you can set the specified nodes to either of the following statuses:
 - **Set to Unscheduleable:** Pods will not be scheduled to this node when you deploy new applications.
 - **Set to Unscheduleable and Drain:** Pods will not be scheduled to this node when you deploy new applications. Pods on this node will be evicted, except for the pods that are managed by DaemonSets.

In this example, **Set to Unscheduleable** is selected.



5. Click **OK**. The node status is now changed to **Unscheduleable**.



What's next

Pods will not be scheduled to the node when you deploy new applications.

3.4.3.5. Remove a node

To restart or release an ECS node in a cluster, you must remove the node from the cluster first. This topic describes how to remove a node.

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

- You can use `kubectl` to connect to the Kubernetes cluster. For more information, see [Connect to a Kubernetes cluster through kubectl](#).

Context

- When you remove a node, pods running on the node will be migrated to other nodes, which may cause service interruptions. We recommend that you remove nodes during off-peak hours.
- Unexpected errors may occur when you remove a node. We recommend that you back up your data in advance.
- The node that you choose to remove will be set to the unschedulable state.
- You can only remove worker nodes.

Procedure

1. Run the following command to migrate the pods on the target node to other nodes.

 **Note** Make sure that the other nodes have sufficient resources.

```
kubectl drain node-name
```

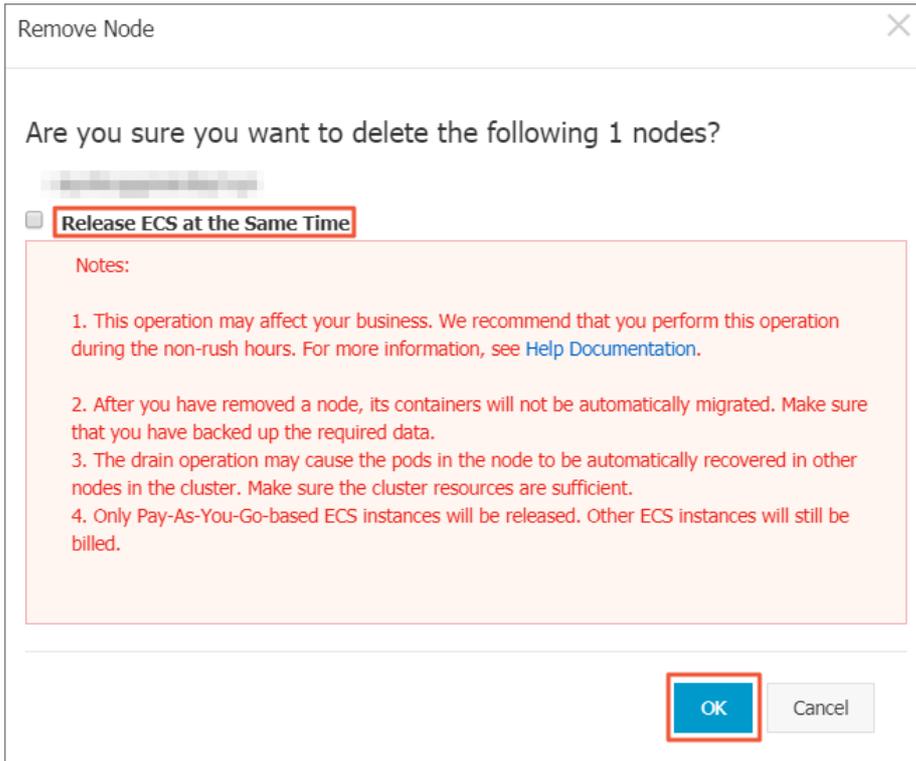
 **Note** *node-name* must be in the format of *your-region-name.node-id*.

- *your-region-name* represents the region where your cluster is deployed.
- *node-id* represents the ID of the ECS instance where the target node is deployed. For example, *cn-hangzhou.i-xxx*.

2. In the left-side navigation pane, choose **Clusters > Nodes** to go to the **Nodes** page.
3. Select the target cluster. Find the target node and choose **More > Remove** in the Actions column. The **Remove Node** page appears.

 **Note** To remove multiple nodes at the same time, select nodes on the **Nodes** page and click **Batch Remove**.

4. (Optional) To release the ECS instance where the target node is deployed, select the **Release ECS Instance** check box.



Note

- This option only releases pay-as-you-go ECS instances.
- Subscription ECS instances will be automatically released after the subscription expires.
- If you do not select the Release ECS Instance check box, you will continue to be billed for the ECS instance where the target node is deployed.

5. Click OK to remove the node.

3.4.3.6. View node resource usage

You can view the resource usage of the nodes in a cluster through the console.

Prerequisites

You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

Procedure

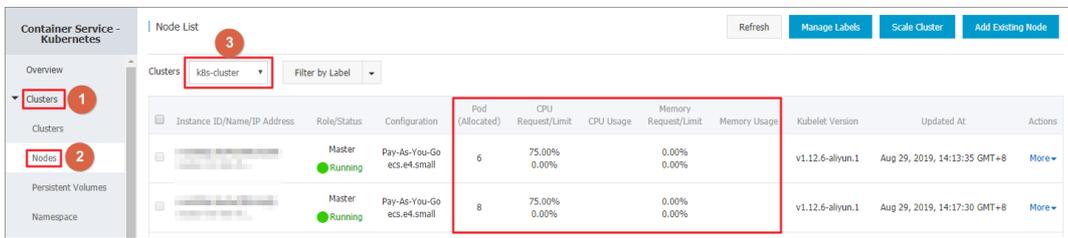
1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose Clusters > Nodes to go to the Nodes page.

You can view the request and usage rate of CPU and memory on each node.

- CPU request rate = sum (The amount of CPU requested by all pods on the node) / Total CPU of the node
- CPU usage rate = sum (The amount of CPU used by all pods on the node) / Total CPU of the node
- Memory request rate = (The amount of memory requested by all pods on the node) / Total memory of the node
- Memory usage rate = sum (The amount of memory used by all pods on the node) / Total memory of the node

Note

- You can adjust the workload of a node based on resource usage. For more information, see [Set node scheduling](#).
- When the request or usage rate of a node reaches 100%, pods will not be scheduled to the node.



3.4.3.7. Upgrade the NVIDIA driver on a GPU node

This topic describes how to upgrade the NVIDIA driver on a GPU node when workloads are deployed on the node and when no workload is deployed on the node.

Upgrade the NVIDIA driver on a GPU node where workloads are deployed

- Connect to a Kubernetes cluster through `kubectl`
- Run the following command to set the target node to unschedulable.

```
kubectl cordon node-name
```

Note

- Currently, you can only upgrade the NVIDIA driver on worker nodes.
- `node-name` must be in the format of `your-region-name.node-id`.
 - `your-region-name` represents the region where your cluster is deployed.
 - `node-id` represents the ID of the ECS instance where the target node is deployed.

You can run the following command to query `node-name`.

```
kubectl get node
```

```
[root@gpu-test ~]# kubectl cordon cn-hangzhou.i-... node/cn-hangzhou.i-... already cordoned
```

- Run the following command to migrate pods from the target node to other nodes:

```
kubectl drain node-name --grace-period=120 --ignore-daemonsets=true
```

```
[root@gpu-test ~]# kubectl drain cn-hangzhou.i-... --grace-period=120 --ignore-daemonsets=true
node/cn-hangzhou.i-... cordoned
WARNING: Ignoring DaemonSet-managed pods: flexvolume-..., kube-flannel-ds-..., kube-proxy-worker-..., logtail-ds-...
pod/domain-nginx-... evicted
pod/old-nginx-... evicted
pod/new-nginx-... evicted
pod/old-nginx-... evicted
```

- Run the following command to log on to the target node:

```
ssh root@xxx.xxx.x.xx
```

- Run the following command to check the current NVIDIA driver version:

```
nvidia-smi
```

```
[root@ ~]# nvidia-smi
Fri Jan 18 16:44:52 2019

+-----+
| NVIDIA-SMI 384.111                Driver Version: 384.111 |
+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp            Perf          Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|  0   Tesla P4             0n          6W / 75W | 0MiB / 7606MiB |      0%      Default |
+-----+-----+-----+-----+-----+-----+

Processes:
GPU      PID    Type   Process name                      GPU Memory
Usage
+-----+-----+-----+-----+-----+-----+
No running processes found
+-----+-----+-----+-----+-----+-----+

```

6. Run the following commands to uninstall the existing driver:

Note

- If your driver version is *384.111*, perform the following steps.
- If your driver version is not *384.111*, download the corresponding driver from the official NVIDIA website first.

```
cd /tmp

curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run

chmod u+x NVIDIA-Linux-x86_64-384.111.run

./NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
```

7. Run the following command to restart the target node:

```
reboot
```

8. Download the driver that you want to use from the official NVIDIA website. In this example, version *410.79* is used.

9. Run the following command to install the downloaded driver under the directory where it was saved:

```
sh ./NVIDIA-Linux-x86_64-410.79.run -a -s -q
```

10. Run the following commands to configure the driver:

```
nvidia-smi -pm 1 || true

nvidia-smi -acp 0 || true
```

11. Run the following commands to update device-plugin:

```
mv /etc/kubernetes/manifests/nvidia-device-plugin.yml /

mv /nvidia-device-plugin.yml /etc/kubernetes/manifests/
```

12. Log on to a master node and run the following command to set the target node to schedulable:

```
kubectl uncordon node-name
```

Result

Run the following command on a master node to check the NVIDIA driver version on the target node. The driver version is now *410.79*.

Note Replace *node-name* with the target node name.

```
kubectl exec -n kube-system -t nvidia-device-plugin-node-name nvidia-smi
```

```
[root@gpu-test ~]# kubectl exec -n kube-system -t nvidia-device-plugin-cn- nvidia-smi
Mon Jan 21 03:14:48 2019
+-----+
| NVIDIA-SMI 410.79      | Driver Version: 410.79      | CUDA Version: N/A      |
+-----+-----+-----+
| GPU  Name            | Persistence-M| Bus-Id        | Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf     | Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+
|   0   Tesla P4             |      0n      | 00000000:00:08:0 | Off    |          0          |
| N/A   21C    P8             |   6W / 75W   |     0MiB / 7611MiB |    0%    | Default           |
+-----+-----+-----+-----+-----+
+-----+
| Processes:                                | GPU Memory Usage |
| GPU      PID  Type  Process name                               | Usage            |
+-----+-----+-----+-----+-----+
| No running processes found                |                  |
+-----+-----+-----+-----+-----+
```

Upgrade the NVIDIA driver on a GPU node where no workload is deployed

1. Run the following command to log on to the target node:

```
ssh root@xxx.xxx.x.xx
```

2. Run the following command to check the current NVIDIA driver version:

```
nvidia-smi
```

```
[root@ ~]# nvidia-smi
Fri Jan 18 16:44:52 2019
+-----+
| NVIDIA-SMI 384.111    | Driver Version: 384.111    |
+-----+-----+-----+
| GPU  Name            | Persistence-M| Bus-Id        | Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf     | Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+
|   0   Tesla P4             |      0n      | 00000000:00:08:0 | Off    |          0          |
| N/A   24C    P8             |   6W / 75W   |     0MiB / 7606MiB |    0%    | Default           |
+-----+-----+-----+-----+-----+
+-----+
| Processes:                                | GPU Memory Usage |
| GPU      PID  Type  Process name                               | Usage            |
+-----+-----+-----+-----+-----+
| No running processes found                |                  |
+-----+-----+-----+-----+-----+
```

3. Run the following commands to uninstall the existing driver:

Note

- If your driver version is *384.111*, perform the following steps.
- If your driver version is not *384.111*, download the corresponding driver from the official NVIDIA website first.

```
cd /tmp
```

```
curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run
```

```
chmod u+x NVIDIA-Linux-x86_64-384.111.run
```

```
./NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
```

4. Run the following command to restart the target node.

```
reboot
```

5. Download the driver that you want to use from the official NVIDIA website. In this example, version *410.79* is used.

6. Run the following command to install the downloaded driver under the directory where it was saved:

```
sh ./NVIDIA-Linux-x86_64-410.79.run -a -s -q
```

7. Run the following commands to configure the driver:

```
nvidia-smi -pm 1 || true
```

```
nvidia-smi -acp 0 || true
```

Result

Run the following command on a master node to check the NVIDIA driver version on the target node. The driver version is now *410.79*.

Note Replace *node-name* with the target node name.

```
kubectl exec -n kube-system -t nvidia-device-plugin-node-name nvidia-smi
```

```
[root@gpu-test ~]# kubectl exec -n kube-system -t nvidia-device-plugin-cn-... nvidia-smi
Mon Jan 21 03:14:48 2019
+-----+
| NVIDIA-SMI 410.79      | Driver Version: 410.79      | CUDA Version: N/A      |
+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC  |
| Fan  Temp   Perf     Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M.  |
+-----+-----+
|  0   Tesla P4             0n          | 00000000:00:08.0 Off  |          0%          |
| N/A   21C    P8             6W / 75W    |  0MiB / 7611MiB |          0%          |
+-----+-----+
+-----+-----+
| Processes:                               GPU Memory Usage |
|  GPU       PID  Type  Process name                               | Mem Used |
+-----+-----+
| No running processes found               |          |
+-----+-----+
```

3.4.3.8. Create a Kubernetes cluster for GPU computing

This topic describes how to create a Kubernetes cluster for GPU computing.

Prerequisites

The Container Service for Kubernetes, Resource Orchestration Service (ROS), and Resource Access Management (RAM) services are activated.

Note Container Service for Kubernetes uses ROS to deploy applications in Kubernetes clusters. Before you create a Kubernetes cluster, you must activate ROS.

Context

Starting from version 1.8, Kubernetes adds support for the following hardware acceleration devices by using **device plug-ins**: NVIDIA GPUs, InfiniBand devices, and field-programmable gate arrays (FPGAs). GPU solutions developed by the community will be phased out in version 1.10, and removed from the master code in version 1.11. Container Service for Kubernetes enables you to use a GPU-accelerated Kubernetes cluster to run compute-intensive tasks such as machine learning and image processing. You can deploy applications and achieve auto scaling without the need to install NVIDIA drivers or Compute Unified Device Architecture (CUDA) in advance.

You must complete the following operations in the Container Service for Kubernetes console to create a Kubernetes cluster:

- Create Elastic Compute Service (ECS) instances, configure a public key to enable SSH logon from master nodes to other nodes, and configure the Kubernetes cluster by using cloud-init.
- Create a security group that allows access to the Virtual Private Cloud (VPC) network over Internet Control Message Protocol (ICMP).
- Create a VPC network and a VSwitch and create SNAT rules for the VSwitch if you do not specify an existing VPC network.
- Create VPC routing rules.
- Create a NAT gateway and an elastic IP address.
- Create a RAM user and grant it permissions to query, create, and delete ECS instances and permissions to add and delete cloud disks. The RAM user is also granted all permissions on Server Load Balancer (SLB), Cloud Monitor, VPC, Log Service, and Network Attached Storage (NAS). The Kubernetes cluster dynamically creates SLB instances, cloud disks, and VPC routing rules based on your settings.
- Create an internal SLB instance and open port 6443.
- Create a public SLB instance and open ports 6443, 8443, and 22. If you choose to enable SSH logon when you create the cluster, port 22 is enabled. Otherwise, port 22 is not enabled.

Limits

- Kubernetes clusters support only VPC networks.
- By default, each account has specific quotas on the amount of cloud resources that can be created. You cannot create clusters if the quota limit is exceeded. Make sure that you have sufficient quotas before you create a cluster. To request a quota increase, submit a ticket.
 - You can create up to five clusters across all regions for an account. A cluster can contain up to 40 nodes. To create more clusters or nodes, submit a ticket.

Note In a Kubernetes cluster, you can create up to 48 route entries per VPC. This means that a cluster can contain up to 48 nodes. To increase the number of nodes, submit a ticket to increase the number of route entries first.

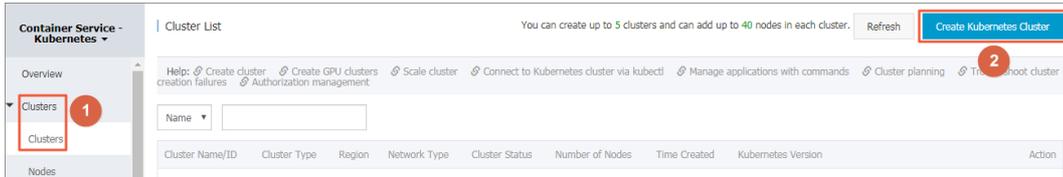
- You can create up to 100 security groups for each account.
- You can create up to 60 pay-as-you-go SLB instances for each account.
- You can create up to 20 elastic IP addresses for each account.
- ECS instances have the following limit:

Only CentOS is supported.

Create a GN5 Kubernetes cluster

GN5 Kubernetes clusters support only Kubernetes 1.12.6-aliyun.1. Kubernetes 1.11.5 is not supported.

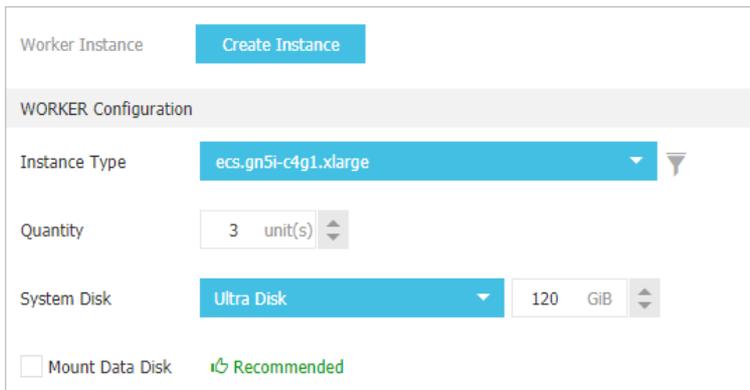
1. Log on to the [Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Clusters > Clusters**. The **Clusters** page appears.
3. In the upper-right corner of the page, click **Create Kubernetes Cluster**. The **Create Kubernetes Cluster** page appears.



You are redirected to the **Dedicated Kubernetes** tab by default.

Note To create a cluster for GPU computing, select ECS instance types with GPU capabilities to create worker nodes. For more information about other parameters, see [Cluster parameters](#).

4. Configure worker nodes. In this example, worker nodes are used to run GPU tasks and the gn5i-c4g1 instance type is selected.
 - i. If you choose to create worker instances, you must set Instance Type and Quantity. Three worker nodes with GPU capabilities are created in this example.



Note We recommend that you use SSD disks.

- ii. If you choose to add existing instances, you must create GPU-accelerated instances in the target region in advance.
5. Set the other parameters and click **Create Cluster** to start the deployment. After the cluster is created, choose **Clusters > Nodes** to go to the **Nodes** page.

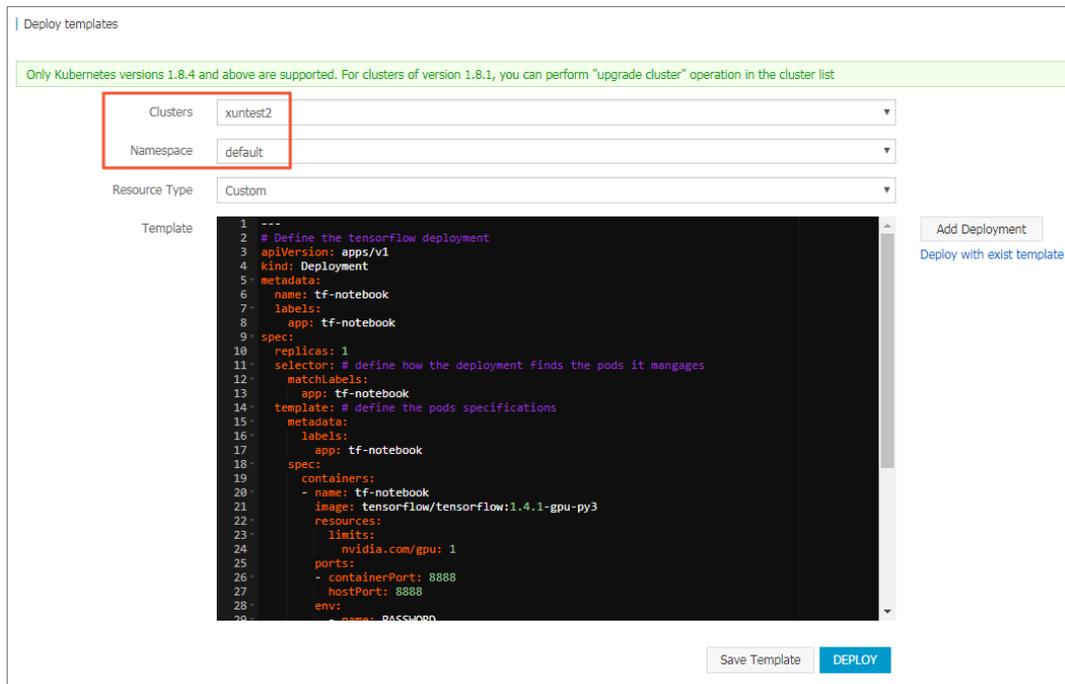
Select the target cluster. Find one of the created nodes and choose **More > Details** to view the GPU-based devices attached to the node.

Create a GPU experimental environment to run TensorFlow

Jupyter is a standard tool that is used by data scientists to create the experimental environment to run TensorFlow. The following example shows how to deploy a Jupyter application.

1. In the left-side navigation pane, choose **Applications > Deployments** to go to the **Deployments** page.
2. In the upper-right corner of the page, click **Create from Template**.

3. Select the target cluster and namespace. Select a sample template, or set Sample Template to Custom and customize the template in the Template field. Then, you can click Create.



In this example, a Jupyter application template is implemented. The template includes a deployment and a service.

```

---
# Define the tensorflow deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tf-notebook
  labels:
    app: tf-notebook
spec:
  replicas: 1
  selector: # define how the deployment finds the pods it manages
  matchLabels:
    app: tf-notebook
  template: # define the pods specifications
  metadata:
    labels:
      app: tf-notebook
  spec:
    containers:
      - name: tf-notebook
        image: tensorflow/tensorflow:1.4.1-gpu-py3
        resources:
          limits:
            nvidia.com/gpu: 1          #Specifies the number of NVIDIA GPUs that are called by the application.
    
```

```

ports:
  - containerPort: 8888
    hostPort: 8888
env:
  - name: PASSWORD          #Specifies the password used to access the Jupyter instance. You can mod
    ify the password as required.
    value: mypassw0rd

# Define the tensorflow service
---
apiVersion: v1
kind: Service
metadata:
  name: tf-notebook
spec:
  ports:
    - port: 80
      targetPort: 8888
      name: jupyter
  selector:
    app: tf-notebook
  type: LoadBalancer      #Creates an SLB service to ensure that the Jupyter instance is accessible over
                           the Internet.

```

4. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. Select the target cluster and namespace. Find the tf-notebook service and check its external endpoint.

Name	Label	Type	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint	Action
kubernetes	component:apiserver provider:kubernetes	ClusterIP	05/17/2019,18:12:33		kubernetes:443 TCP	-	Details Update View YAML Delete
tf-notebook	-	LoadBalancer	05/23/2019,10:46:02		tf-notebook:80 TCP tf-notebook:30708 TCP		Details Update View YAML Delete

5. To connect to the Jupyter instance in a browser, enter `http://EXTERNAL-IP` in the address bar and enter the password specified in the template.
6. You can run the following program to verify that the Jupyter instance has access to GPU-based devices. The program lists all devices that can be used by TensorFlow:

```

from tensorflow.python.client import device_lib

def get_available_devices():
    local_device_protos = device_lib.list_local_devices()
    return [x.name for x in local_device_protos]

print(get_available_devices())

```

```

In [2]: from tensorflow.python.client import device_lib

def get_available_devices():
    local_device_protos = device_lib.list_local_devices()
    return [x.name for x in local_device_protos]

print(get_available_devices())

['/device:CPU:0', '/device:GPU:0']
    
```

3.4.3.9. Use labels to schedule pods to GPU nodes

To use Kubernetes clusters for GPU computing, you need to schedule pods to GPU nodes. To make the scheduling more flexible and efficient, you can add labels to GPU nodes.

Context

When Kubernetes deploys nodes with NVIDIA GPUs, the attributes of these GPUs will be discovered and exposed as node labels, which have the following benefits:

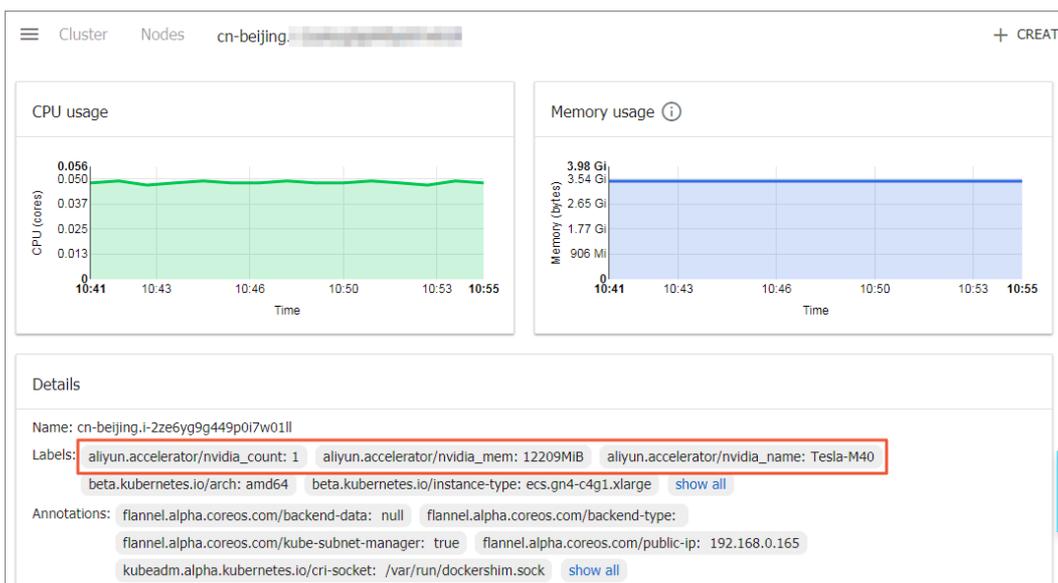
- You can quickly filter GPU nodes by label.
- You can use labels as scheduling conditions when you deploy pods.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Nodes. The Nodes page appears.

Note This example selects a cluster with three worker nodes, among which two are equipped with GPUs. Note the node IP addresses.

3. Select a GPU node and choose More > Details in the Actions column to go to Kubernetes Dashboard. You can view the labels attached to the node.



You can also log on to a master node and run the following command to view the labels on GPU nodes:

```
# kubectl get nodes
NAME                STATUS  ROLES  AGE  VERSION
cn-beijing.i-2ze2dy2h9w97v65u**** Ready  master  2d   v1.12.6-aliyun.1
cn-beijing.i-2ze8o1a45qdv5q8a**** Ready  <none>  2d   v1.12.6-aliyun.1 # Compare these nodes with t
he nodes displayed in the console to identify GPU nodes.
cn-beijing.i-2ze8o1a45qdv5q8a**** Ready  <none>  2d   v1.12.6-aliyun.1
cn-beijing.i-2ze9xylyn11vop7g**** Ready  master  2d   v1.12.6-aliyun.1
cn-beijing.i-2zed5sw8snjniq6m**** Ready  master  2d   v1.12.6-aliyun.1
cn-beijing.i-2zej9s0zjykp9pw**** Ready  <none>  2d   v1.12.6-aliyun.1
```

Select a GPU node and run the following command to query its labels:

```
# kubectl describe node cn-beijing.i-2ze8o1a45qdv5q8a****
Name:          cn-beijing.i-2ze8o1a45qdv5q8a7luz
Roles:        <none>
Labels:       aliyun.accelerator/nvidia_count=1 # This field is important.
              aliyun.accelerator/nvidia_mem=12209MiB
              aliyun.accelerator/nvidia_name=Tesla-M40
              beta.kubernetes.io/arch=amd64
              beta.kubernetes.io/instance-type=ecs.gn4-c4g1.xlarge
              beta.kubernetes.io/os=linux
              failure-domain.beta.kubernetes.io/region=cn-beijing
              failure-domain.beta.kubernetes.io/zone=cn-beijing-a
              kubernetes.io/hostname=cn-beijing.i-2ze8o1a45qdv5q8a****
.....
```

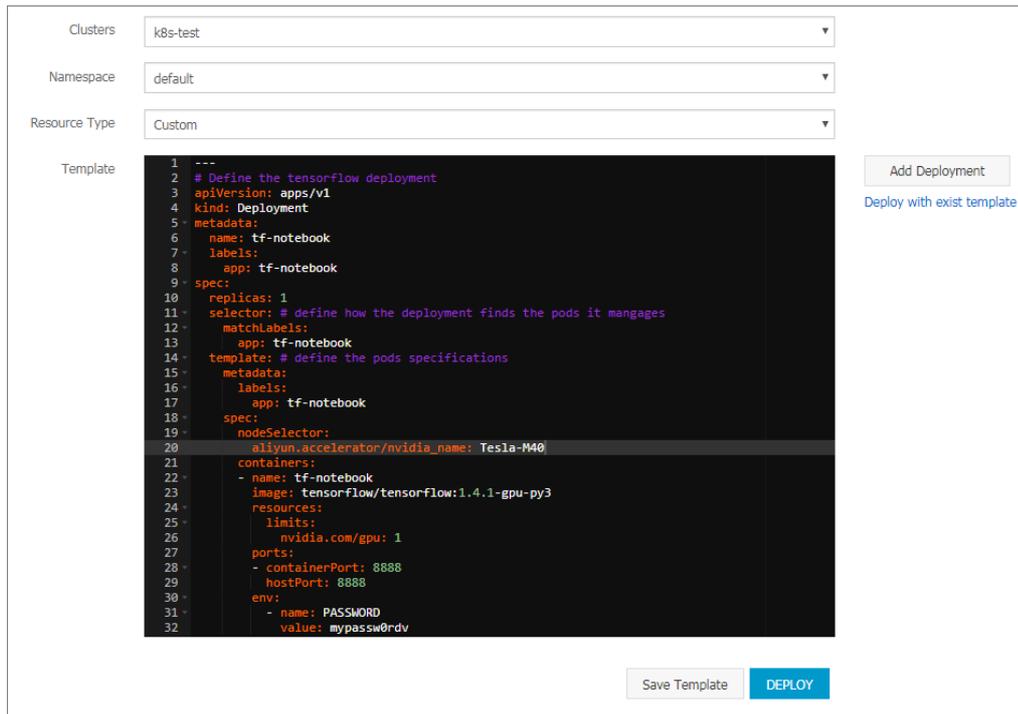
In this example, the GPU node is attached with the following three labels:

key	value
aliyun.accelerator/nvidia_count	The number of GPU cores.
aliyun.accelerator/nvidia_mem	The size of the GPU memory in MiB.
aliyun.accelerator/nvidia_name	The name of the NVIDIA graphics card.

GPU nodes of the same type have the same graphics card name. You can use this label to filter nodes.

```
# kubectl get no -l aliyun.accelerator/nvidia_name=Tesla-M40
NAME                STATUS  ROLES  AGE  VERSION
cn-beijing.i-2ze8o1a45qdv5q8a**** Ready  <none>  2d   v1.12.6-aliyun.1
cn-beijing.i-2ze8o1a45qdv5q8a**** Ready  <none>  2d   v1.12.6-aliyun.1
```

4. Go to the homepage of the Container Service console. In the left-side navigation pane, choose **Applications > Deployments**. On the page that appears, click **Create from Template** in the upper-right corner.
 - i. Create a TensorFlow deployment and schedule it to a GPU node.



The screenshot shows a web interface for configuring a Kubernetes deployment. At the top, there are three dropdown menus: 'Clusters' set to 'k8s-test', 'Namespace' set to 'default', and 'Resource Type' set to 'Custom'. Below these is a 'Template' section with a code editor containing a YAML template. To the right of the code editor are two buttons: 'Add Deployment' and 'Deploy with exist template'. At the bottom right of the interface are two buttons: 'Save Template' and 'DEPLOY'.

```
1 ---
2 # Define the tensorflow deployment
3 apiVersion: apps/v1
4 kind: Deployment
5 metadata:
6   name: tf-notebook
7   labels:
8     app: tf-notebook
9 spec:
10  replicas: 1
11  selector: # define how the deployment finds the pods it mangages
12  matchLabels:
13    app: tf-notebook
14  template: # define the pods specifications
15    metadata:
16      labels:
17        app: tf-notebook
18    spec:
19      nodeSelector:
20        aliyun.accelerator/nvidia_name: Tesla-M40
21      containers:
22      - name: tf-notebook
23        image: tensorflow/tensorflow:1.4.1-gpu-py3
24        resources:
25          limits:
26            nvidia.com/gpu: 1
27        ports:
28          - containerPort: 8888
29            hostPort: 8888
30        env:
31          - name: PASSWORD
32            value: mypassw0rdv
```

This example uses the following YAML template:

```
---
# Define the tensorflow deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tf-notebook
  labels:
    app: tf-notebook
spec:
  replicas: 1
  selector: # define how the deployment finds the pods it mangages
    matchLabels:
      app: tf-notebook
  template: # define the pods specifications
    metadata:
      labels:
        app: tf-notebook
    spec:
      nodeSelector:
        aliyun.accelerator/nvidia_name: Tesla-M40          # This field is important.
      containers:
        - name: tf-notebook
          image: tensorflow/tensorflow:1.4.1-gpu-py3
          resources:
            limits:
              nvidia.com/gpu: 1                               # This field is important.
          ports:
            - containerPort: 8888
              hostPort: 8888
          env:
            - name: PASSWORD
              value: mypassw0rdv
```

- ii. You can also avoid deploying an application to a GPU node. The following example deploys an NGINX pod and schedules the pod based on node affinity. For more information, see the part about node affinity in [Create an application from an image](#).

This example uses the following YAML template:

```
apiVersion: v1
kind: Pod
metadata:
  name: not-in-gpu-node
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: aliyun.accelerator/nvidia_name
                operator: DoesNotExist
  containers:
    - name: not-in-gpu-node
      image: nginx
```

5. In the left-side navigation pane, choose **Applications > Pods**. On the page that appears, select the target cluster and namespace.

Result

On the Pods page, the two pods from preceding examples have been scheduled to the target nodes. You can use labels to schedule pods to specific GPU nodes with ease.

3.4.3.10. Manually upgrade the kernel of a GPU node in a cluster

This topic describes how to manually upgrade the kernel of a GPU node in a cluster.

Context

The current kernel version is earlier than `3.10.0-957.21.3`.

Procedure

1. [Connect to a Kubernetes cluster through kubectl](#).
2. Run the following command to set the target GPU node to unschedulable. This example uses node `cn-beijing.i-2ze19qyi8votgjz12345` as the target node.

```
kubectl cordon cn-beijing.i-2ze19qyi8votgjz12345

node/cn-beijing.i-2ze19qyi8votgjz12345 already cordoned
```

3. Run the following command to drain the target GPU node:

```
# kubectl drain cn-beijing.i-2ze19qyi8votgjz12345 --grace-period=120 --ignore-daemonsets=true

node/cn-beijing.i-2ze19qyi8votgjz12345 cordoned
WARNING: Ignoring DaemonSet-managed pods: flexvolume-9scb4, kube-flannel-ds-r2qmh, kube-proxy-worker-l62sf, logtail-ds-f9vbg
pod/nginx-ingress-controller-78d847fb96-5fkkw evicted
```

4. Uninstall the existing nvidia-driver.

 **Note** This step uninstalls the version 384.111 driver. If your driver version is not 384.111, you need to download a driver from the official NVIDIA website and replace 384.111 with your actual version number.

i. Log on to the target GPU node and run the `nvidia-smi` command to query the driver version.

```
# nvidia-smi -a | grep 'Driver Version'
Driver Version          : 384.111
```

ii. Run the following commands to download the driver installation package:

```
cd /tmp/
curl -O https://cn.download.nvidia.cn/tesla/384.111/NVIDIA-Linux-x86_64-384.111.run
```

 **Note** The installation package is required to uninstall the driver.

iii. Run the following commands to uninstall the existing nvidia-driver:

```
chmod u+x NVIDIA-Linux-x86_64-384.111.run
./NVIDIA-Linux-x86_64-384.111.run --uninstall -a -s -q
```

5. Run the following commands to upgrade kernel:

```
yum clean all && yum makecache
yum update kernel -y
```

6. Run the following command to restart the GPU node:

```
reboot
```

7. Log on to the GPU node and run the following command to install the kernel-devel package.

```
yum install -y kernel-devel-$(uname -r)
```

8. Run the following commands to download the required driver and install it on the target node. In this example, version 410.79 is used.

```
cd /tmp/  
curl -O https://cn.download.nvidia.cn/tesla/410.79/NVIDIA-Linux-x86_64-410.79.run  
chmod u+x NVIDIA-Linux-x86_64-410.79.run  
sh . /NVIDIA-Linux-x86_64-410.79.run -a -s -q  
  
# warm up GPU  
nvidia-smi -pm 1 || true  
nvidia-smi -acp 0 || true  
nvidia-smi --auto-boost-default=0 || true  
nvidia-smi --auto-boost-permission=0 || true  
nvidia-modprobe -u -c=0 -m || true
```

9. Check the `/etc/rc.d/rc.local` file and check whether the following configurations are included. If not, add the following content.

```
nvidia-smi -pm 1 || true  
nvidia-smi -acp 0 || true  
nvidia-smi --auto-boost-default=0 || true  
nvidia-smi --auto-boost-permission=0 || true  
nvidia-modprobe -u -c=0 -m || true
```

10. Run the following commands to restart kubelet and Docker.

```
service kubelet stop  
service docker restart  
service kubelet start
```

11. Run the following command to set the GPU node to schedulable:

```
# kubectl uncordon cn-beijing.i-2ze19qyi8votgjz12345  
  
node/cn-beijing.i-2ze19qyi8votgjz12345 already uncordoned
```

12. Run the following command on the `nvidia-device-plugin` container to check the driver version:

```
kubectl exec -n kube-system -t nvidia-device-plugin-cn-beijing.i-2ze19qyi8votgjz12345 nvidia-smi
Thu Jan 17 00:33:27 2019

+-----+
| NVIDIA-SMI 410.79   Driver Version: 410.79   CUDA Version: N/A   |
+-----+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+
| 0 Tesla P100-PCIE... On  | 00000000:00:09:0 Off |          0 |
| N/A   27C   P0   28W / 250W | 0MiB / 16280MiB | 0%      Default |
+-----+-----+-----+

+-----+
| Processes:                                     GPU Memory |
|  GPU   PID  Type  Process name                               Usage      |
|=====+=====+=====+
| No running processes found                       |
+-----+
```

3.4.4. Storage

3.4.4.1. Overview

In the Container Service console, you can create volumes of other Apsara Stack services, enabling you to create stateful applications and use Apsara Stack disks and OSS to implement persistent storage.

Both static and dynamic volumes are supported. The following table shows how static and dynamic volumes are supported.

Apsara Stack storage	Static volume	Dynamic volume
Apsara Stack disk	<p>You can use a static disk volume through either of the following methods:</p> <ul style="list-style-type: none"> • Use a volume directly • Use a volume through a PV and PVC 	Supported
Apsara Stack NAS	<p>You can use a static NAS volume through either of the following methods:</p> <ul style="list-style-type: none"> • Use a volume through the FlexVolume plug-in <ul style="list-style-type: none"> ◦ Use a volume directly ◦ Use a volume through a PV or PVC • Use a volume through the Kubernetes NFS driver 	Supported

Apsara Stack storage	Static volume	Dynamic volume
Apsara Stack OSS	<p>You can use a static OSS volume through either of the following methods:</p> <ul style="list-style-type: none"> • Use a volume directly • Use a volume through a PV or PVC 	Not supported

3.4.4.2. Use Apsara Stack disks

You can use Apsara Stack disks to create volumes.

You can use volumes created from Apsara Stack disks in Kubernetes clusters.

Apsara Stack disks can be attached to Kubernetes clusters as the following volume types:

- **Statically provisioned volumes**

You can use the statically provisioned volumes in either of the following ways:

- Use a disk as a volume.
- Create a persistent volume (PV) or a persistent volume claim (PVC).

- **Dynamically provisioned volumes**

Considerations

- A disk is a non-shared storage device that can only be attached to a single pod at a time.
- To use a disk as a volume, you must have created the disk and obtained its disk ID.

The disk must meet the following capacity requirements:

- A basic disk must have a minimum capacity of 5 GiB.
- An ultra disk must have a minimum capacity of 20 GiB.
- An SSD disk must have a minimum capacity of 20 GiB.

- `volumeId`: the ID of the attached disk. The value must be the same as `volumeName` and `PV Name`.
- The node and the disk to be attached must be in the same zone.
- Only pay-as-you-go disks can be attached to nodes. If you change the billing method of an Elastic Compute Service (ECS) instance in the cluster from pay-as-you-go to subscription, you cannot change the billing method of its disks to subscription. Otherwise, the disks cannot be attached to the cluster.

Statically provisioned volumes

You can use disks as volumes or by creating PVs and PVCs.

Prerequisites

A disk is created in the ECS console.

- **Use a disk as a volume**

Use the following `disk-deploy.yaml` file to create a pod.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-disk-deploy
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx-flexvolume-disk
          image: nginx
          volumeMounts:
            - name: "d-bp1j17ifxfasvts3tf40"
              mountPath: "/data"
      volumes:
        - name: "d-bp1j17ifxfasvts3tf40"
          flexVolume:
            driver: "alicloud/disk"
            fsType: "ext4"
            options:
              volumeId: "d-bp1j17ifxfasvts3tf40"
```

- **Use a disk to create a PV and a PVC**

- i. **Use a disk to create a PV**

You can create a PV in the console or by using a YAML file.

- Create a PV by using a YAML file

Use the following `disk-pv.yaml` file to create a PV.

 **Note** The PV name must be the same as the disk ID.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: d-bp1j17ifxfasvts3tf40
  labels:
    failure-domain.beta.kubernetes.io/zone: cn-hangzhou-b
    failure-domain.beta.kubernetes.io/region: cn-hangzhou
spec:
  capacity:
    storage: 20Gi
  storageClassName: disk
  accessModes:
    - ReadWriteOnce
  flexVolume:
    driver: "alicloud/disk"
    fsType: "ext4"
    options:
      volumeId: "d-bp1j17ifxfasvts3tf40"
```

- Create a PV in the console
 - a. Log on to the Container Service for Kubernetes console.
 - b. In the left-side navigation pane, choose Clusters > Persistent Volumes to go to the PVs and PVCs page.
 - c. On the Persistent Volumes tab, select the target cluster and click Create in the upper-right corner.
 - d. Set the parameters in the Create PV dialog box.

PV parameters

Parameter	Description
PV Type	In this example, Cloud Disk is selected.
Volume Plug-in	Flexvolume and CSI are supported.
Access Mode	By default, ReadWriteOnce is used.
Disk ID	Select a disk that is in the same region and zone as your cluster.
File System Type	Select the file system of the disk. Supported file systems include ext4, ext3, xfs, and vfat. Default value: ext4.
Label	Add labels to the PV.

The screenshot shows a 'Create PV' dialog box with the following configuration:

- PV Type:** Cloud Disk (selected), NAS, OSS
- Access Mode:** ReadWriteOnce (selected)
- Disk ID:** Select Disk
- File System Type:** ext4
- Label:** Add Label

Buttons: Create, Cancel

- e. Click Create.
- ii. Create a PVC

Use the following `disk-pvc.yaml` file to create a PVC:

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-disk
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: disk
resources:
  requests:
    storage: 20Gi

```

iii. Create a pod

Use the following *disk-pod.yaml* file to create a pod.

```

apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-alicloud-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: pvc-disk
          mountPath: "/data"
  volumes:
    - name: pvc-disk
      persistentVolumeClaim:
        claimName: pvc-disk

```

Dynamically provisioned volumes

To use a dynamically provisioned volume, you must manually create a StorageClass, and use `storageClassName` to specify the disk type in a PVC.

1. Create a StorageClass

```

kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: alicloud-disk-common-hangzhou-b
provisioner: alicloud/disk
parameters:
  type: cloud_ssd
  regionid: cn-hangzhou
  zoneid: cn-hangzhou-b

```

Parameters

- `provisioner`: Set the value to `alicloud/disk`, which means that the Alibaba Cloud Provisioner plug-in is

used to create the StorageClass.

- **type:** the type of the disk. Valid values: `cloud_efficiency`, `cloud_ssd`, and `available`. If you set this parameter to `available`, the system will try to create a disk in the following order: ultra disk, SSD disk, and basic disk. The system will stop trying until a disk is created.
- **regionid:** the region of the disk.
- **zoneid:** the zone of the disk.

2. Create a service

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: disk-common
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: alicloud-disk-common-hangzhou-b
  resources:
    requests:
      storage: 20Gi
---
kind: Pod
apiVersion: v1
metadata:
  name: disk-pod-common
spec:
  containers:
    - name: disk-pod
      image: nginx
      volumeMounts:
        - name: disk-pvc
          mountPath: "/mnt"
  restartPolicy: "Never"
  volumes:
    - name: disk-pvc
      persistentVolumeClaim:
        claimName: disk-common
```

Default options

By default, Kubernetes clusters support the following types of StorageClasses:

- `alicloud-disk-efficiency`: ultra disk.
- `alicloud-disk-ssd`: SSD disk.
- `alicloud-disk-available`: This option ensures high availability. The system tries to create an ultra disk first. If no ultra disk is available in the specified zone, the system tries to create an SSD disk.

3. Create a multi-instance StatefulSet by using a disk

We recommend that you use the `volumeClaimTemplates` parameter. This allows the system to dynamically create multiple PVCs and PVs. PVCs are associated with PVs and pods.

```
apiVersion: v1
```

```
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: "nginx"
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
              name: web
          volumeMounts:
            - name: disk-common
              mountPath: /data
      volumeClaimTemplates:
        - metadata:
            name: disk-common
          spec:
            accessModes: [ "ReadWriteOnce" ]
            storageClassName: "alicloud-disk-common"
            resources:
              requests:
                storage: 10Gi
```

3.4.4.3. Use NAS volumes

This topic describes how to use Network Attached Storage (NAS) file systems in Kubernetes clusters.

NAS file systems can be mounted to Kubernetes clusters as the following volume types:

- **Statically provisioned NAS volumes**

You can use statically provisioned NAS volumes in the following ways:

- Use the FlexVolume plug-in
 - Use a NAS file system as a volume
 - Use a NAS file system to create a persistent volume (PV) and a persistent volume claim (PVC)
- Use the Network File System (NFS) driver of Kubernetes.

- **Dynamically provisioned NAS volumes**

Prerequisites

A NAS file system is created in the NAS console and a mount target is added. The mount target is used to mount the file system to the Kubernetes cluster. The NAS file system and the Kubernetes cluster are deployed in the same Virtual Private Cloud (VPC) network.

Statically provisioned NAS volumes

You can use the FlexVolume plug-in provided by Alibaba Cloud or the NFS driver provided by Kubernetes to manage NAS file systems.

Use the FlexVolume plug-in

The FlexVolume plug-in allows you to use a NAS file system as a volume. You can also use a NAS file system to create a PV and a PVC.

 **Note**

- NAS is a shared storage system that provides storage services for multiple pods at the same time.
- server: the mount target of the NAS volume.
- path: the mounted directory in the NAS file system. You can specify a subdirectory as a volume. If no subdirectory exists, the system automatically creates and mounts a subdirectory.
- vers: the version of the NFS mounting protocol. Version 4.0 is supported.
- mode: the access permissions on the mounted directory. If the root directory of the NAS file system is specified as the mounted directory, you cannot modify the access permissions. If the NAS file system stores a large amount of data, the mounting process may take a long time or fail. Therefore, we recommend that you do not set the mode parameter.

Use a NAS file system as a volume

Use the following `nas-deploy.yaml` file to create a pod.

```

apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-nas-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: "nas1"
          mountPath: "/data"
  volumes:
    - name: "nas1"
      flexVolume:
        driver: "alicloud/nas"
        options:
          server: "0cd8b4a576-grs79.cn-hangzhou.nas.aliyuncs.com"
          path: "/k8s"
          vers: "4.0"

```

Use a NAS file system to create a PV and a PVC

Step 1: Create a PV

You can use a YAML file or the Container Service for Kubernetes console to create a PV.

- Create a PV by using a YAML file

Use the following `nas-pv.yaml` file to create a PV.

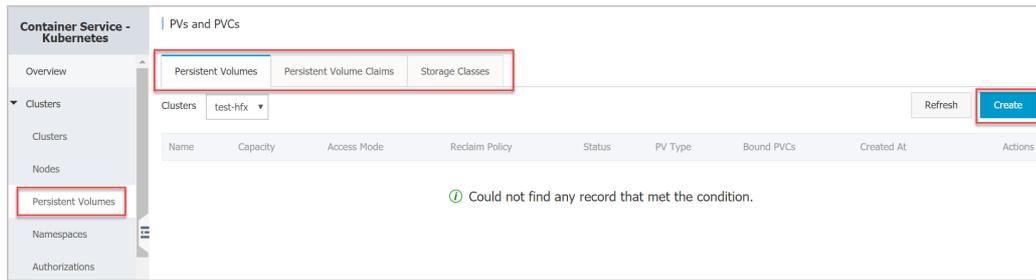
```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-nas
spec:
  capacity:
    storage: 5Gi
  storageClassName: nas
  accessModes:
    - ReadWriteMany
  flexVolume:
    driver: "alicloud/nas"
    options:
      server: "0cd8b4a576-uih75.cn-hangzhou.nas.aliyuncs.com"
      path: "/k8s"
      vers: "4.0"

```

- Create a PV in the console
 - Log on to the [Container Service for Kubernetes console](#).
 - In the left-side navigation pane, choose **Clusters > Persistent Volumes**. The **PVs and PVCs** page appears.

iii. On the **Persistent Volumes** tab, select the target cluster and click **Create** in the upper-right corner.



iv. In the **Create PV** dialog box, set the following parameters:

- **PV Type:** In this example, **NAS** is selected.
- **Volume Name:** the name of the PV. The name must be unique in the cluster. In this example, **pv-nas** is specified.
- **Volume Plug-in:** In this example, **Flexvolume** is selected.
- **Capacity:** the capacity of the PV. The capacity of the PV cannot exceed that of the **NAS** file system.
- **Access Mode:** The default mode is **ReadWriteMany**.
- **Mount Target Domain Name:** Enter the address of the mount target that is used to mount the **NAS** file system to the cluster.
- **Subdirectory:** Enter a subdirectory in the **NAS** file system. The subdirectory must start with a forward slash (/). If this parameter is set, the specified subdirectory is mounted as the PV.
 - If the specified subdirectory does not exist, the system automatically creates this subdirectory.
 - This parameter is optional. By default, the root directory of the **NAS** file system is mounted.
- **Permissions:** Set the access permissions on the mounted directory, for example, **755**, **644**, or **777**.
 - The permissions can be set only when a subdirectory is mounted as the PV.
 - This parameter is optional. By default, the original permissions are used.
- **chmod (Change Mode):** In this example, **Non-recursive** is selected.
- **Version:** The version of the **NFS** mounting protocol. Version 3 and 4.0 are supported. We recommend that you use the default version 3.
- **Label:** Add labels for the PV.

Create PV
✕

Make sure that FlexVolume is upgraded to the latest version.

PV Type Cloud Disk **NAS** OSS

* Volume Name:
The name can only contain lowercase letters, numbers, periods (.), and hyphens (-). It must start with a lowercase letter.

Volume Plug-in **Flexvolume** CSI
Flexvolume is not installed in the cluster. You may not be able to use the PV.

* Capacity

Access Mode **ReadWriteMany** ReadWriteOnce

* Mount Target Domain Name: **Select Mount Target** Custom

Subdirectory:
Permissions: [Configuration Guide](#)

chmod (Change Mode) Non-recursive Recursive [Configuration Guide](#)

Version

▲ Hide

Label [+ Add Label](#)

v. Click Create.

Step 2: Create a PVC

Use the following *nas-pvc.yaml* file to create a PVC.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-nas
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: nas
resources:
  requests:
    storage: 5Gi

```

Step 3: Create a pod

Use the following *nas-pod.yaml* file to create a pod.

```

apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-nas-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: pvc-nas
          mountPath: "/data"
  volumes:
    - name: pvc-nas
      persistentVolumeClaim:
        claimName: pvc-nas

```

Use the NFS driver**Step 1: Create a NAS file system**

Log on to the NAS console. For more information, see the *Create a file system* chapter of *NAS User Guide*.

 **Note** The NAS file system and the Kubernetes cluster must be deployed in the same region.

In this example, the following mount target is used: `055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com`.

Step 2: Create a PV

You can use a YAML template or the Container Service for Kubernetes console to create a PV.

- **Create a PV by using a YAML template**

Use the *nas-pv.yaml* file to create a PV.

Run the following command to create a PV from the NAS file system:

```
root@master # cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nas
spec:
  capacity:
    storage: 8Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  nfs:
    path: /
    server: 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com
EOF
```

- **Create a PV in the console**

For more information, see [Use a NAS file system to create a PV and a PVC.](#)

Step 3: Create a PVC

Create a PVC and associate the PVC with the PV that is created in Step 2.

```
root@master # cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nasclaim
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 8Gi
EOF
```

Step 4: Create a Pod

Create an application to use the PV.

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: myfrontend
    image: registry.aliyuncs.com/spacexnice/netdia:latest
    volumeMounts:
    - mountPath: "/var/www/html"
      name: mypd
  volumes:
  - name: mypd
    persistentVolumeClaim:
      claimName: nasclaim
EOF
```

The NAS file system is now mounted to the application that runs in the pod.

Dynamically provisioned NAS volumes

If you want to use dynamically provisioned NAS volumes, install a driver plug-in and configure a NAS mount target.

Install the plug-in

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: alicloud-nas
provisioner: alicloud/nas
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: alicloud-nas-controller
  namespace: kube-system
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: run-alicloud-nas-controller
subjects:
- kind: ServiceAccount
  name: alicloud-nas-controller
  namespace: kube-system
roleRef:
  kind: ClusterRole
```

```
name: alicloud-disk-controller-runner
apiGroup: rbac.authorization.k8s.io
---
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: alicloud-nas-controller
  namespace: kube-system
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: alicloud-nas-controller
    spec:
      tolerations:
        - effect: NoSchedule
          operator: Exists
          key: node-role.kubernetes.io/master
        - effect: NoSchedule
          operator: Exists
          key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
        node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-nas-controller
      containers:
        - name: alicloud-nas-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alibabacloud-nas-controller:v1.8.4
          volumeMounts:
            - mountPath: /persistentvolumes
              name: nfs-client-root
          env:
            - name: PROVISIONER_NAME
              value: alicloud/nas
            - name: NFS_SERVER
              value: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            - name: NFS_PATH
              value: /
      volumes:
        - name: nfs-client-root
          nfs:
            server: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            path: /
```

Use a dynamically provisioned NAS volume

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 2
  volumeClaimTemplates:
  - metadata:
    name: html
    spec:
      accessModes:
      - ReadWriteOnce
      storageClassName: alicloud-nas
      resources:
        requests:
          storage: 2Gi
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        volumeMounts:
        - mountPath: "/usr/share/nginx/html/"
          name: html
```

3.4.4.4. Use OSS volumes

You can use OSS buckets to create PVs in Kubernetes clusters.

You can use OSS volumes in the following ways:

- Directly as volumes
- Through PVs and PVCs

Prerequisites

You have created a bucket in the OSS console.

Background

- Currently, you can only use static OSS volumes.
- OSS is a shared storage system that can provide storage services to multiple pods at the same time.
- bucket: Only buckets can be mounted to a Kubernetes cluster. The subdirectories or files in a bucket cannot be mounted to a Kubernetes cluster.
- url: The OSS endpoint, namely, the domain that is used to mount an OSS bucket to a cluster.

- **akId**: Your AccessKey ID.
- **akSecret**: Your AccessKey secret.
- **otherOpts**: The custom parameters for mounting an OSS bucket, in the format of `-o *** -o ***`.

 **Note** To use OSS volumes, you must create a secret with your AccessKey information when you deploy the flexvolume service.

Use static OSS volumes

- **Use an OSS bucket as a volume**

Use the following *oss-deploy.yaml* file to create a pod.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-oss-deploy
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx-flexvolume-oss
          image: nginx
          volumeMounts:
            - name: "oss1"
              mountPath: "/data"
      volumes:
        - name: "oss1"
          flexVolume:
            driver: "alicloud/oss"
            options:
              bucket: "docker"
              url: "oss-cn-hangzhou.aliyuncs.com"
              akId: ***
              akSecret: ***
              otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

- **Use an OSS bucket to create a PV and a PVC**

i. Create a PV

You can create a PV through the console or by using a YAML file.

■ Create a PV by using a YAML file

Use the following *oss-pv.yaml* file to create a PV.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-oss
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  storageClassName: oss
  flexVolume:
    driver: "alicloud/oss"
    options:
      bucket: "docker"
      url: "oss-cn-hangzhou.aliyuncs.com"
      akId: ***
      akSecret: ***
      otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

■ Create a PV through the console

- a. [Log on to the Container Service console.](#)
- b. In the left-side navigation pane, choose **Clusters > Persistent Volumes**. The **PVs and PVCs** page appears.
- c. On the **Persistent Volumes** tab, select the target cluster and click **Create** in the upper-right corner.
- d. In the **Create PV** dialog box, set the following parameters.

Create a PV through the console

Parameter	Description
PV Type	In this example, select OSS.
Volume Name	The name of the PV. The name must be unique in the cluster. In this example, enter pv-oss.
Volume Plugin-in	Supports Flexvolume and CSI.
Capacity	The capacity of the PV.
Access Mode	Default is ReadWriteMany.
AccessKey ID and AccessKey Secret	The AccessKey pair is required to access OSS buckets. To obtain your AccessKey pair, go to the ASCM console, choose Enterprise > Organizations , click  at the right of the target organization, and click AccessKey .
Optional Parameters	Enter custom parameters in the format of <code>-o *** -o ***</code> .

Parameter	Description
Bucket ID	The name of the OSS bucket that you want to use. Click Select Bucket . In the dialog box that appears, choose the target bucket and click Select .
Endpoint	We recommend that you choose Internal Endpoint .
Label	Add labels to the PV.

e. Click **Create**.

ii. Create a PVC

Use the following *oss-pvc.yaml* file to create a PVC.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-oss
spec:
  storageClassName: oss
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
```

iii. Create a pod

Use the following *oss-pod.yaml* file to create a pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: "flexvolume-oss-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: pvc-oss
          mountPath: "/data"
  volumes:
    - name: pvc-oss
      persistentVolumeClaim:
        claimName: pvc-oss
```

Use dynamic OSS volumes

Currently, dynamic OSS volumes are not supported.

3.4.4.5. Create a PVC

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created a PV. This example uses a PV created from a disk. For more information, see [Use Apsara Stack disks](#).

By default, PVCs are associated with PVs that have the `alicloud-pvname` label. PVs created through the Container Service console all have this label. If a PV does not have this label, you need to add the label before you can associate it with a PVC.

Procedure

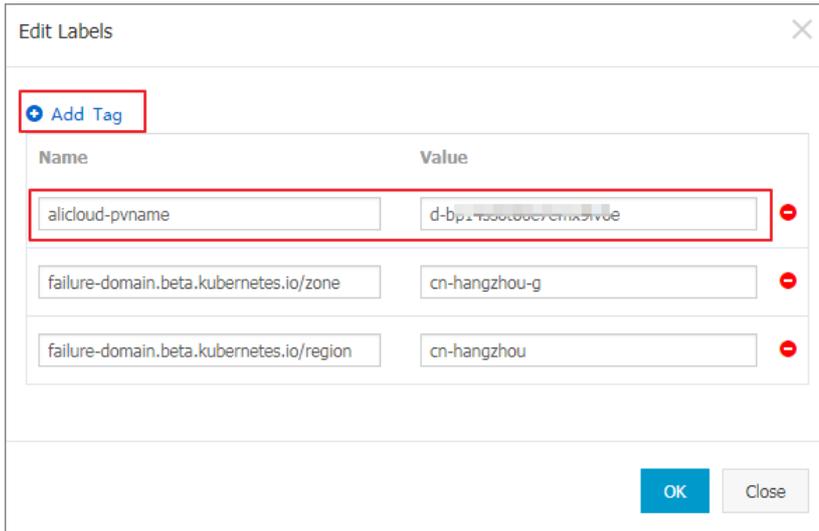
1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Persistent Volume Claims**. The **PVs and PVCs** page appears.

3. On the **Persistent Volume Claims** tab, select the target cluster and namespace, and click **Create** in the upper-right corner.
4. In the **Create PVC** dialog box, set the parameters, and click **Create**.

- **Source:** Select an existing PV or use a storage class.
- **PVC Type:** The same as the types of PVs. Three types, cloud disk, NAS, and OSS, are supported.
- **Name:** The name of the PVC.
- **Allocation Mode:** Currently, only existing volumes are supported.
- **Existing Volumes:** Select PVs of the same type as the PVC.
- **Capacity:** The claimed usage, which cannot be larger than the total capacity of associated PVs.

Note If your cluster has a PV that is not used, but you cannot find it in the **Select PV** dialog box, the reason may be that the PV does not have the `alicloud-pvname` label.

If you cannot find available PVs, you can choose **Clusters > Persistent Volumes** in the left-side navigation pane. Find the PV that you want to use and click **Manage Labels** in the **Actions** column. You can attach a label to the PV and set the label name to `alicloud-pvname` and the value to the PV name. By default, the disk ID is used as the PV name if the PV is created from a disk.



5. On the Persistent Volume Claims page, the newly created PVC is now displayed.

3.4.4.6. Use a PVC

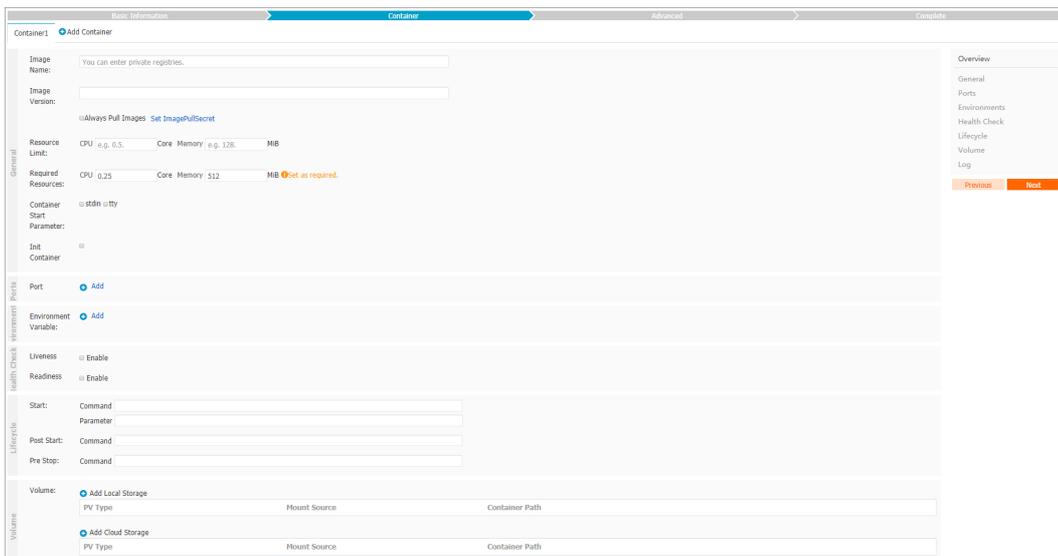
You can use persistent volume claims (PVCs) in your applications.

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created a PVC. This example uses a PVC named pvc-disk. For more information, see [Create PVCs](#).

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears. In the upper-right corner, click **Create from Image**.
3. On the **Basic Information** page, specify the application name, cluster, namespace, number of replicas, type, labels, and annotations. Then click **Next**.
4. On the **Container** page, select the image. Then, specify the type of cloud volume. Currently, cloud disks, NAS, and OSS are supported. In this example, select the pvc-disk PVC and click **Next**.



5. Configure the test-nginx application, and then click **Create**.

- After the application is created, choose **Applications > Pods** in the left-side navigation pane. Select the pod to which the application belongs, and click **View Details**.
- On the pod details page, click the **Volumes** tab. Verify that the pod is now associated with the **pvc-disk** PVC.

Pod details page showing Overview, Conditions, and Volumes tabs. The Volumes tab is active, displaying a table with one volume entry:

Name	Type	Details
volume-1530693170118	persistentVolumeClaim	claimName: pvc-disk

3.4.5. Network management

3.4.5.1. Set access control for pods

This topic describes how to use network policies to control access between pods.

Prerequisites

You have created a Kubernetes cluster and selected the **Terway network plug-in**. For more information, see [Create a Kubernetes cluster](#).

Context

You can declare network policies to control access between pods and thus prevent applications from interfering each other.

Procedure

For more information about standard Kubernetes network policies, see [Network policies](#).

- Create a pod that runs as a server and attach `label run=nginx` to the pod. For more information, see [Create an application from an orchestration template](#). The sample YAML file is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: server
  labels:
    run: nginx
spec:
  containers:
  - name: nginx
    image: registry.acs.intranet.env22.com/nginx:1.8
```

2. Create a network policy. For more information, see [Create an application from an orchestration template](#). The sample YAML file is as follows:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: access-nginx
spec:
  podSelector:
    matchLabels:
      run: nginx # Apply the network policy to pods with the run=nginx label
  ingress:
    - from:
      - podSelector:
          matchLabels:
            access: "true" # Only pods with the access=true label are accessible
```

3. Use the *client.yaml* and *client-label* files to create two pods that run as clients. One pod has the required label and the other does not.
 - i. Create the *client.yaml* and *client-label* files with the following contents respectively.

```
# This pod has no label
apiVersion: v1
kind: Pod
metadata:
  name: client
spec:
  containers:
    - name: busybox
      image: registry.acs.intranet.env22.com/acs/busybox
      command: ["sh", "-c", "sleep 200000"]
```

```
# This pod has the label
apiVersion: v1
kind: Pod
metadata:
  name: client-label
  labels:
    access: "true"
spec:
  containers:
    - name: busybox
      image: registry.acs.intranet.env22.com/acs/busybox
      command: ["sh", "-c", "sleep 200000"]
```

- ii. Run the following commands to create these pods:

```
kubectl apply -f client.yaml
kubectl apply -f client-label.yaml
```

You can see that only the pod with the required label can access the server.

3.4.5.2. Set bandwidth limits for pods

This topic describes how to limit the bandwidth of inbound and outbound traffic that flows through a pod.

Prerequisites

You have created a Kubernetes cluster and selected the Terwaynetwork plug-in. For more information, see [Create a Kubernetes cluster](#).

Context

Throttling pods helps prevent performance degradation of the host or other workloads when certain pods occupy excessive resources.

Method

You can use the `k8s.aliyun.com/ingress-bandwidth` and `k8s.aliyun.com/egress-bandwidth` annotations for pod throttling.

- `k8s.aliyun.com/ingress-bandwidth` : limits the pod inbound bandwidth.
- `k8s.aliyun.com/egress-bandwidth` : limits the pod outbound bandwidth.
- The bandwidth limit is measured in m and k, which represent Mbit/s and Kbit/s respectively.

Procedure

1. Create a pod that runs as a server in the console. For more information, see [Create an application from an orchestration template](#). The sample YAML file is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: server
  annotations:
    k8s.aliyun.com/ingress-bandwidth: 10m # Set the inbound bandwidth limit to 10 Mbit/s
    k8s.aliyun.com/egress-bandwidth: 10m
spec:
  containers:
    - name: nginx
      image: registry.acs.intranet.env22.com/nginx:1.8
```

2. Run the `kubectl exec` command to connect to the pod. To verify that pod throttling is effective, run the following commands to create a file on the pod. Assume that the IP address of the pod created in [step 1](#) is 172.16.XX.XX.

```
cd /usr/share/nginx/html
dd if=/dev/zero of=bigfile bs=1M count=1000
```

3. Use the `client-deploy.yaml` file to create a pod that runs as a client.

- i. Create the `client-deploy.yaml` file with the following content:

```
apiVersion: v1
kind: Pod
metadata:
  name: client
  annotations:
    k8s.aliyun.com/ingress-bandwidth: 10m # Set the inbound bandwidth limit to 10 Mbit/s
    k8s.aliyun.com/egress-bandwidth: 10m
spec:
  containers:
    - name: busybox
      image: registry.acs.intranet.env22.com/acs/netdia
      command: ["sh", "-c", "sleep 200000"]
```

- ii. Run the following command to create the pod:

```
kubectl apply -f client-deploy.yaml
```

4. Run the following command to check whether bandwidth is limited:

```
kubectl exec -it client sh
```

3.4.6. Namespaces

3.4.6.1. Create a namespace

You can create a namespace in the console.

Prerequisites

You have created a Kubernetes cluster.

Context

In a Kubernetes cluster, you can use namespaces to create multiple virtual spaces. When a large number of users share a cluster, you can use namespaces to divide different workspaces and allocate cluster resources to different tasks. Furthermore, you can use [resource quotas](#) to allocate resources to each namespace.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster and click **Create** in the upper-right corner.
4. In the dialog box that appears, set the parameters.

Create Namespace

Name:

The namespace name must be 1 to 63 characters in length and can contain numbers lowercase letters and hyphens (-). It must start with a letter or a number.

Variable Key	Variable Value	Actions
env	test	Edit Delete

Create a namespace

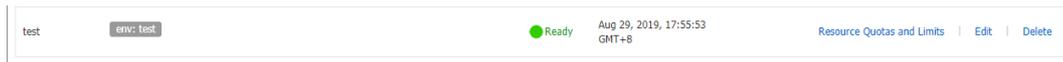
Parameter	Description
Name	Enter a name for the namespace. In this example, enter test. The name must be 1 to 63 characters in length and can contain digits, letters, and hyphens (-). It must start and end with a letter or digit.
Label	<p>Label: Add one or more labels to the namespace. Labels are used to add marks to namespaces. For example, you can use a label to mark the namespace as one used in the test environment.</p> <p>Enter a variable key and value. Then click Add on the right to add a label.</p>

5. Click **OK**.
6. The newly created namespace is now displayed on the **Namespaces** page.

Namespace

Clusters: k8s-cluster01

Name	Label	Status	Created At	Actions
default		● Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete
kube-public		● Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete
kube-system		● Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete



3.4.6.2. Set resource quotas and limits

You can set resource quotas and limits for a namespace in the console.

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created the sample namespace test. For more information, see [Create a namespace](#).
- You can connect to a master node of the cluster. For more information, see [Connect to a Kubernetes cluster through kubectl](#).

Context

By default, a running pod uses the CPU and memory resources of a node without limit. This means that any pod can use the computing resources of a cluster without restraints. Therefore, pods of a namespace may deplete the cluster resources.

Namespaces can be used as virtual clusters to serve multiple purposes and meet different needs. We recommend that you set resource quotas for all namespaces.

For a namespace, you can set quotas on resources such as CPU, memory, and pod quantity. For more information, see [Resource Quotas](#).

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Namespaces**. On the Namespaces page, select the target cluster. Find the target namespace and click **Resource Quotas and Limits** in the Actions column.
3. In the dialog box that appears, set resource quotas and default resource limits.

Note After you set CPU and memory quotas for a namespace, you must specify CPU and memory limits when you create a pod. Alternatively, you can set the default resource limits for the namespace. For more information, see [Resource Quotas](#).

i. Set resource quotas for the namespace.

The screenshot shows the 'Resource Quotas and Limits' dialog box with the 'Resource Quota' tab selected. The settings are as follows:

Category	Resource	Total	Unit
Compute Resource Quota	CPU Limit	2	Cores
	Memory Limit	4Gi	
Storage Resource Quota	Storage Capacity	1024Gi	
	PVCs	50	
Other Limits	ConfigMaps	100	
	Pods	50	
	Services	20	
	Load Balancer Services	5	
	Secrets	10	

ii. You can set resource limits and resource requests for containers in the namespace. This enables you to control the amount of resources consumed by containers. For more information, see <https://kubernetes.io/memory-default-namespace/>.

The screenshot shows the 'Resource Quotas and Limits' dialog box with the 'LimitRange' tab selected. The settings are as follows:

Resource	Limit	Request	Unit
CPU	0.5	0.1	Cores
Memory	512Mi	256Mi	

4. After you set resource quotas and limits, connect to a master node of the cluster and run the following commands to query the resource configurations of the namespace.

```
# kubectl get limitrange,ResourceQuota -n test
NAME AGE
limitrange/limits 8m

NAME AGE
resourcequota/quota 8m

# kubectl describe limitrange/limits resourcequota/quota -n test
Name: limits
Namespace: test
Type Resource Min Max Default Request Default Limit Max Limit/Request Ratio
-----
Container cpu -- 100m 500m -
Container memory -- 256Mi 512Mi -

Name: quota
Namespace: test
Resource Used Hard
-----
configmaps 0 100
limits.cpu 0 2
limits.memory 0 4Gi
persistentvolumeclaims 0 50
pods 0 50
requests.storage 0 1Ti
secrets 1 10
services 0 20
services.loadbalancers 0 5
```

3.4.6.3. Edit a namespace

You can edit an existing namespace.

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created the sample namespace test. For more information, see [Create a namespace](#).

Context

When you edit a namespace, you can add, delete, or modify namespace labels based on needs.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster. Find the namespace that you want to edit and click **Edit** in the Actions column.
4. In the dialog box that appears, select a label and click **Edit** to modify its key and value. This example changes a label to `env:test-V2`. Then click **Save**.

5. Click **OK**. Go to the Namespaces page and check the newly edited namespace label.

Name	Label	Status	Created At	Actions
default		Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete
kube-public		Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete
kube-system		Ready	Aug 29, 2019, 13:29:29 GMT+8	Resource Quotas and Limits Edit Delete
test	env: test-V2	Ready	Aug 29, 2019, 17:55:53 GMT+8	Resource Quotas and Limits Edit Delete

3.4.6.4. Delete a namespace

You can delete namespaces that are no longer in use.

Prerequisites

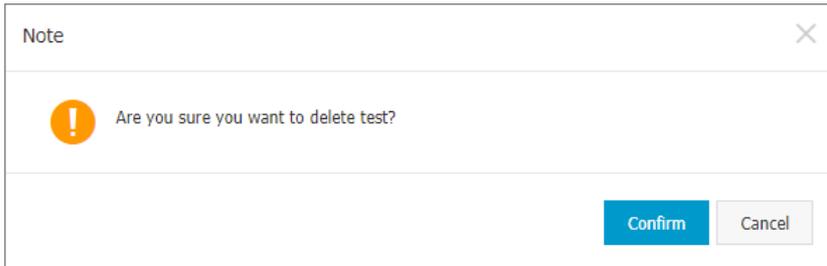
- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created the sample namespace test. For more information, see [Create a namespace](#).

Context

Note When you delete a namespace, all resource objects under the namespace will be deleted. Exercise caution when you perform this operation.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Namespaces**. The Namespaces page appears.
3. Select the target cluster. Find the namespace that you want to delete and click **Delete** in the Actions column.
4. In the dialog box that appears, click **Confirm**.



5. The namespace is now deleted from the Namespaces page. Resource objects under the namespace are also deleted.

3.4.7. Applications

3.4.7.1. Create an application from an image

You can use an image to create an NGINX application that is accessible over the Internet.

Prerequisites

- A Kubernetes cluster is created. For more information, see [Create a Kubernetes cluster](#).
- Your Kubernetes cluster is accessible over the Internet.

Procedure

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Applications > Deployments** to go to the Deployments page. Then, click **Create from Image** in the upper-right corner of the page.
3. In the dialog box that appears, set the parameters, including **Name**, **Cluster**, **Namespace**, **Replicas**, and **Type**, and select **Synchronize Timezone**. Click **Next**.

If you do not set the **Namespace** parameter, the default namespace is used.

The screenshot shows the 'Create Application' dialog box with the following fields and values:

- Name:** nginx
- Cluster:** k8s-cluster01
- Namespace:** default
- Replicas:** 2
- Type:** Deployment

At the bottom right, there are 'Back' and 'Next' buttons.

4. **Configure containers.**

Note You can configure multiple containers for the pods of the application.

i. Configure general settings.

Container general settings

Parameter	Description
Image Name	You can click Select Image , and in the dialog box that appears, select the required image and click OK . In this example, NGINX is selected. You can also enter a private registry URL to specify the image. The registry URL follows this format: <code>domainname/namespace/imagename</code> .
Image Version	You can click Select Image Version to select the required version. If you do not specify the image version, the latest version is used.
Always Pull Images	To improve efficiency, Container Service for Kubernetes caches images. During the deployment, if the version of the specified image is the same as that of a cached image, Container Service for Kubernetes will reuse the cached image instead of pulling the image again. Therefore, when you update the application code, if you do not change the image version for reasons such as to support the upper-layer workloads, the previously cached image will be used. When this check box is selected, Container Service for Kubernetes will always pull the image from the repository to deploy the application. This ensures that the latest image and code are used.
Set Image Pull Secret	Click Set Image Pull Secret to set the secret. The secret is required if you need to access a private repository.
Resource Limit	The upper limits of CPU and memory resources that are available to this application. This prevents the application from occupying excessive resources. The unit of CPU resources is Core. The unit of memory is MiB.
Required Resources	The amount of CPU and memory resources that are reserved for this application. These resources are exclusive to the container. This prevents the application from being unavailable if other services or processes share the resources.
Container Start Parameter	Select stdin to enable standard inputs for the container. Select tty to assign a virtual terminal that is used to send signals to the container. We recommend that you select both check boxes. This allows you to associate the terminal (tty) with the standard inputs (stdin) of the container. For example, an interactive program can be used to obtain standard inputs from users and then display the inputs on the terminal.
Init Container	When this check box is selected, the system creates an Init Container that contains useful tools. For more information, see .

ii. (Optional)Set environment variables.

You can use key-value pairs to set environment variables for pods. Environment variables are used to apply pod configurations to containers. For more information, see [Pod variable](#).

iii. (Optional)Configure health check settings.

Health check settings include liveness and readiness probes. Liveness probes determine when to restart the container. Readiness probes determine whether the container is ready to start accepting traffic. For more information about health checks, see .

The screenshot shows two sections for configuring health checks: Liveness and Readiness. Both sections have an 'Enable' checkbox checked. The 'Request type' is set to 'HTTP Request' for both. The 'Protocol' is set to 'HTTP'. The 'Path' and 'Port' fields are empty. The 'HTTP Header' section has 'name' and 'value' input fields. The 'Initial Delay (s)' is 3, 'Period (s)' is 10, and 'Timeout (s)' is 1. For the Liveness probe, 'Success' is 1 and 'Failure Threshold' is 3. For the Readiness probe, 'Success' is 1 and 'Failure Threshold' is 3.

Request type	Description
<p>HTTP request</p>	<p>Sends an HTTP GET request to the container. Supported parameters include:</p> <ul style="list-style-type: none"> ■ Protocol: HTTP or HTTPS. ■ Path: the requested path on the server. ■ Port: the port opened in the container. The port number must range from 1 to 65535. ■ HTTP Header: The custom headers in the HTTP request. Duplicate headers are allowed. Key-value pairs are supported. ■ Initial Delay (s): the <code>initialDelaySeconds</code> field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 3. ■ Period (s): the <code>periodSeconds</code> field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ■ Timeout (s): the <code>timeoutSeconds</code> field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ■ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ■ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.

Request type	Description
TCP connection	<p>Sends a TCP socket to the container. The kubelet will attempt to open the socket on the specified port. If the connection can be established, the container is considered healthy. Otherwise, it is considered unhealthy. Supported parameters include:</p> <ul style="list-style-type: none"> ▪ Port: the port opened in the container. The port number must range from 1 to 65535. ▪ Initial Delay (s): the <code>initialDelaySeconds</code> field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 15. ▪ Period (s): the <code>periodSeconds</code> field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ▪ Timeout (s): the <code>timeoutSeconds</code> field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ▪ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ▪ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.
Command line	<p>Runs a probe command in the container to check its health status. Supported parameters include:</p> <ul style="list-style-type: none"> ▪ Command: the probe command that is used to check the health status of the container. ▪ Initial Delay (s): the <code>initialDelaySeconds</code> field, which specifies the period between when the container is started and when the system performs the first probe. Unit: seconds. Default value: 15. ▪ Period (s): the <code>periodSeconds</code> field, which specifies the intervals between two adjacent probes. Unit: seconds. Default value: 10. Minimum value: 1. ▪ Timeout (s): the <code>timeoutSeconds</code> field, which specifies the period after which a probe times out. Unit: seconds. Default value: 1. Minimum value: 1. ▪ Healthy Threshold: the minimum number of consecutive successes that must occur for a probe to be considered successful after having failed. Default value: 1. Minimum value: 1. For liveness probes, this parameter must be set to 1. ▪ Unhealthy Threshold: the minimum number of consecutive failures that must occur for a probe to be considered failed after having succeeded. Default value: 3. Minimum value: 1.

iv. Configure lifecycle events.

You can set the following parameters to configure the lifecycle of the container in the Start, Post Start, and Pre Stop fields. For more information, visit <https://kubernetes.io/docs/tasks/configure-pod-container/attach-handler-lifecycle-event/>.

- **Start:** the pre-start command and parameter.
- **Post Start:** the post-start command.
- **Pre Stop:** the pre-stop command.

Lifecycle	Start:	Command	<code>[\"/bin/sh\", \"-c\", \"echo Hello > /user/share/message\"]</code>
		Parameter	
	Post Start:	Command	
	Pre Stop:	Command	<code>[\"/user/sbin/nginx\", \"-s\", \"quit\"]</code>

v. (Optional)Configure volumes.

Local storage and cloud storage are supported.

- **Local Storage:** supports hostPaths, ConfigMaps, secrets, and temporary directories, and mounts a mount source to the specified path in the container. For more information, see [Volumes](#).
- **Cloud Storage:** supports three types of persistent volumes (PVs): cloud disks, Network Attached Storage (NAS), and Object Storage Service (OSS).

In this example, a PV is created from a cloud disk, and the PV is mounted to the `/tmp` path in the container. Data generated in this path is stored in the cloud disk.

Volume	Volume:										
	<div style="display: flex; justify-content: space-between;"> + Add Local Volume </div> <table border="1"> <thead> <tr> <th>PV Type</th> <th>Mount Source</th> <th>Container Path</th> </tr> </thead> <tbody> <tr> <td colspan="3">+ Add Cloud Volume</td> </tr> <tr> <td>Disk</td> <td>pvc-yunpan-test</td> <td>/tmp</td> </tr> </tbody> </table>			PV Type	Mount Source	Container Path	+ Add Cloud Volume			Disk	pvc-yunpan-test
PV Type	Mount Source	Container Path									
+ Add Cloud Volume											
Disk	pvc-yunpan-test	/tmp									

vi. (Optional)Configure Log Service. You can configure collection methods and custom tags.

Note Make sure that the Log Service agent is installed in the cluster.

Log collection parameters include:

- **Logstore:** the Logstore that is used to store log data in Log Service.
- **Log Path in Container:** You can set this parameter to stdout or a log path.
 - **stdout:** collects standard outputs of the container.
 - **Log Path:** collects logs in the specified path of the container. In this example, logs in the following path are collected: `/var/log/nginx`. Wildcards are supported.

You can also set custom tags, which will be collected and output along with logs. Custom tags simplify statistical analysis of log data.

5. Set other parameters based on your needs and click **Next**.

6. Configure advanced settings, including the access control settings. You can configure the method to access pods and click **Create** to create the application. In this example, a service of the Cluster IP type and an Ingress are created to enable Internet access to the NGINX application.

Note

You can configure access control settings based on your needs:

- **Internal applications:** For applications that run inside the cluster, you can create a service of the Cluster IP or Node Port type to enable internal communication to fit your needs.
- **External applications:** For applications that need to be accessed over the Internet, you can configure access control settings by using one of the following methods:
 - Create a service of the Server Load Balancer (SLB) type and enable access to your application over the Internet by using the SLB instance.
 - Create a service of the Cluster IP or Node Port type, create an Ingress, and then enable access to your application over the Internet by using the Ingress. For more information, see .

- i. To create a service, click **Create** in the **Service** section. In the dialog box that appears, configure the service and click **Create**.

The screenshot shows a 'Create Service' dialog box with the following fields and values:

- Name:** nginx-svc
- Type:** Cluster IP
- Headless Service:**
- Backend:** (empty)
- Port Mapping:**

Service Port	Container Port	Protocol
8080	8080	TCP
- Annotations:** + Add
- Label:** + Add

Buttons: Create, Cancel

Parameter	Description
Name	Enter a name for the service. Default value: <code>applicationname-svc</code> .
Type	<p>Select one of the following types:</p> <ul style="list-style-type: none"> Cluster IP: enables access to the service through an internal IP address of the cluster. If you select this type, the service is only accessible within the cluster. Node Port: enables access to the service through the IP address and static port on each node. The NodePort field specifies the static port. A NodePort service can be used to route requests to a Cluster IP service. The system automatically creates the Cluster IP service. You can access a Node Port service from outside the cluster by requesting <code><NodeIP>:<NodePort></code>. Server Load Balancer: enables access to the service by using an SLB instance over the Internet or an internal network. The SLB instance can route requests to Node Port and Cluster IP services.
Port Mapping	Set a service port and a container port. If the Type parameter is set to Node Port, you must set a node port to avoid port conflicts. TCP and UDP protocols are supported.
Annotations	Add annotations to the service. SLB parameters are supported. For more information, see Access services by using SLB .
Label	Add labels to the service.

- ii. To create an Ingress, click **Create** in the **Ingress** section. In the dialog box that appears, configure Ingress rules and click **Create**. For more information about Ingress configuration, see [Ingress configurations](#). When you create an application from an image, you can create an Ingress for only one service. In this example, a virtual host name is used as the test domain. You must add the following entry to the hosts file to map the domain to the IP address of the Ingress. In actual scenarios, use a domain that has obtained an ICP number.

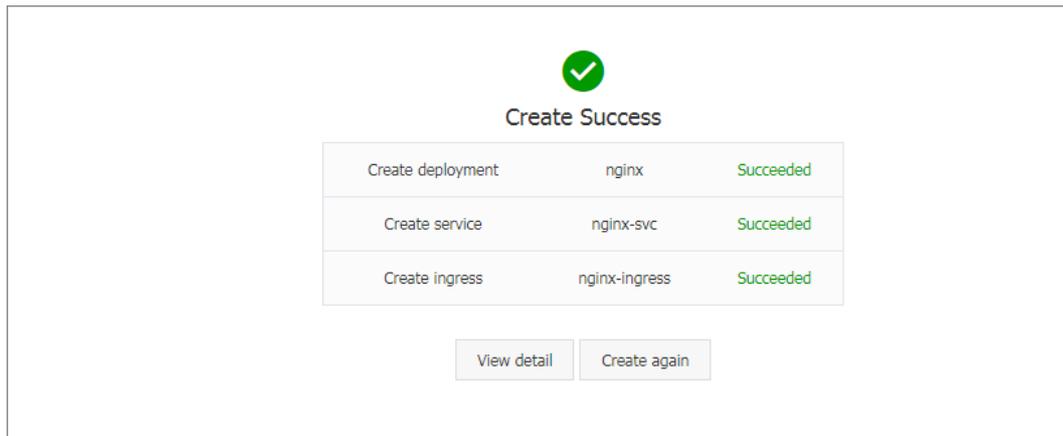
```
101.37.224.146 foo.bar.com #The IP address of the Ingress.
```

- iii. You can find the newly created service and Ingress in the **Access Control** section. You can click **Update** or **Delete** to make changes.

service port	Container Port	Protocol
8080	8080	TCP

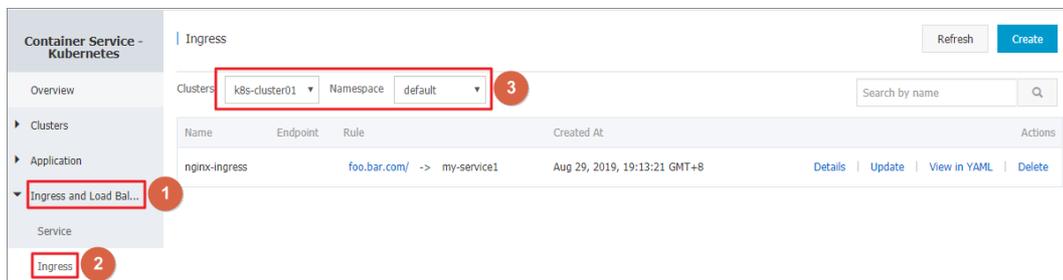
Domain	path	Name	service port
foo.bar.com		nginx-svc	8080

- 7. Click **Create**.
- 8. After the application is created, a message appears to display the resource objects included in the application. You can click **View Details** to view application details.



The nginx-deployment page is displayed by default.

- In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The following rule is displayed on the Ingresses page.



- Enter the test domain in the address bar of your browser and press the Enter key. The NGINX welcome page appears.



3.4.7.2. Create an application from an orchestration template

Container Service provides orchestration templates that you can use to create applications quickly. You can also modify the templates based on YAML syntax to customize applications.

Prerequisites

You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

Context

The following example demonstrates how to create an NGINX application consisting of a deployment and a service. The service is associated with a pod created by the deployment.

Procedure

- Log on to the [Container Service console](#).
- In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.

3. In the upper-right corner, click **Create from Template**.
4. Set the parameters and click **Create**.
 - **Cluster:** Select the cluster where the resource objects are to be deployed.
 - **Namespace:** Select the namespace to which the resource objects belong. The default namespace is default. Except for underlying computing resources such as nodes and PVs, most resources are scoped to namespaces.
 - **Sample Template:** Container Service provides YAML templates of various resource types to help you deploy resource objects quickly. You can also create a custom template based on YAML syntax to describe the resource that you want to define.
 - **Add Deployment:** This feature allows you to quickly define a YAML template.
 - **Use Existing Template:** You can import an existing template to the configuration page.

The screenshot shows the 'Create from Template' configuration page in the Container Service console. The 'Clusters' dropdown is set to 'k8s-cluster', 'Namespace' is 'default', and 'Sample Template' is 'Custom'. The 'Template' field contains a YAML configuration for an NGINX deployment and service. A green progress bar at the bottom indicates 'The creation process has started. Click here to check the progress: Kubernetes Dashboard'. The 'Create' button is highlighted with a red circle and the number '1', and the 'Kubernetes Dashboard' link in the progress bar is highlighted with a red circle and the number '2'.

```

1  apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment
5    labels:
6      app: nginx
7  spec:
8    replicas: 2
9    selector:
10   matchLabels:
11     app: nginx
12   template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       containers:
18         - name: nginx
19           image: nginx:1.7.9 # replace it with your exactly <image_name>:tags
20           ports:
21             - containerPort: 80
22   ---
23
24
25  apiVersion: v1 # for versions before 1.8.0 use apps/v1beta1
26  kind: Service
27  metadata:
28    name: my-service1 #T000: to specify your service name
29    labels:
30      app: nginx
31  spec:
32    selector:
33      app: nginx #T000: change label selector to match your backend pod
34    ports:
35      - protocol: TCP
36        name: http

```

Based on an orchestration template provided by Container Service, the following sample template creates a deployment of an NGINX application.

Note Container Service supports YAML syntax. You can use the `---` symbol to separate multiple resource objects. This enables you to create multiple resource objects in a single template.

```
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80

---

apiVersion: v1 # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
  name: my-service1 #TODO: to specify your service name
  labels:
    app: nginx
spec:
  selector:
    app: nginx #TODO: change label selector to match your backend pod
  ports:
  - protocol: TCP
    name: http
    port: 30080 #TODO: choose an unique port on each node to avoid port conflict
    targetPort: 80
  type: LoadBalancer ## This example changes the service type from NodePort to LoadBalancer.
```

5. Click **Create**. A message appears indicating the deployment status.

In the left-side navigation pane, choose **Ingresses and Load Balancing > Services** to view the newly created service.

6. On **Kubernetes Dashboard**, verify that a `my-service1` service is running and its external endpoint is displayed. Click the address in the **External Endpoint** column.

Name	Type	Created At	ClustersIP	InternalEndpoint	ExternalEndpoint	Actions
my-service1	LoadBalancer	Aug 29, 2019, 14:59:39 GMT+8	[Redacted]	my-service1:30080 TCP my-service1:30134 TCP	[Redacted]:30080	Details Update View in YAML Delete

7. You can visit the NGINX welcome page in the browser.



What's next

You can also choose **Ingresses and Load Balancing > Services** in the left-side navigation pane to view the NGINX service.

3.4.7.3. Create an application through Kubernetes Dashboard

You can create an application through Kubernetes Dashboard.

Procedure

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column to go to Kubernetes Dashboard.

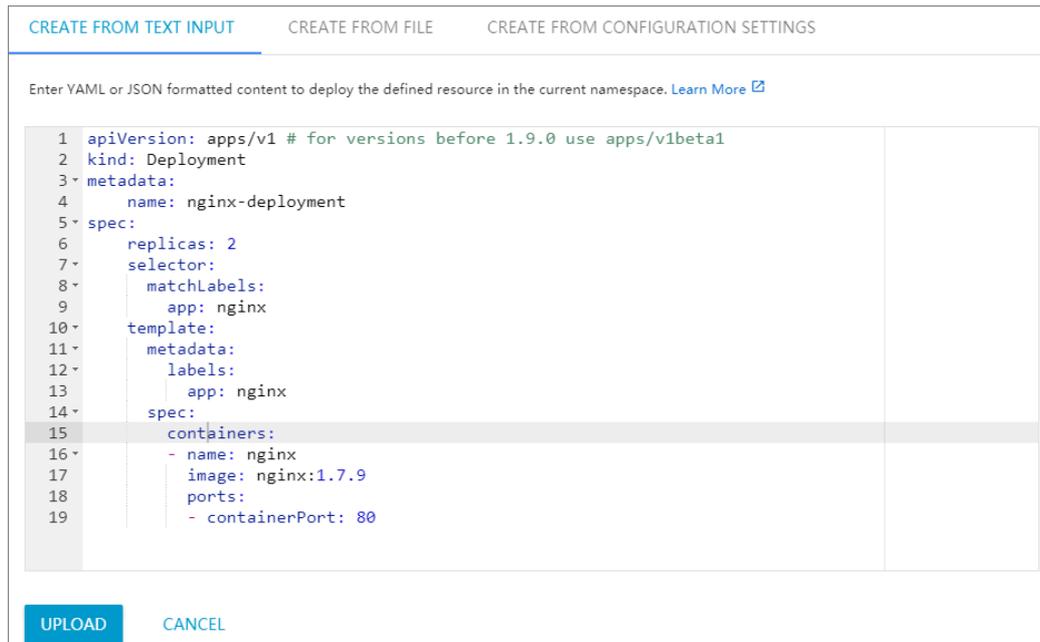
Cluster Name/ID	Type	Region (All)	Network Type	Cluster Status	Nodes	Created At	Version	Actions
k8s-cluster	Kubernetes	cn-qingdao-16e-d02	VPC VDC-	Running	6	Aug 29, 2019, 14:02:00 GMT+8	1.12.6-aliyun.1	Manage View Logs Dashboard Scale Cluster More

4. On the Overview page, click **CREATE** in the upper-right corner.

5. In the dialog box that appears, configure the application.

You can use one of the following methods to create the application.

- **CREATE FROM TEXT INPUT:** Directly enter an orchestration template in YAML or JSON format to create the application. A sample template in YAML format is provided as follows:



- **CREATE FROM FILE:** Import a YAML or JSON configuration file to create the application.
- **CREATE AN APP:** Set parameters to create the application.

Application parameters

Parameter	Description
App name	The name of the application. In this example, enter <code>nginx</code> .
Container image	The URL of the image. In this example, a Docker NGINX image is used.
Number of pods	The number of pods that constitute the application.
Service	Valid values: External and Internal . External: Create a service that is accessible from outside the cluster. Internal: Create a service that is accessible within the cluster.
Advanced options	Click SHOW ADVANCED OPTIONS to set labels and environment variables. The following setting distributes traffic to three pods based on load balancing.

CREATE FROM TEXT INPUT
 CREATE FROM FILE
 CREATE FROM CONFIGURATION SETTINGS

App Name *
nginx-test 10 / 24

Container Image *
nginx

Number of Pods *
3

Service *
External

Port *	Target Port *	Protocol *
80	9080	TCP
Port	Target Port	Protocol

An "app" label with the specified value will be added to the Deployment and service. [Learn More](#)

Enter the URL of a public image on any registry, or a private image hosted on a Docker Hub and Google Container Registry. [Learn More](#)

A Deployment will be created to maintain the pods across your cluster. [Learn More](#)

An internal or external service port is specified to map the container listening port. Internal DNS name for the specified service: nginx-test. [Learn More](#)

[SHOW ADVANCED](#)

DEPLOY
 CANCEL

6. Click **DEPLOY** to deploy the pods and service.

You can also click **SHOW ADVANCED OPTIONS** to configure other parameters.

What's next

After you click **DEPLOY**, you can click the left-side navigation pane to view the service or pods that constitute the application.

In the left-side navigation pane, click **Pods**. The icon on the left side of each pod indicates the status of the pod. 🔄 indicates that the pod is being deployed. ✅ indicates that the pod has been successfully deployed. ❗ indicates that an error occurred while deploying the pod.

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
✅ hello-pod	cn-hangzhou.i- b3d4em483agpml3gpm1	Running	0	2018-04-27 17:04:18	0	1.445 Mi
✅ test-mariadb-9bb8f87dd-fjm2m	cn-hangzhou.i- b3d4em483agpml3gpm1	Running	0	2018-04-25 20:50:20	0.002	217.426 Mi
✅ test-wordpress-5b74dcf48c-r8j9h	cn-hangzhou.i- b3d4em483agpml3gpm1	Running	0	2018-04-25 20:50:20	0.005	183.367 Mi
✅ nginx-deployment-basic-6c54bd5869-wq2l5	cn-hangzhou.i- b3d4em483agpml3gpm1	Running	0	2018-04-25 12:11:48	0	1.344 Mi
✅ nginx-deployment-basic-6c54bd5869-krp7	cn-hangzhou.i- b3d4em483agpml3gpm1	Running	0	2018-04-24 18:46:03	0	1.395 Mi

3.4.7.4. Use commands to manage applications

You can use commands to create applications or view application containers.

Prerequisites

Before you use commands on your local host, you have connected to a Kubernetes cluster through kubectl. For more information, see [Connect to a Kubernetes cluster through kubectl](#).

Run a command to create an application

You can use the following command to run a simple container (an NGINX Web server in this example):

```
# kubectl run -it nginx --image=registry.aliyuncs.com/spacexnice/netdia:latest
```

This command creates a service portal for this container. After you specify `--type=LoadBalancer`, an SLB route to the NGINX container is created.

```
# kubectl expose deployment nginx --port=80 --target-port=80 --type=LoadBalancer
```

Run a command to view container information

Run the following command to list all running containers in the default namespace:

```
root@master # kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-2721357637-dvwq3    1/1     Running   1          9h
```

3.4.7.5. Create a service

You can create a service for your application through the Container Service console. The service provides access to the application.

A Kubernetes service, generally known as microservice, is an abstraction which defines a logical set of pods and a policy by which to access the pods. A label selector usually determines whether the set of pods can be accessed by the service.

Each pod has its own IP address. Pods are created and deleted dynamically and quickly. Using pods to provide services externally is therefore not a highly available solution. The service abstraction enables the decoupling between the frontend and the backend. The frontend does not need to be aware of how the backend is implemented, which leads to a loosely coupled microservices based architecture.

For more information, see [Kubernetes service](#).

Prerequisites

You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

Step 1: Create a deployment

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. In the upper-right corner, click **Create from Template**.
3. Select the target cluster and namespace. Set the **Sample Template** field to **Custom** and enter the following code in the **Template** field. Then click **Create**.

Clusters: k8s-cluster

Namespace: default

Sample Template: Resource - basic Deployment

```

1  apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment-basic
5    labels:
6      app: nginx
7  spec:
8    replicas: 2
9    selector:
10   matchLabels:
11     app: nginx
12   template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       containers:
18         - name: nginx
19           image: nginx:1.7.9 # replace it with your exactly
20           <image_name:tags>
21           ports:
22             - containerPort: 80
    
```

Add Deployment

Deploy with Existing Template

The creation process has started. Click here to check the progress: [Kubernetes Dashboard](#)

Save Template Create

In this example, the template of an NGINX deployment is used.

```

apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
          ports:
            - containerPort: 80 ##expose this port in the service
    
```

4. Click **Kubernetes Dashboard** to view the status of the deployment.

Step 2: Create a service

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Ingresses and Load Balancing > Services. The Services page appears.
3. Select the target cluster and namespace. Then click Create in the upper-right corner.
4. In the Create Service dialog box that appears, set the following parameters.

The screenshot shows the 'Create Service' dialog box with the following configuration:

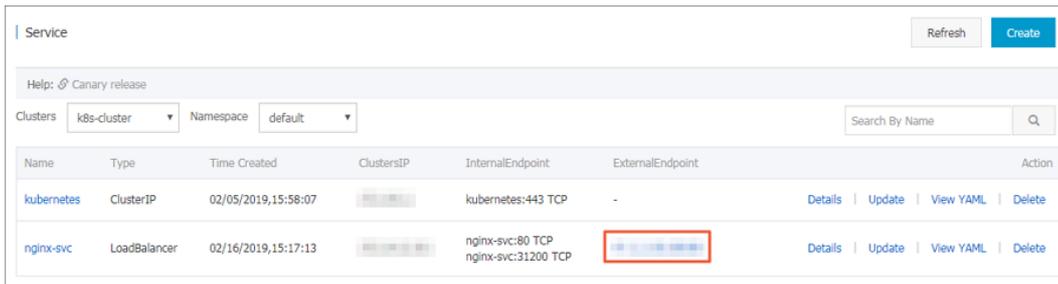
- Name:** nginx-svc
- Type:** Server Load Balancer (dropdown), Public Access (dropdown)
- Backend:** nginx-deployment (dropdown)
- Port Mapping:**
 - + Add
 - Table with columns: Service Port, Container Port, Protocol
 - Row 1: 8080, 8080, TCP (dropdown), -
- Annotations:**
 - + Add SLB Parameters
 - Table with columns: Name, Value
 - Row 1: service.beta.kubernetes.io, 20, -
- Label:**
 - + Add
 - Table with columns: Name, Value
 - Row 1: app, nginx, -

Buttons: Create (blue), Cancel (grey)

- **Name:** The name of the service. In this example, enter nginx-svc.
- **Type:** The type of the service, namely, how to expose the service.
 - **Cluster IP:** Exposes the service through an internal IP address in the cluster. When this option is selected, the service is only accessible within the cluster. This is the default service type.
 - **Node Port:** Exposes the service through the IP address and static port (NodePort) of each node. A NodePort service can route requests to a Cluster IP service, which is automatically created by the system. You can access a Node Port service from outside the cluster by requesting `<NodeIP>:<NodePort>`.
 - **Server Load Balancer:** Exposes the service through an SLB instance, which supports Internet access or internal access. An SLB service can route requests to Node Port and Cluster IP services.
- **Backend:** The backend object that you want to associate with the service. In this example, select nginx-deployment-basic created from the previous step. If you do not specify a deployment, no Endpoint object will be created. You can manually bind the service to an Endpoint object. For more information, see

services-without-selectors.

- **Port Mapping:** Set the service port and container port. The container port must be the same as the one exposed by the backend pod.
 - **Annotations:** Add one or more annotations to the service to configure SLB parameters. For example, set the name to `service.beta.kubernetes.io` and value to `20`. This means that the maximum bandwidth of the service is 20 Mbit/s. For more information, see [Access services by using SLB](#).
 - **Label:** Add labels to the service.
5. Click **Create**. Service `nginx-svc` is displayed on the Services page.
 6. You can view basic information about the service. You can also access its external endpoint through a browser.



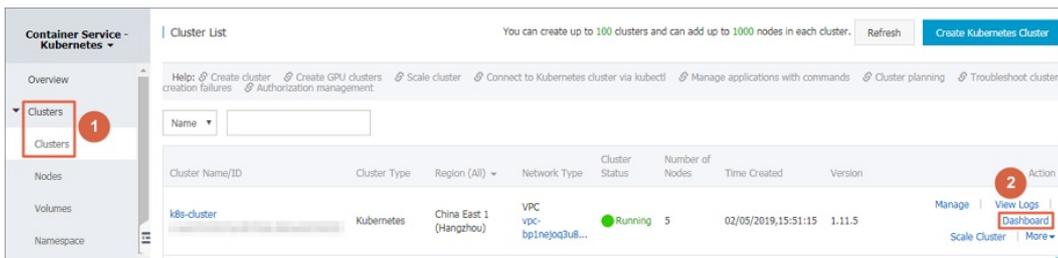
You have created a service and associated it with a backend deployment. You can now visit the NGINX welcome page.

3.4.7.6. Scale a service

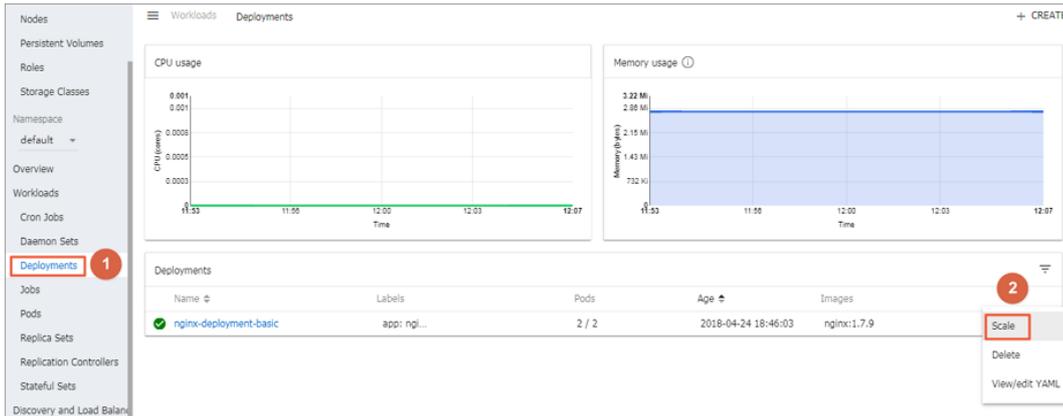
After an application is created, you can scale in or scale out the service based on your needs.

Procedure

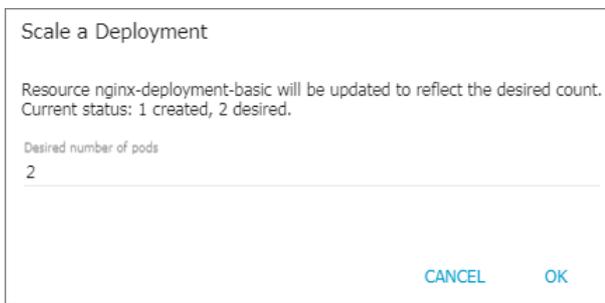
1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Find the target cluster and click **Dashboard** in the Actions column to go to Kubernetes Dashboard.



4. In the left-side navigation pane, click **Deployments**.
5. Find the target application, click the **More** icon on the right, and choose **Scale**.



6. In the dialog box that appears, change the **Desired number of pods** to 2. Then, click **OK**. This action adds a new pod and increases the number of replicas to two.



What's next

The icon on the left side of each Kubernetes object indicates the status of the object.  indicates that the object is being deployed.  indicates that the object has been successfully deployed.

After a deployment is complete, you can click the deployment name to view details of the running web services. You can view the replica sets included in the deployment, and the CPU and memory usage of these replica sets. You can also click  to view container logs.

 **Note** If no resources are displayed, wait a few minutes and then refresh the page.

3.4.7.7. View a service

You can view details about a service through the Container Service console.

Context

If an external service is configured when you create an application, Kubernetes Dashboard creates the external service and preconfigures the SLB instance to direct traffic to containers in the cluster.

Procedure

1. **Log on to the Container Service for Kubernetes console.**
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.
3. Select the target cluster and namespace. Find the target service and click **Details** in the Actions column.
4. (Optional) You can also go to Kubernetes Dashboard. In the left-side navigation pane, click **Services** to view all services.

3.4.7.8. Update a service

You can update a service through the Container Service console.

Update a service through the Services page

1. [Log on to the Container Service for Kubernetes console.](#)
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.
3. Select the target cluster and namespace. Find the target service and click **Update** in the Actions column. In this example, the target service is `nginx-svc`.
4. In the dialog box that appears, modify the configurations based on your needs and click **Update**.

Update Service
✕

Name:

Type: Server Load Balancer Public Access

Port Mapping: + Add

Service Port	Container Port	Protocol	
<input type="text" value="8080"/>	<input type="text" value="8080"/>	TCP	-

Annotations: + Add SLB Parameters

Name	Value	
<input type="text" value="service.beta.kubernetes."/>	<input type="text" value="20"/>	-

Label: + Add

Name	Value	
<input type="text" value="app"/>	<input style="border: 2px solid green;" type="text" value="nginx-v2"/>	-

Update
Cancel

5. Find the service on the Services page and click **Details** in the Actions column to view configuration changes. In this example, the label of the service is updated.

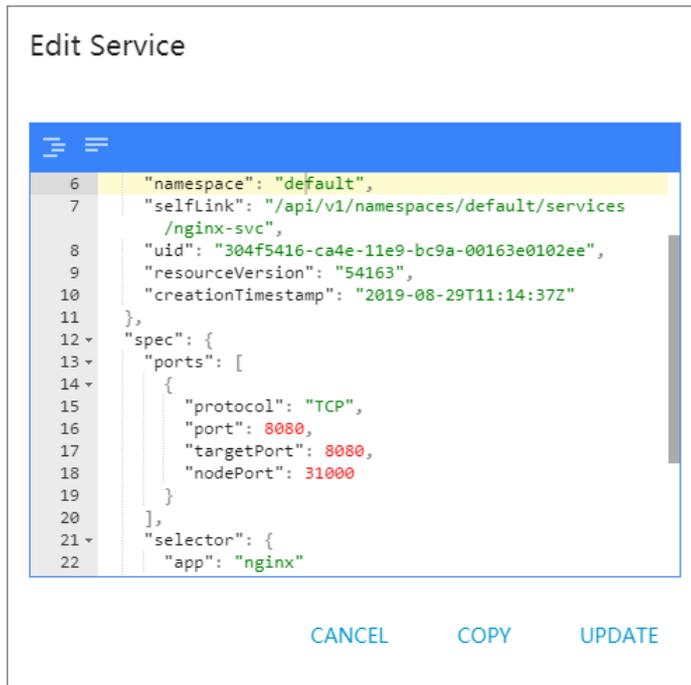
Basic Information	
Name:	nginx-svc
Namespace:	default
Created At:	Aug 29, 2019, 19:44:49 GMT+8
Labels:	app:nginx-v2
Annotations:	service.beta.kubernetes.io:20
Type:	LoadBalancer
ClusterIP:	[Redacted]
InternalEndpoint:	nginx-svc:8080 TCP nginx-svc:31933 TCP
ExternalEndpoint:	[Redacted]:80

Update a service through Kubernetes Dashboard

1. Log on to the Container Service for Kubernetes console.
2. In the left-side navigation pane, choose Clusters. The Clusters page appears.
3. Find the target cluster and click Dashboard in the Actions column.
4. On Kubernetes Dashboard, select the target namespace and click Services in the left-side navigation pane.
5. Find the target service, click the More icon on the right, and choose View/edit YAML.

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age	Actions
nginx-test	k8s-app: nginx-test	[Redacted]	nginx-test:80 TCP nginx-test:30287 TCF	-	08/29/2019, 19:34:27	⋮
nginx-svc	-	[Redacted]	nginx-svc:8080 TCP	[Redacted]	08/29/2019, 19:14:37	Delete
my-service1	app: nginx	[Redacted]	my-service1:30080 T my-service1:32750 T	[Redacted]	08/29/2019, 15:05:19	View/Edit YAML
kubernetes	component: apiser. provider: kubernetc.	[Redacted]	kubernetes:443 TCP	-	08/29/2019, 13:29:29	⋮

6. In the dialog box that appears, modify the configurations. For example, change nodePort to 31000 . Then click UPDATE.



3.4.7.9. Delete a service

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created a service. For more information, see [Create Services](#).

Procedure

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**. The Services page appears.
3. Select the target cluster and namespace. Find the target service and click **Delete** in the Actions column. In this example, the target service is nginx-svc.
4. In the dialog box that appears, click **Confirm**. The target service is deleted and disappears from the Services page.

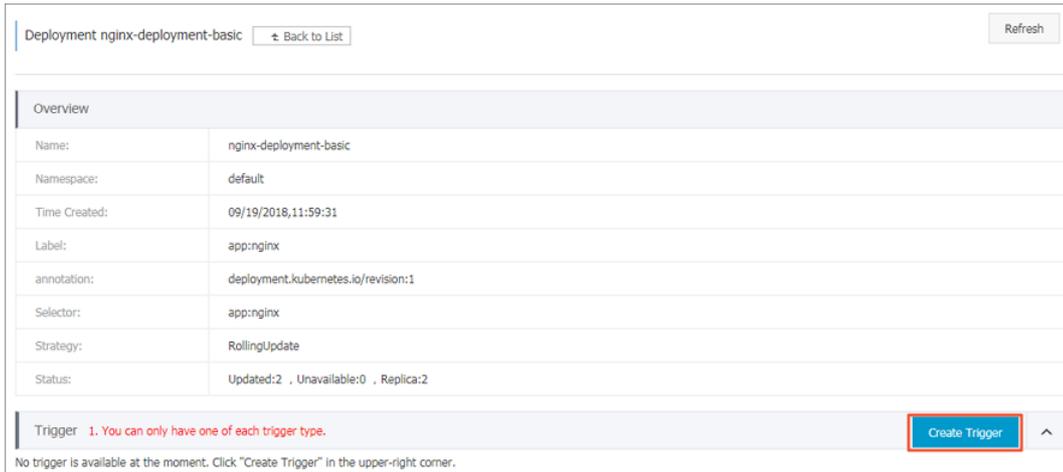
3.4.7.10. Create a trigger on an application

Prerequisites

- You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).
- You have created an application based on which the trigger is created and tested. In this example, an NGINX application is created.

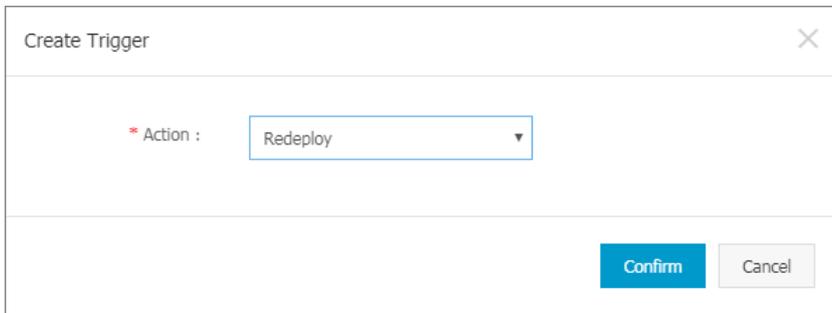
Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. On the Deployments page that appears, select the target cluster and namespace. Then find the NGINX application and click **Details** in the Actions column.
3. On the application details page that appears, click **Create Trigger** in the Trigger section.

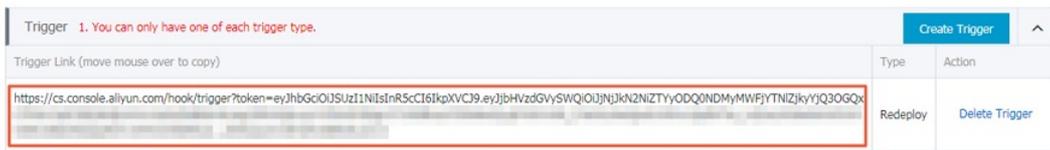


4. In the dialog box that appears, set Action to Redeploy and click OK.

Note Currently, you can only create a trigger to redeploy the application.



After the trigger is created, a trigger link address is displayed in the Trigger section on the Deployment nginx page.



5. Copy the link and open it in your browser. A message appears, displaying information such as the request ID.



6. Go to the Deployment nginx page. A new pod is displayed on the Pods tab.



After the new pod is successfully deployed, the old pod will be automatically deleted.

What's next

You can call triggers through GET or POST requests from a third-party system. For example, you can run curl commands to call triggers.

To call the redeploy trigger, run the following command:

```
curl https://cs.console.aliyun.com/hook/trigger?token=xxxxxxx
```

3.4.7.11. View pods

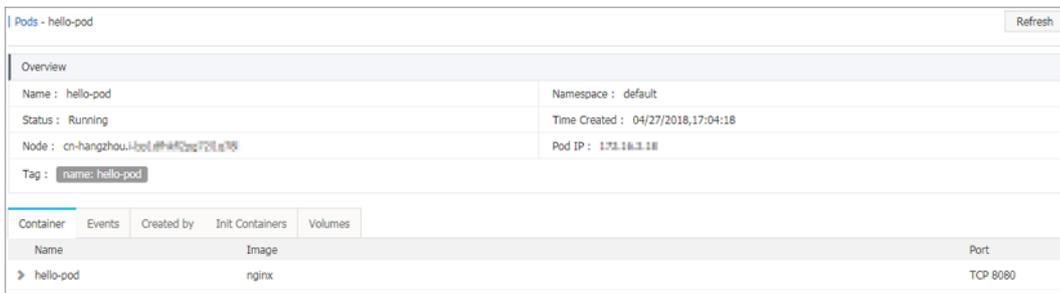
You can view pods through the console.

View pods on the Pods page

1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Applications > Pods. The Pods page appears.
3. Select the target cluster and namespace. Find the target pod and click View Details.

Note You can update or delete pods on the Pods page. We recommend that you use deployments to manage pods if they were created by deployments.

4. On the pod details page, you can view detailed information about the pod.



View pods through Kubernetes Dashboard

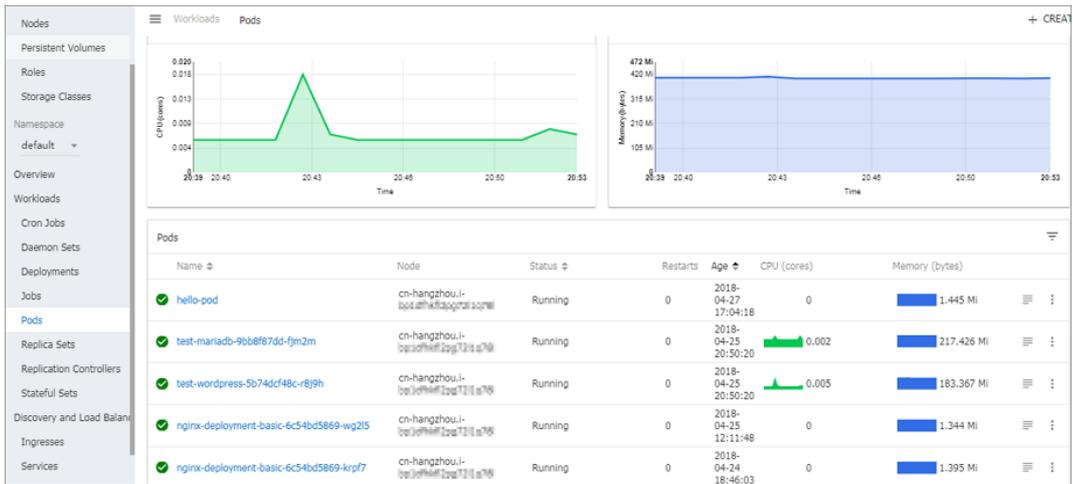
1. Log on to the Container Service console.
2. In the left-side navigation pane, choose Clusters > Clusters. The Clusters page appears.
3. Find the target cluster and click Dashboard in the Actions column. The Kubernetes Dashboard page appears.
4. In the left-side navigation pane, click Pods to view pods of the cluster.

You can also click Services in the left-side navigation pane and then click a service name to view pods of the service.



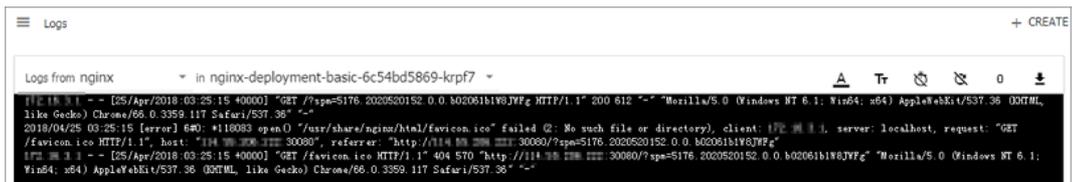
5. The icon at the left side of each pod indicates the status of the pod. 🔄 indicates that the pod is being

deployed.  indicates that the pod has been successfully deployed.

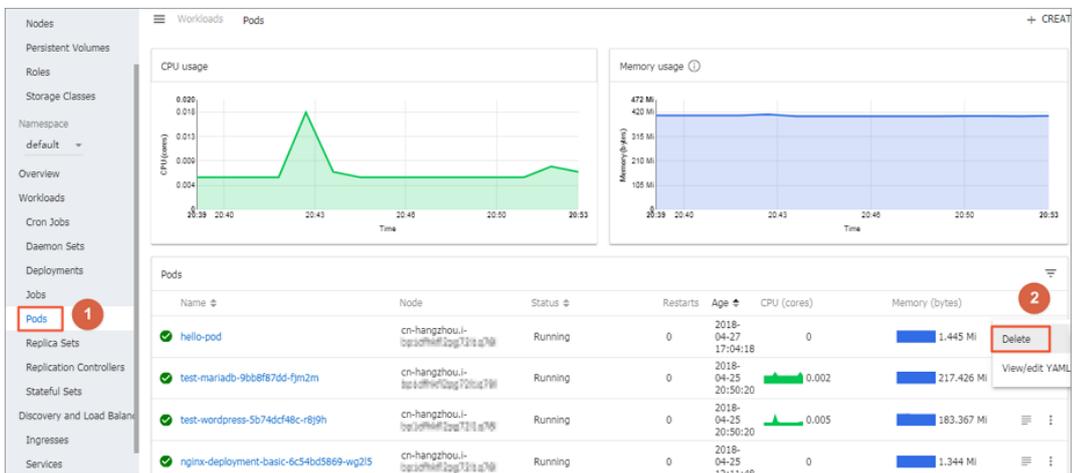


6. Select a pod and click its name to view pod details, including the CPU and memory usage.

7. Select a pod and click  at the right to view logs.



8. You can also click the More icon and click Delete to delete the pod.



3.4.7.12. Schedule pods to nodes

You can add labels to nodes and then configure `nodeSelector` to schedule pods to specific nodes. For more information about how `nodeSelector` works, see [nodeSelector](#).

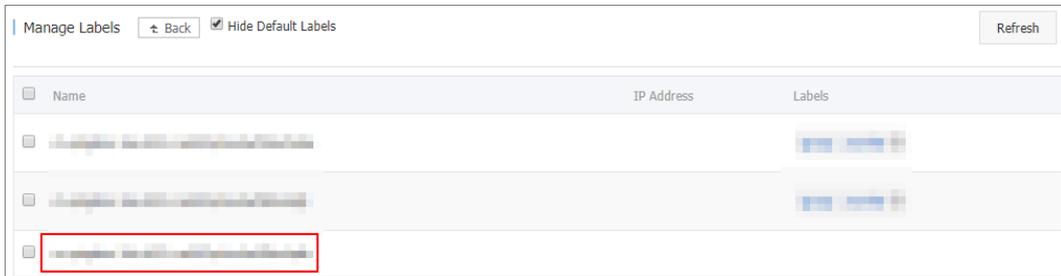
To meet business needs, you may need to deploy a management service on a master node, or deploy certain services on nodes with SSD storage. You can use the following method to schedule pods to specific nodes based on needs.

Prerequisites

You have created a Kubernetes cluster. For more information, see [Create a Kubernetes cluster](#).

Step 1: Add a label to a node

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Clusters > Nodes**. The Nodes page appears.
3. Select the target cluster and click **Manage Labels** in the upper-right corner.
4. Select one or more nodes and then click **Add Label**. In this example, select a worker node.



5. In the dialog box that appears, enter the label name and value, and then click **OK**.

On the Manage Labels page, you can find the `group:worker` label next to the selected node.

You can also use the following command to add a label to a node: `kubectl label nodes <node-name> <label-key>=<label-value> .`

Step 2: Schedule a pod to the node

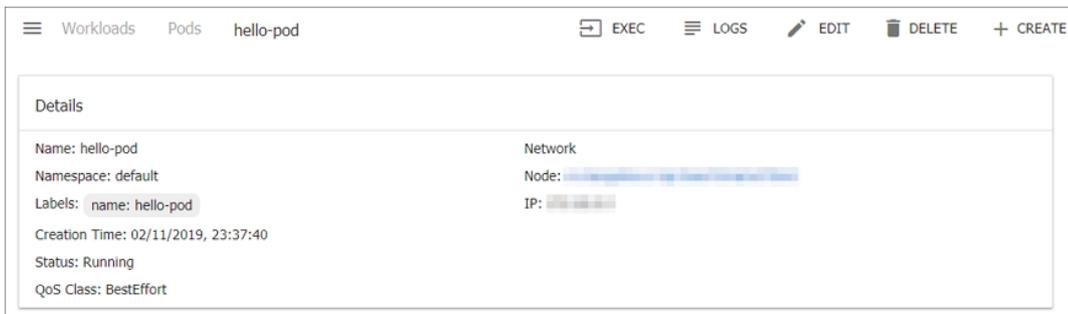
1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. The Deployments page appears.
3. In the upper-right corner, click **Create from Template**.
4. Configure the template to create a pod and schedule it to the node from step 1. Then, click **Create**.
 - **Cluster:** Select the cluster where the pod is deployed.
 - **Namespace:** Select the namespace where the pod belongs. In this example, select the default namespace.
 - **Sample Template:** In this example, select Custom.

Enter the following template content:

```

apiVersion: v1
kind: Pod
metadata:
  labels:
    name: hello-pod
  name: hello-pod
spec:
  containers:
  - image: nginx
    imagePullPolicy: IfNotPresent
    name: hello-pod
    ports:
    - containerPort: 8080
      protocol: TCP
  resources: {}
  securityContext:
    capabilities: {}
    privileged: false
    terminationMessagePath: /dev/termination-log
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  nodeSelector:
    group: worker ##This value must be the same as the label of the node from step 1.
  status: {}
    
```

5. Click **Create**. A message appears indicating the deployment status. After the pod is created, click **Kubernetes Dashboard** to check the status of hello-pod.



3.4.7.13. Simplify Kubernetes application deployment by using Helm

This topic introduces the basic concepts and components of Helm and describes how to use Helm to deploy the sample applications WordPress and Spark in a Container Service for Kubernetes cluster.

Prerequisites

- A Kubernetes cluster is created in the Container Service for Kubernetes console. For more information, see [Create a Kubernetes cluster](#).

Tiller is automatically deployed to the cluster when the Kubernetes cluster is created. The Helm command-line interface (CLI) is automatically installed on each master node. You must configure the Helm CLI to point to the Alibaba Cloud chart repository.

- The supported Kubernetes version is used.

Only Kubernetes 1.8.4 and later versions are supported. For Kubernetes 1.8.1, you can upgrade the cluster to the required version. To upgrade the cluster, log on to the Container Service for Kubernetes console, go to the Clusters page, find the cluster, and then choose **More > Upgrade Cluster** in the Actions column for the cluster.

Context

When you run and manage applications with Kubernetes, you can use Helm as the package manager to simplify application distribution and deployment. The Helm project enables consistent software packaging and supports version control. In the Container Service for Kubernetes console, the App Catalog feature integrates the Helm binaries and supports the Alibaba Cloud chart repository. This allows you to easily deploy applications by using the Helm CLI or in the Container Service for Kubernetes console.

Overview

Helm is an open source tool that is created by Deis. It can be used to simplify the deployment and management of Kubernetes applications.

Helm works as a Kubernetes package manager and allows you to discover, share, and run applications that are created in Kubernetes. When you use Helm, you must understand the following basic concepts:

- **Chart:** a packaging format used by Helm. Each chart contains the images, dependencies, and resource definitions that are required for running an application. A chart may contain service definitions in a Kubernetes cluster. You can use a chart in a similar way as you use a Homebrew formula, the dpkg packages manager of the Advanced Package Tool (APT) package management system, or the Red Hat Package Manager (RPM) package for Yellowdog Updater, Modified (YUM).
- **Release:** an instance of a chart that runs in a Kubernetes cluster. A chart can be installed many times into the same cluster. After a chart is installed, a new release is created. For example, you can install a MySQL chart. If you want to run two databases in your cluster, you can install the MySQL chart twice. Each installation generates a release with a release name.
- **Repository:** the location where charts are stored and released.

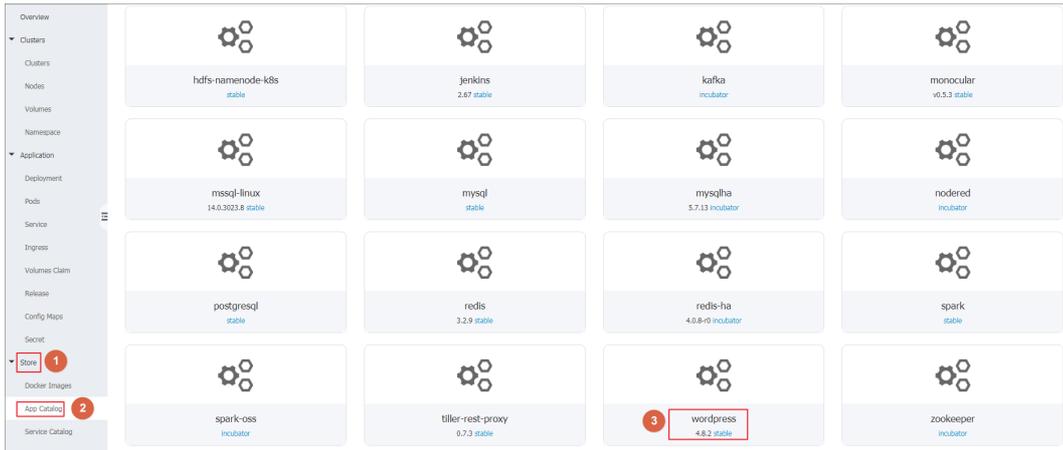
Helm components

Helm works in a client-server architecture and consists of the following components:

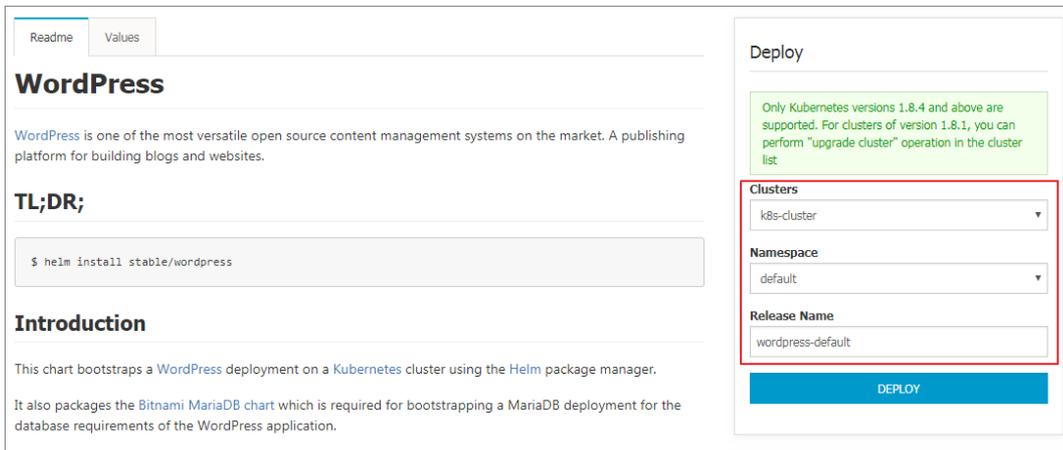
- The Helm CLI is the Helm client that runs on your on-premises computer or on the master nodes of a Kubernetes cluster.
- Tiller is the server-side component and runs on a Kubernetes cluster. Tiller manages the lifecycles of Kubernetes applications.
- A repository is used to store charts. The Helm client can access the index file and packaged charts in a chart repository over HTTP.

Deploy an application in the Container Service for Kubernetes console

1. [Log on to the Container Service console](#)
2. In the left-side navigation pane, choose **Marketplace > App Catalog** to go to the App Catalog page.
3. Click a chart, for example, **WordPress**, to go to the page that shows the details of the chart.

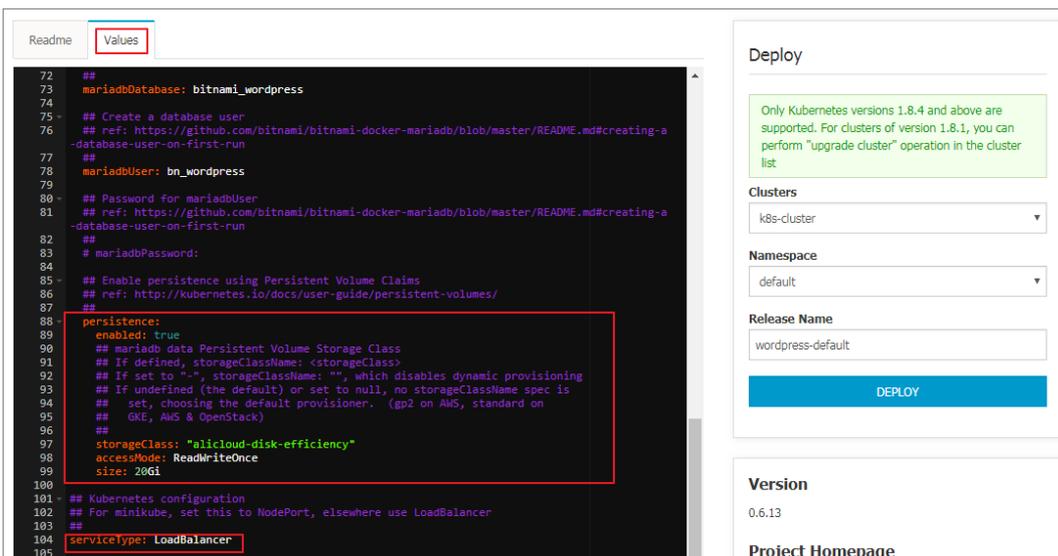


4. In the Deploy section on the right of the page, enter the basic information for the deployment.
 - **Cluster:** Select the cluster to which you want to deploy the application.
 - **Namespace:** Select a namespace for the application. By default, this parameter is set to default.
 - **Release Name:** Enter a release name for the application.

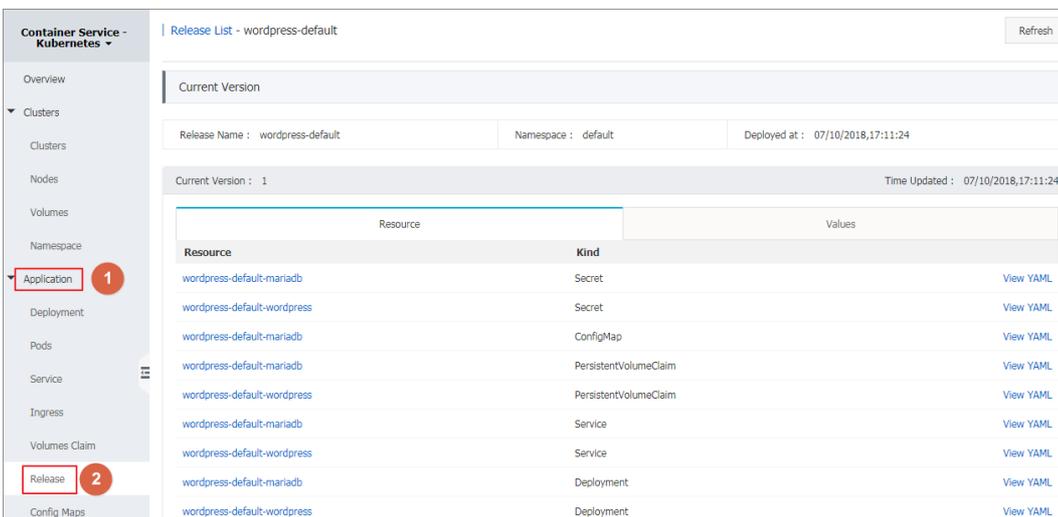


5. Click the **Parameters** tab to set the parameters. In this example, a dynamically provisioned volume is associated with a persistent volume claim (PVC). For more information, see [Use Apsara Stack disks](#).

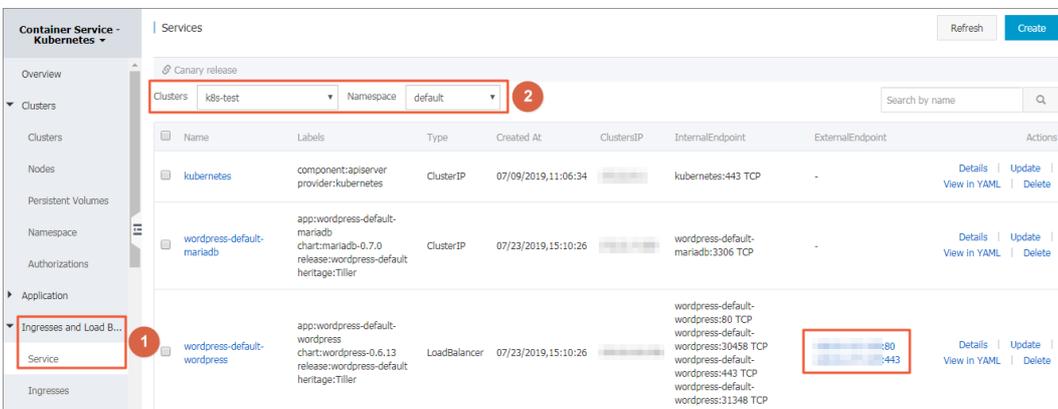
Note Before the association, you must create a persistent volume (PV) as the dynamic volume. The capacity of the PV cannot be less than the value defined by the PVC.



6. After you set the parameters, click Create to deploy the application. After the application is deployed, you are navigated to the release page of the application.



7. In the left-side navigation pane, choose Ingresses and Load Balancing > Services, select the target cluster and namespace, and then find the required service from the list of services. In the External Endpoint column for the service, you can see the external HTTP and HTTPS endpoints.



8. Click either of the endpoints to go to the WordPress application where you can publish blog posts.

Deploy an application by using the Helm CLI

After the Helm CLI is automatically installed on the master node of the Kubernetes cluster and points to the required repository, you can log on to the master node by using SSH. This allows you to deploy applications by using the Helm CLI. For more information, see [Connect to a master node through SSH](#). You can also install and configure the Helm CLI and kubectl on your on-premises computer.

In this example, on your on-premises computer, the Helm CLI and kubectl are installed and configured and the WordPress and Spark applications are deployed.

1. Install and configure the Helm CLI and kubectl.

- i. Install and configure kubectl on your on-premises computer. For more information, see [Connect to a Kubernetes cluster through kubectl](#).

To view the information of the target Kubernetes cluster, on the command line, enter `kubectl cluster-info`.

- ii. Install Helm on your on-premises computer. For more information, see [Install Helm](#).

2. Deploy the WordPress application.

To deploy a WordPress blog website by using Helm, perform the following steps:

- i. On the command line, run the following command:

```
helm install --name wordpress-test stable/wordpress
```

 **Note** Container Service for Kubernetes allows you to use block storage or disks as dynamically provisioned volumes. Before you deploy the WordPress application, you must create dynamically provisioned volumes based on disks.

The following example shows the output:

```
NAME: wordpress-test
LAST DEPLOYED: Mon Nov 20 19:01:55 2017
NAMESPACE: default
STATUS: DEPLOYED
...
```

- ii. On the command line, run the following commands to view the release and service of WordPress.

```
helm list
kubectl get svc
```

- iii. On the command line, run the following command to view the pod that is associated with the WordPress application. The pod may take a few minutes to change to the running state.

```
kubectl get pod
```

- iv. On the command line, run the following command to obtain the endpoint of the WordPress application.

```
echo http://$(kubectl get svc wordpress-test-wordpress -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
```

You can enter the preceding endpoint in your browser to access the WordPress application.

You can also follow the chart instructions described in the console and run the following commands to obtain the administrator account and password of the WordPress application:

```
echo Username: user
echo Password: $(kubectl get secret --namespace default wordpress-test-wordpress -o jsonpath="{.data.wordpress-password}" | base64 --decode)
```

- v. On the command line, run the following command to delete the WordPress application:

```
helm delete --purge wordpress-test
```

Use a third-party chart repository

You can use the default Alibaba Cloud chart repository. You can also use a third-party chart repository if the third-party chart repository is accessible. On the command line, run the following command to add a third-party chart repository:

```
helm repo add Repository name Repository URL
helm repo update
```

For more information about Helm commands, see [Helm documentation](#).

References

The Kubernetes community has experienced rapid technological developments based on Helm. These developments have allowed software providers, such as Bitnami, to offer high-quality charts. For more information about available charts, visit <https://kubernetes.io/docs/concepts/cluster-administration/manage-deployments/#charts>.

3.4.8. SLB and Ingress

3.4.8.1. Overview

Container Service allows you to flexibly manage load balancing and customize load balancing policies for Kubernetes clusters. Kubernetes clusters provide you with a variety of methods to access containerized applications. They also allow you to use SLB or Ingress to access internal services and implement load balancing.

3.4.8.2. Access a service through SLB

You can access a service through Server Load Balancer (SLB).

Use the CLI

1. Create an NGINX application by using the CLI.

```
root@master # kubectl run nginx --image=registry.aliyuncs.com/acs/netdia:latest
root@master # kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
nginx-2721357637-dvwq3              1/1     Running  1          6s
```

2. Create a service of the Server Load Balancer type for the NGINX application. Set `type=LoadBalancer` to expose the NGINX application to the Internet.

```
root@master # kubectl expose deployment nginx --port=80 --target-port=80 --type=LoadBalancer
root@master # kubectl get svc
NAME            CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
nginx          172.19.XX.XX 101.37.XX.XX  80:31891/TCP    4s
```

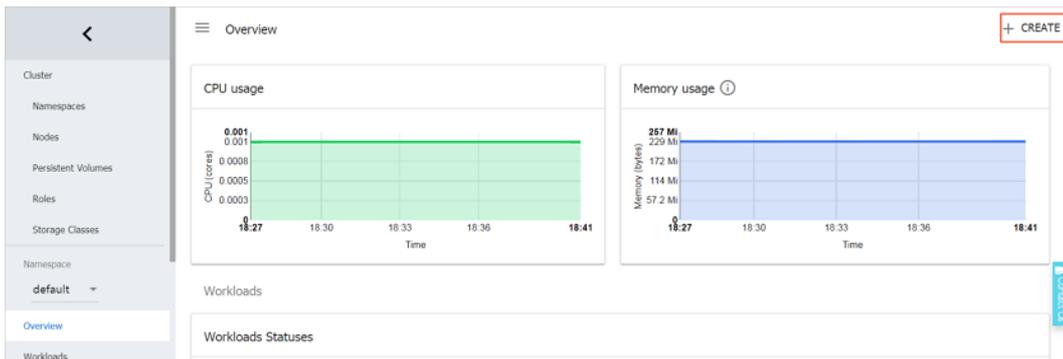
3. Visit `http://101.37.XX.XX` in a browser to access the NGINX application.

Use the Kubernetes Dashboard

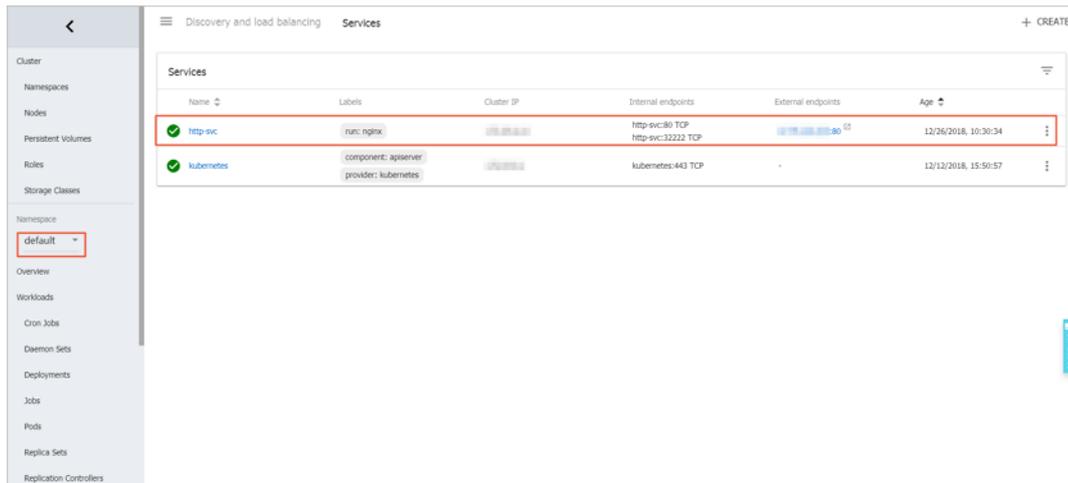
1. Create an `nginx-svc.yml` file with the following code on a local computer.

```
apiVersion: v1
kind: Service
metadata:
  labels:
    run: nginx
  name: http-svc
  namespace: default
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    run: nginx
  type: LoadBalancer
```

2. Log on to the Container Service console.
3. Select the target cluster and click Dashboard in the Actions column to go to the Kubernetes Dashboard.
4. Click Create in the upper-right corner to create an application.



5. Click the **CREATE FROM FILE** tab. Select file *nginx-svc.yml*.
6. Click **UPLOAD**.
This creates an `http-svc` service of the Server Load Balancer type and associates the service with the NGINX application.
7. In the left-side navigation pane, select the default namespace and click **Services**.
You can find the newly created service `http-svc` and its external endpoint `http://114.55.XX.XX:80`.



8. Enter the endpoint in a browser to access the service.

See also

Server Load Balancer provides a variety of parameters that you can use to configure settings such as health check, billing method, and SLB instance type. For more information, see [SLB parameters](#).

Annotations

You can add annotations to use the features provided by Server Load Balancer.

Use an existing internal SLB instance

You need to add two annotations as follows. Note that you must replace "yourloadbalancer-id" with your SLB instance ID.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    service.beta.kubernetes.io/alibaba-cloud-loadbalancer-address-type: intranet
    service.beta.kubernetes.io/alibaba-cloud-loadbalancer-id: your-loadbalancer-id
  labels:
    run: nginx
  name: nginx
  namespace: default
spec:
  ports:
    - name: web
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    run: nginx
  sessionAffinity: None
  type: LoadBalancer
```

Save the preceding code as an `slb.svc` file and run the following command: `kubectl apply -f slb.svc`.

Create an HTTPS-based service of the Server Load Balancer type

Create a certificate in the Apsara Stack console and copy cert-id. Then create an HTTPS-based service of the Server Load Balancer type as follows:

```

apiVersion: v1
kind: Service
metadata:
  annotations:
    service.beta.kubernetes.io/alibaba-cloud-loadbalancer-cert-id: your-cert-id
    service.beta.kubernetes.io/alibaba-cloud-loadbalancer-protocol-port: "https:443"
  labels:
    run: nginx
  name: nginx
  namespace: default
spec:
  ports:
  - name: web
    port: 443
    protocol: TCP
    targetPort: 443
  selector:
    run: nginx
  sessionAffinity: None
  type: LoadBalancer
    
```

 **Note** Annotations are case sensitive.

SLB parameters

Annotation	Description	Default value
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-protocol-port	The listening port. Separate multiple ports with commas (.). For example, https:443,http:80.	-
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-address-type	The type of the SLB instance. Valid values: internet and intranet.	internet
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-slb-network-type	The network type of the SLB instance. Valid values: classic and vpc.	classic
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-charge-type	The billing method of the SLB instance. Valid values: paybytraffic and paybybandwidth.	paybybandwidth
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-id	The ID of the SLB instance. You can use loadbalancer-id to specify an existing SLB instance and its existing listeners will be overwritten. The SLB instance will not be deleted if the service is deleted.	-
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-backend-label	The labels for specifying the nodes to be added as backend servers of the SLB instance.	-

Annotation	Description	Default value
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-region</code>	The region where the SLB instance is deployed.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-bandwidth</code>	The bandwidth of the SLB instance.	50
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-cert-id</code>	The certificate ID. You need to upload the certificate first.	""
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-flag</code>	Specifies whether to enable the health check feature. Valid values: on and off.	off. This annotation is not required for TCP listeners. By default, the health check feature is enabled for TCP listeners and cannot be disabled.
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-type</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-uri</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-connect-port</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-healthy-threshold</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-unhealthy-threshold</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-interval</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-connect-timeout</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-
<code>service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-timeout</code>	For more information, see the <i>CreateLoadBalancerTCPListener</i> chapter of SLB Developer Guide.	-

3.4.8.3. Configure ingress monitoring

You can enable the default VTS module to view ingress monitoring data.

Enable VTS module by using the CLI

1. Modify the ingress ConfigMap to add the following configuration item: `enable-vts-status: "true"` .

```
root@master # kubectl edit configmap nginx-configuration -n kube-system
configmap "nginx-configuration" edited
```

The modified ingress ConfigMap is as follows:

```
apiVersion: v1
data:
  enable-vts-status: "true" # Enable the VTS module
  proxy-body-size: 20m
kind: ConfigMap
metadata:
  annotations:
    kubectL.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"proxy-body-size":"20m"},"kind":"ConfigMap","metadata":{"annotations":{},"labels":{"app":"ingress-nginx"},"name":"nginx-configuration","namespace":"kube-system"}}
  creationTimestamp: 2018-03-20T07:10:18Z
  labels:
    app: ingress-nginx
    name: nginx-configuration
    namespace: kube-system
  selfLink: /api/v1/namespaces/kube-system/configmaps/nginx-configuration
```

2. Verify that the VTS module is enabled.

```
root@master # kubectl get pods --selector=app=ingress-nginx -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
nginx-ingress-controller-79877595c8-78gq8  1/1     Running  0          1h
root@master # kubectl exec -it nginx-ingress-controller-79877595c8-78gq8 -n kube-system -- cat /etc/nginx/nginx.conf | grep vhost_traffic_status_display
vhost_traffic_status_display;
vhost_traffic_status_display_format html;
```

3. Access the NGINX Ingress Controller from a local computer.

 **Note** By default, the VTS port is not enabled for security concerns. The following example uses port forwarding to access the controller.

```
root@master # kubectl port-forward nginx-ingress-controller-79877595c8-78gq8 -n kube-system 18080
Forwarding from 127.0.0.1:18080 -> 18080
Handling connection for 18080
```

4. Visit `http://localhost:18080/nginx_status` to access the NGINX Ingress Controller.

Nginx Vhost Traffic Status

Server main

Host	Version	Uptime	Connections				Requests			Shared memory				
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode
nginx-ingress-controller-79877595c8-78gq8	1.13.7	32m 41s	7	0	1	6	93566	93566	1428	1	vhost_traffic_status	10.0 MiB	2.4 KiB	1

Server zones

Zone	Requests			Responses				Traffic				Cache										
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
-	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0
*	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0

Upstreams

upstream-default-backend

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Traffic									
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s					
172.16.3.6:8080	up	0ms	1	0	0	0	0	0ms	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

update interval: 1 sec

[JSON](#) | [GITHUB](#)

Enable VTS module by using the Kubernetes Dashboard

1. Log on to the Container Service console.
2. Find the target cluster and click Dashboard in the Actions column to go to Kubernetes Dashboard.
3. In the left-side navigation pane, select the kube-system namespace and edit the nginx-configuration ConfigMap to add the following configuration item: `enable-vts-status: "true"`.

The modified ingress ConfigMap is as follows:

```
{
  "kind": "ConfigMap",
  "apiVersion": "v1",
  "metadata": {
    "name": "nginx-configuration",
    "namespace": "kube-system",
    "selfLink": "/api/v1/namespaces/kube-system/configmaps/nginx-configuration",
    "creationTimestamp": "2018-03-20T07:10:18Z",
    "labels": {
      "app": "ingress-nginx"
    },
    "annotations": {
      "kubectl.kubernetes.io/last-applied-configuration": "{\"apiVersion\":\"v1\",\"data\":{\"proxy-body-size\":\"20m\"},\"kind\":\"ConfigMap\",\"metadata\":{\"annotations\":{},\"labels\":{\"app\":\"ingress-nginx\"},\"name\":\"nginx-configuration\",\"namespace\":\"kube-system\"}}\n"
    }
  },
  "data": {
    "proxy-body-size": "20m",
    "enable-vts-status": "true"
  }
}
```

4. Access the NGINX Ingress Controller from a local computer.


```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME      HOSTS      ADDRESS      PORTS      AGE
simple    *          101.37.192.211 80         11s
```

You can now visit `http://101.37.192.211/svc` to access the NGINX service.

Simple fanout based on domains

If you have multiple services exposed externally through different domains, you can use the following configuration to implement a simple fanout based on domains:

```

root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
  - host: foo.example.com
    http:
      paths:
      - path: /film
        backend:
          serviceName: http-svc3
          servicePort: 80
EOF

```

```

root@master # kubectl get ing
NAME          HOSTS          ADDRESS          PORTS    AGE
simple-fanout *      101.37.192.211 80              11s

```

You can now visit `http://foo.bar.com/foo` to access service `http-svc1`, visit `http://foo.bar.com/bar` to access service `http-svc2`, and visit `http://foo.example.com/film` to access service `http-svc3`.

Note

- In a production environment, you need to point the domain to the returned address `101.37.192.211`.
- In a test environment, you need to add the following mapping rules to the `hosts` file.

```

101.37.192.211 foo.bar.com
101.37.192.211 foo.example.com

```

Default domain for simple routing

If you have no domain address, you can use the default domain associated with the ingress to access the service. The default domain is in the following format: `*.[cluster-id].[region-id].alicontainer.com`. You can obtain the domain address on the cluster basic information page in the console.

You can use the following configuration to expose the two services through the default domain.

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: shared-dns
spec:
  rules:
  - host: foo.[cluster-id].[region-id].alicontainer.com ## Replace with the default domain of your cluster
    http:
      paths:
      - path: /
        backend:
          serviceName: http-svc1
          servicePort: 80
  - host: bar.[cluster-id].[region-id].alicontainer.com ## Replace with the default domain of your cluster
    http:
      paths:
      - path: /
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME          HOSTS          ADDRESS          PORTS    AGE
shared-dns    foo.[cluster-id].[region-id].alicontainer.com,bar.[cluster-id].[region-id].alicontainer.com    47.95.160.17
1 80 40m
```

You can now visit `http://foo.[cluster-id].[region-id].alicontainer.com/` to access service `http-svc1` and visit `http://bar.[cluster-id].[region-id].alicontainer.com` to access service `http-svc2`.

Secure routing

Container Service supports managing multiple certificates to enhance protection for your services.

1. Prepare your certificate.

If you have no certificate, use the following method to generate a test certificate.

 **Note** The domain must be the same as the one specified in your ingress configuration.

```
root@master # openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"
```

After you run the preceding command, a certificate file `tls.crt` and a private key file `tls.key` are generated.

Use the certificate and private key to create a Kubernetes secret named `foo.bar`. You need to reference the secret when you create the ingress.

```
root@master # kubectl create secret tls foo.bar --key tls.key --cert tls.crt
```

2. Create a secure ingress.

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: tls-fanout
spec:
  tls:
  - hosts:
    - foo.bar.com
    secretName: foo.bar
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
```

```
root@master # kubectl get ing
NAME          HOSTS          ADDRESS          PORTS          AGE
tls-fanout    *              101.37.192.211  80             11s
```

3. As described in [Simple fanout based on domains](#), you need to configure the `hosts` file or set a domain to access the `tls` ingress.

You can visit `http://foo.bar.com/foo` to access service `http-svc1` and visit `http://foo.bar.com/bar` to access service `http-svc2`.

You can also access the HTTPS service by using HTTP. By default, the ingress redirects HTTP traffic to the HTTPS address. Traffic to `http://foo.bar.com/foo` is automatically redirected to `https://foo.bar.com/foo`.

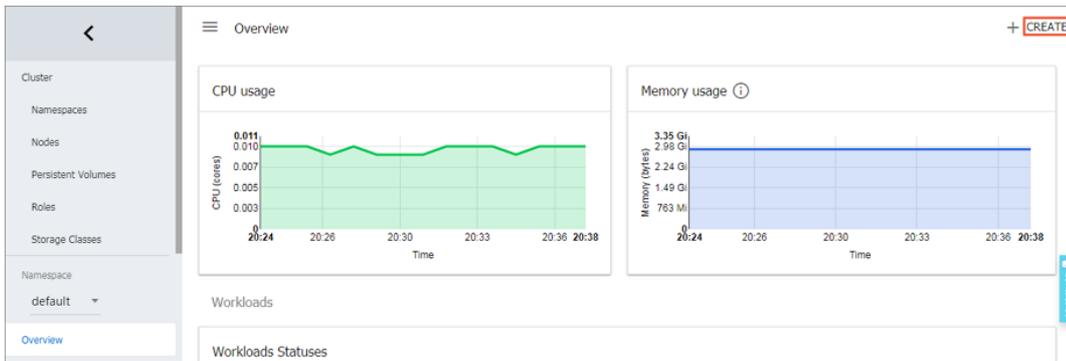
Deploy an ingress in Kubernetes Dashboard

1. Create an `nginx-svc.yml` file with the following code:

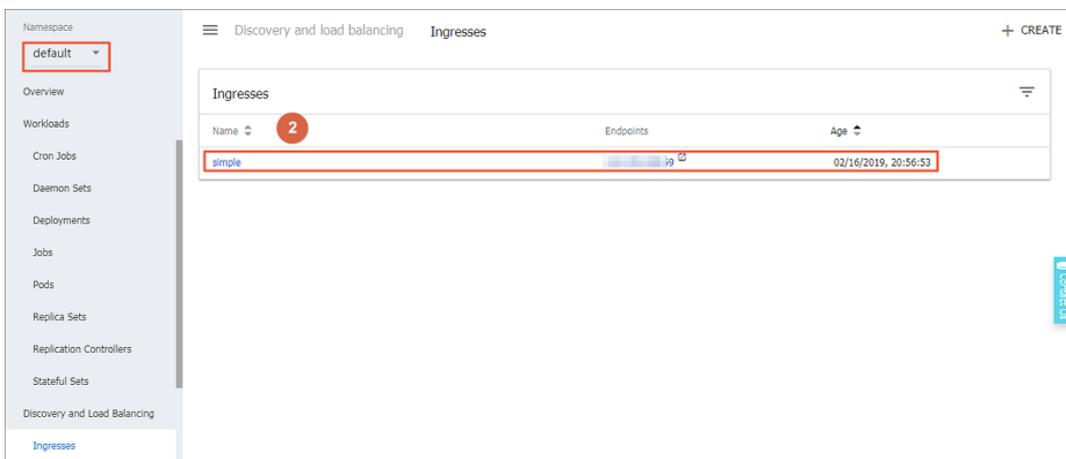
```

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple
spec:
  rules:
  - http:
      paths:
      - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
    
```

2. Log on to the Container Service console.
3. In the left-side navigation pane, click Clusters. Find the target cluster and click Dashboard in the Actions column to go to the Overview page.
4. Click CREATE in the upper-right corner to create an ingress.



5. Click the CREATE FROM FILE tab. Select file `nginx-ingress.yml`.
6. Click UPLOAD.
This creates an ingress that routes Layer-7 traffic for service `http-svc`.
7. In the left-side navigation pane, select the default namespace and click Ingresses.
You can view the newly created ingress and its endpoint.



8. Enter the endpoint into your browser to access the `http-svc` service.

3.4.8.5. Ingress configurations

Container Service provides Ingress controller components. Integrated with Apsara Server Load Balancer, these components provide Kubernetes clusters with flexible and reliable Ingress service.

An Ingress orchestration template is provided below. When you configure an Ingress through the console, you need to configure annotations and may need to create dependencies. For more information, see [Create an ingress through the console](#), [Ingress support](#), and [Kubernetes Ingress](#). You can also create ConfigMaps to configure Ingresses. For more information, see <https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/>.

```

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/service-match: 'new-nginx: header("foo", /^bar$/)' #Canary release rule. In this example, the request header is used.
    Nginx.ingress.kubernetes.io/service-weight: 'New-nginx: 50, old-nginx: 50' #The route weight.
  creationTimestamp: null
  generation: 1
  name: nginx-ingress
  selfLink: /apis/extensions/v1beta1/namespaces/default/ingresses/nginx-ingress
spec:
  rules: ##The Ingress rule.
  - host: foo.bar.com
    http:
      paths:
        - backend:
            serviceName: new-nginx
            servicePort: 80
          path: /
        - backend:
            serviceName: old-nginx
            servicePort: 80
          path: /
  tls:
    ## Enable TLS for secure routing.
  - hosts:
    - *.xxxxxx.cn-hangzhou.alicontainer.com
    - foo.bar.com
    secretName: nginx-ingress-secret ##The Secret name.
status:
  loadBalancer: {}

```

Annotations

For each Ingress, you can configure its annotations, Ingress controller, and rules, such as the route weight, canary release rule, and rewrite rules. For more information about annotations, see <https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/>.

For example, the following rewrite annotation, `nginx.ingress.kubernetes.io/rewrite-target: /`, indicates that `/path` is redirected to the root path `/`, which can be recognized by the backend service.

Rules

Ingress rules are used to manage external access to the services in the cluster and can be HTTP or HTTPS rules. You can configure the following items in rules: domain name (virtual hostname), URL path, service name, and port.

For each rule, you need to set the following parameters:

- **Domain:** The test domain or virtual hostname of your service, such as `foo.bar.com`.
- **Path:** The URL path of your service. Each path is associated with a backend service. Server Load Balancer only forwards traffic to the backend if the incoming request matches the domain and path.
- **Service:** Specify the service in the form of `service:port`. You also need to specify a route weight for each service. The Ingress routes traffic to the matching service based on the route weight.
 - **Name:** The name of the backend service.
 - **Port:** The port of the service.
 - **Weight:** The route weight of the service in the service group.

Note

- a. The weight is a percentage value. For example, you can set two services to the same weight of 50%.
- b. A service group includes services that have the same domain and path defined in the Ingress configuration. If no weight is set for a service, the default value, 100, is used.

Canary release

Container Service supports multiple traffic splitting approaches to suit scenarios such as canary release and A/B testing.

 **Note** Currently, only Ingress controllers of 0.12.0-5 and later versions support traffic splitting.

1. Traffic splitting based on request header
2. Traffic splitting based on cookie
3. Traffic splitting based on query parameter

After canary release is configured, only requests that match certain rules are routed to the corresponding service. If the weight of the corresponding service is lower than 100%, requests that match certain rules are routed to one of the services in the service group based on the weight.

TLS

You can use a Secret that contains a TLS private key and certificate to encrypt the Ingress. This ensures secure routing. The TLS Secret must contain a certificate named `tls.crt` and a private key named `tls.key`. For more information about how TLS works, see [TLS](#). For how to create a Secret, see [Configure a secure Ingress](#).

Labels

You can add labels to the Ingress.

3.4.8.6. Create an Ingress in the console

The Container Service for Kubernetes console is integrated with the Ingress service. You can create an Ingress in the console and manage inbound traffic that is forwarded to different services to meet your business requirements.

Prerequisites

- A Kubernetes cluster is created and an Ingress controller runs normally in the cluster. For more information, see [Create a Kubernetes cluster](#).
- You can use kubectl to connect to a master node. For more information, see [Connect to a Kubernetes cluster through kubectl](#).
- In this example, the image is retrieved over the Internet. You can replace the image address with the one that is accessible to your own cluster. Otherwise, you can build the image that is used in this example and push the image to a repository. To use the image, you can pull the image from the repository.

Step 1: Create a deployment and a service

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. The **Deployments** page appears.
3. Click **Create from Template** in the upper-right corner of the page.
4. Select the target cluster and namespace, select a sample template or enter a custom template, and then click **Create**.

In this example, two NGINX applications are created. One is named old-nginx and the other is named new-nginx.

The following code example shows the template that is used to create old-nginx:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: old-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      run: old-nginx
  template:
    metadata:
      labels:
        run: old-nginx
    spec:
      containers:
        - image: registry.cn-hangzhou.aliyuncs.com/xianlu/old-nginx
          imagePullPolicy: Always
          name: old-nginx
          ports:
            - containerPort: 80
              protocol: TCP
          restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
  name: old-nginx
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    run: old-nginx
  sessionAffinity: None
  type: NodePort
```

The following code example shows the template that is used to create new-nginx:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: new-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      run: new-nginx
  template:
    metadata:
      labels:
        run: new-nginx
    spec:
      containers:
        - image: registry.cn-hangzhou.aliyuncs.com/xianlu/new-nginx
          imagePullPolicy: Always
          name: new-nginx
          ports:
            - containerPort: 80
              protocol: TCP
          restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
  name: new-nginx
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
    run: new-nginx
  sessionAffinity: None
  type: NodePort
```

5. In the left-side navigation pane, choose **Ingresses and Load Balancing > Services**.
On the Services page, find the newly created services.

Step 2: Create an Ingress

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The Ingresses page appears.
3. Select the target cluster and namespace, and click **Create** in the upper-right corner of the page.
4. In the dialog box that appears, enter the Ingress name. In this example, nginx-ingress is used.
5. Configure Ingress rules.

Ingress rules are used to manage inbound access to the services in the cluster. The rules include HTTP or HTTPS rules. You can configure the following items in rules: domain name (virtual hostname), URL path, service name, port, and weight. For more information, see [Ingress configurations](#).

In this example, a rule is added to specify two services for the default domain name and virtual hostname of the cluster. Traffic routing is based on domains.

Rule: + Add

Domain ✖

foo.bar.com

Select *. [obscured] container.com or Custom

path

e.g./

Service + Add

Name	Port	Weight	Percent of Weight
new-nginx	80	100	50.0%
old-nginx	80	100	50.0%

Simple fanout configurations based on domains

In this example, a virtual hostname is used as the test domain to provide services. Route weights are specified for both services and a canary release is configured for one of the services. In your production environment, you can use a domain name that has obtained an Internet Content Provider (ICP) number to provide services.

- **Domain:** Enter the test domain name. In this example, `foo.bar.com` is used.

You must modify the hosts file to add a domain name mapping rule.

```
118.178.XX.XX foo.bar.com #The IP address of the Ingress.
```

- **Services:** Set the path, name, port, and weight of each service.
 - **Path:** Enter the URL path of each service. In this example, the default root path `/` is used.
 - **Name:** In this example, two service names are used: `old-nginx` and `new-nginx`.
 - **Port:** In this example, port `80` is opened.
 - **Weight:** Set the weight for each service. The weight is a percentage value. Default value: `100`. In this example, the same weight `50` is set for each service.

6. Configure TLS. Select **Enable TLS** to enable TLS and configure secure routing. For more information, see [Configure a secure Ingress](#).

- You can select an existing secret.

TLS: Enable Exist secret Create secret

foo.bar

- Log on to a master node and create the `tls.key` and `tls.crt` files.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"
```

b. Create a secret.

```
kubectl create secret tls foo.bar --key tls.key --cert tls.crt
```

c. On the command line, enter `kubectl get secret` and verify that the secret has been created. You can then select the `foo.bar` secret.

o You can also use the TLS private key and certificate to create a secret.

a. Log on to a master node and create the `tls.key` and `tls.crt` files.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"
```

b. On the command line, enter `vim tls.key` and `vim tls.crt` to obtain the private key and certificate that are generated.

c. Copy the certificate to the Cert field and the private key to the Key field.

7. Configure a canary release.

Note Currently, only Ingress controllers of 0.12.0-5 and later versions support traffic splitting.

Container Service for Kubernetes supports multiple traffic splitting methods. This allows you to select more suitable solutions for specific scenarios, such as canary releases and A/B testing, including:

- i. Traffic splitting based on request headers
- ii. Traffic splitting based on cookies
- iii. Traffic splitting based on query parameters

After a canary release is configured, requests that match only the specified rules are routed to the new-nginx service. If the weight of new-nginx is lower than 100%, requests that match the specified rules are routed to this service based on the weight.

In this example, the rule is added to specify a request header that matches the regular expression `foo=^bar$`. Only requests that contain this header can access new-nginx.

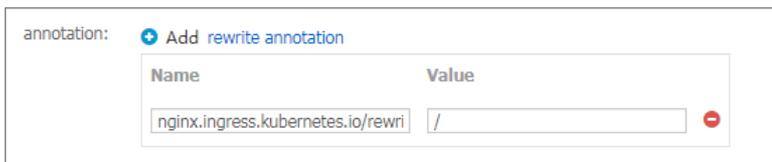
- o **Services:** the service to be accessed.
- o **Type:** the type of the matching rule, such as Header, Cookie, or Query.
- o **Name and Match Value:** custom request fields. The name and matching value comprise a key-value pair.
- o **Matching Rule:** Regular expressions and exact matches are supported.

8. Configure annotations.

Click **Rewrite Annotation** to add a rewrite annotation for the Ingress. For example, `nginx.ingress.kubernetes.io/rewrite-target: /` indicates that `/path` is redirected to the root path `/`. The root path can be recognized by the backend service.

Note In this example, no path is configured for the service. You do not need to configure rewrite annotations. Rewrite annotations allow the Ingress to forward traffic through root paths to the backend service. This avoids the error 404 that is caused by invalid paths.

You can also click **Add** to enter annotation names and values in key-value pairs. For more information about Ingress annotations, visit <https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/>.

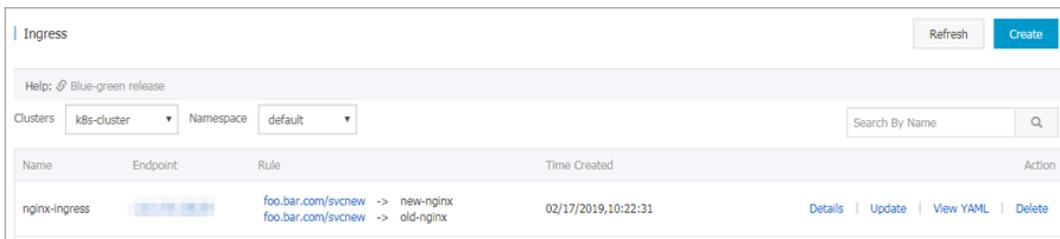


9. Add labels.

Add labels to describe the features of the Ingress.

10. Click **Create**.

You can find the `nginx-ingress` Ingress on the Ingresses page.



11. Click `foo.bar.com` to view the NGINX welcome page.

When you click the domain name that points to `new-nginx`, the `old-nginx` service page appears.

Note By default, when you enter the route address in the browser, requests with headers that do not contain `foo=^bar$` are directed to `old-nginx`.



12. Use SSH to log on to a master node. Run the following commands to simulate requests with specific headers and check the results:

```

curl -H "Host: foo.bar.com" http://47.107.XX.XX
old
curl -H "Host: foo.bar.com" http://47.107.XX.XX
old
curl -H "Host: foo.bar.com" http://47.107.XX.XX          #Similar to a browser request.
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX      #Simulate a request with a specific header. The
results are returned based on the weight.
new
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.XX.XX
new
    
```

3.4.8.7. Update an ingress

You can update an ingress through the console.

Prerequisites

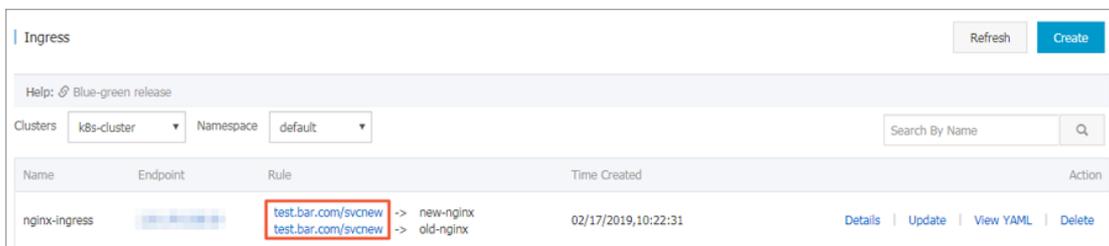
- You have created a Kubernetes cluster and an ingress controller is running normally in the cluster. For more information about creating clusters, see [Create a Kubernetes cluster](#).
- You have created an ingress. For more information, see [Create an ingress through the console](#).

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**. The Ingresses page appears.
3. Select the target cluster and namespace. Find the ingress that you want to update and click **Update** in the Actions column.
4. In the dialog box that appears, modify the parameters and click **OK**. This example changes `foo.bar.com` to `test.bar.com`.

What's next

On the Ingresses page, you can find the updated ingress rule.



3.4.8.8. Delete an ingress

Prerequisites

- You have created a Kubernetes cluster and an ingress controller is running normally in the cluster. For more information about cluster creation, see [Create a Kubernetes cluster](#).
- You have created an ingress. For more information, see [Create an ingress through the console](#).

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Ingresses and Load Balancing > Ingresses**.
3. Select the target cluster and namespace. Find the ingress that you want to delete and click **Delete** in the **Actions** column.
4. In the dialog box that appears, click **OK**.

3.4.9. Config maps and secrets

3.4.9.1. Create a ConfigMap

You can create a ConfigMap on the ConfigMaps page or by using a template.

Create a ConfigMap on the ConfigMaps page

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Configuration > ConfigMaps**. The ConfigMaps page appears.
3. Select the target cluster and namespace, and then click **Create**.
4. Set the parameters and click **OK**.

Create a ConfigMap on the ConfigMaps page

Parameter	Description
Cluster	The ID of the selected cluster.
Namespace	The selected namespace. A ConfigMap is a kind of Kubernetes resource object and must be scoped into a namespace.
ConfigMap Name	Required. The name can contain lowercase letters, digits, hyphens (-), and periods (.). Other resource objects need to reference ConfigMap names to obtain configuration information.
ConfigMap	Enter the Name and Value , and then click Add to add the key-value pair. You can also click Edit YAML file , modify the parameters in the dialog box that appears, and then click OK .

In this example, two variables named `enemies` and `lives` are created. Their values are set to `aliens` and `3` respectively.

* Namespace: default

* Config Map Name: test-config
Name must consist of lowercase alphanumeric characters, '-' or '.'. Name cannot be empty.

Variable Name	Variable Value	Action
enemies	aliens	Edit Delete
lives	3	Edit Delete

Name Value Add

Variable key must be unique. Variable key and value cannot be empty.

Edit YAML file

OK Cancel

5. Click **OK**. You can find the newly created ConfigMap on the ConfigMaps page.

You can also click **Browse** to upload a configuration file to create a ConfigMap.

Create a ConfigMap from a template

1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Applications > Deployments**. The Deployments page appears.
3. In the upper-right corner, click **Create from Template**.
4. On the page that appears, set the parameters and click **Create**.

Create a ConfigMap from a template

Parameter	Description
Cluster	The cluster where the ConfigMap is created.
Namespace	The namespace where the ConfigMap belongs. A ConfigMap is a kind of Kubernetes resource object and must be scoped into a namespace.
Sample Template	Container Service provides various YAML templates for different types of resources. This helps you deploy resources quickly. You can choose <i>Custom</i> and enter your own ConfigMap based on YAML syntax, or select the <i>Resource-ConfigMap</i> template. In the sample template, the ConfigMap is named <i>aliyun-config</i> and contains two variable files <i>game.properties</i> and <i>ui.properties</i> . You can modify the ConfigMap based on your needs.
Template	Enter the template content based on YAML syntax. The template can contain multiple resource objects that are separated by <code>---</code> .
Add Deployment	This feature allows you to quickly define a YAML template. You can click Use Existing Template to import an existing template.

You can find the newly created ConfigMap *aliyun-config* on the ConfigMaps page.

3.4.9.2. Use a ConfigMap in a Pod

You can use a ConfigMap in a Pod in the following scenarios:

- Use a ConfigMap to define environment variables
- Use a ConfigMap to configure command line parameters
- Use a ConfigMap in volumes

For more information, see [Configure a Pod to use a ConfigMap](#).

Limits

To use a ConfigMap in a Pod, make sure that the ConfigMap and Pod are in the same cluster and namespace.

Create a ConfigMap

This example creates a ConfigMap named `special_config`, which consists of two key-value pairs:

```
SPECIAL_LEVEL: very and SPECIAL_TYPE: charm .
```

You can use the following YAML template to create a ConfigMap.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  SPECIAL_LEVEL: very
  SPECIAL_TYPE: charm
```

You can also log on to the Container Service console and choose **Configuration > ConfigMaps** in the left-side navigation pane. You can then click **Create** to create a ConfigMap.

Clusters: [dropdown]

Namespace: default

* ConfigMap Name:

The name must be 1 to 253 characters in length and can contain only lower-case letters numbers hyphens (-) and periods (.).

ConfigMap:

Name	Value
<input type="text" value="SPECIAL_TYPE"/>	<input type="text" value="charm"/>
<input type="text" value="SPECIAL_LEVEL"/>	<input type="text" value="very"/>

A name can contain only numbers letters underscores (_) hyphens (-) and periods (.).

Use ConfigMaps to define Pod environment variables

Define the value of a ConfigMap as an environment variable

You can log on to the Container Service console and choose **Applications > Deployments** in the left-side navigation pane. Click **Create from Template**, select and modify the Pod type template, and deploy the application. You can also go to the Kubernetes dashboard and choose **Upload YAML or JSON File**.

The following sample template creates a Pod and defines environment variables in the Pod. `valueFrom` is used to reference the value of `SPECIAL_LEVEL` to define an environment variable.

```

apiVersion: v1
kind: Pod
metadata:
  name: config-pod-1
spec:
  containers:
  - name: test-container
    image: busybox
    command: [ "/bin/sh", "-c", "env" ]
    env:
    - name: SPECIAL_LEVEL_KEY
      valueFrom:      ##Use valueFrom to denote that env references the value of a ConfigMap.
        configMapKeyRef:
          name: special-config      ##The referenced ConfigMap name.
          key: SPECIAL_LEVEL       ##The referenced ConfigMap key.
    restartPolicy: Never

```

To define the values of multiple ConfigMaps as environment variables, you only need to add multiple env parameters in the Pod definition.

Define the key-value pairs of a ConfigMap as environment variables

To define the key-value pairs of a ConfigMap as Pod environment variables, you can use the envFrom parameter. The keys in a ConfigMap are used as the names of the environment variables.

A sample template is provided as follows:

```

apiVersion: v1
kind: Pod
metadata:
  name: config-pod-2
spec:
  containers:
  - name: test-container
    image: busybox
    command: [ "/bin/sh", "-c", "env" ]
    envFrom:      ##Reference all the key-value pairs in the special-config ConfigMap.
    - configMapRef:
      name: special-config
    restartPolicy: Never

```

Use a ConfigMap to configure command line parameters

You can use ConfigMaps to configure the commands or parameter values in a container by using the environment variable replacement syntax `$(VAR_NAME)`. A sample template is provided as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-3
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "echo ${SPECIAL_LEVEL_KEY} ${SPECIAL_TYPE_KEY}" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: SPECIAL_LEVEL
        - name: SPECIAL_TYPE_KEY
          valueFrom:
            configMapKeyRef:
              name: special-config
              key: SPECIAL_TYPE
      restartPolicy: Never
```

Run the Pod and the output is as follows:

```
very charm
```

Use a ConfigMap in volumes

You can use a ConfigMap to define volumes. The following sample template specifies a ConfigMap name under volumes. This stores the key-value pair data to the mountPath path, which is /etc/config in this example. This generates configuration files that are named after the keys of the ConfigMap. The corresponding values of the ConfigMap are stored in these files.

```

apiVersion: v1
kind: Pod
metadata:
  name: config-pod-4
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/" ] ##List the names of files under this directory.
      volumeMounts:
        - name: config-volume
          mountPath: /etc/config
  volumes:
    - name: config-volume
      configMap:
        name: special-config
  restartPolicy: Never
    
```

Run the Pod and the keys of the ConfigMap are output:

```

SPECIAL_TYPE
SPECIAL_LEVEL
    
```

3.4.9.3. Update a ConfigMap

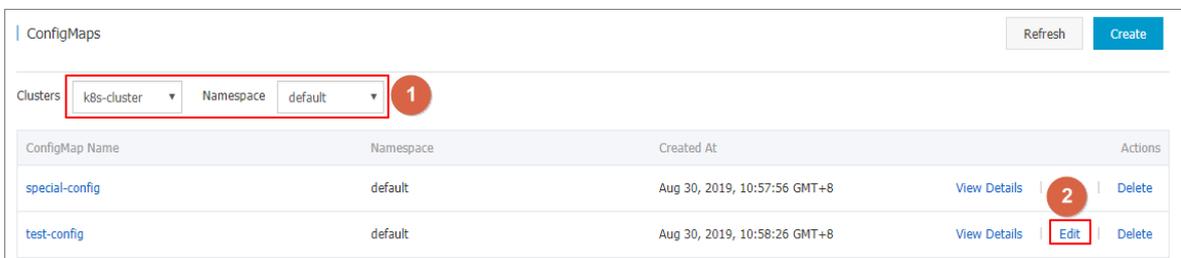
You can use multiple methods to update a ConfigMap.

Note

If you update a ConfigMap, the applications that use this ConfigMap will be affected.

Update a ConfigMap on the ConfigMaps page

1. Log on to the Container Service for Kubernetes console.
2. In the left-side navigation pane, choose Configuration > ConfigMaps to go to the ConfigMaps page.
3. Select the target cluster and namespace, find the ConfigMap that you want to update, and then click Edit in the Actions column for the ConfigMap.



4. In the dialog box that appears, modify the configurations, and click OK.

Update a ConfigMap by using the Kubernetes dashboard

1. Log on to the Container Service for Kubernetes console.
2. In the left-side navigation pane, choose Clusters > Clusters to go to the Clusters page. Find the cluster

that you want to manage and click **Dashboard** in the Actions column for the cluster.

3. On the Overview page, choose **Config and Storage > ConfigMaps** in the left-side navigation pane. Select the target ConfigMap and choose  > **View/edit YAML**.

4. In the dialog box that appears, modify the configurations and click **Update**.

3.4.9.4. Delete a ConfigMap

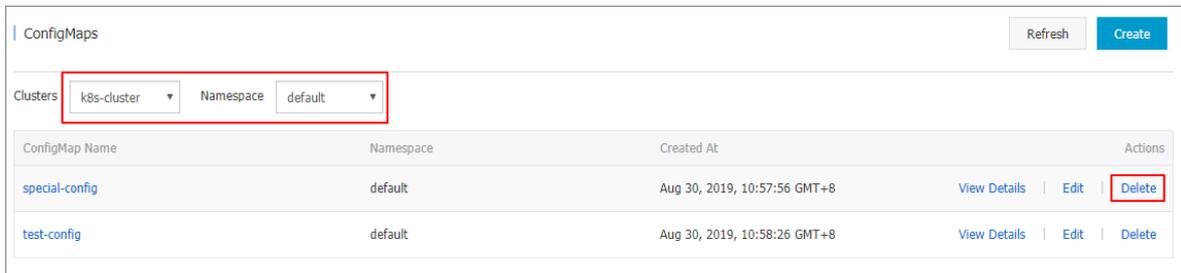
You can use multiple methods to delete a ConfigMap.

Notes

If you delete a ConfigMap, the applications that use this ConfigMap will be affected.

Delete a ConfigMap on the ConfigMaps page

1. **Log on to the Container Service console.**
2. In the left-side navigation pane, choose **Configuration > ConfigMaps**. The ConfigMaps page appears.
3. Select the target cluster and namespace. Find the ConfigMap that you want to delete and click **Delete** in the Actions column.



Delete a ConfigMap through Kubernetes Dashboard

1. **Log on to the Container Service console.**
2. In the left-side navigation pane, choose **Clusters > Clusters**. Find the target cluster and click **Dashboard** in the Actions column.
3. On the Kubernetes Dashboard page, choose **Config and Storage > Config Maps** in the left-side navigation pane. Find the target ConfigMap and choose the **More icon > Delete**.
4. In the dialog box that appears, click **DELETE**.

3.4.9.5. Create a secret

You can create secrets for applications in the Container Service for Kubernetes console.

Prerequisites

A Kubernetes cluster is created.

Context

We recommend that you use secrets to store sensitive information in Kubernetes clusters, such as passwords and certificates.

Secrets are classified into the following types:

- **Service Account:** This type of secret is automatically created by Kubernetes and is automatically mounted to the pod directory `/run/secrets/kubernetes.io/serviceaccount`. You can use this type of secret to access the Kubernetes API.
- **Opaque:** This type of secret is encoded in the Base64 format and used to store sensitive information, such as passwords and certificates.

You can use the Container Service for Kubernetes console to create only Opaque secrets. Opaque data belongs to the map type. The value of this type must be encoded in the Base64 format. You can encode plain text in the Base64 format by using the console.

You can also manually create secrets by using the command-line interface (CLI). For more information, see [Kubernetes secrets](#).

Procedure

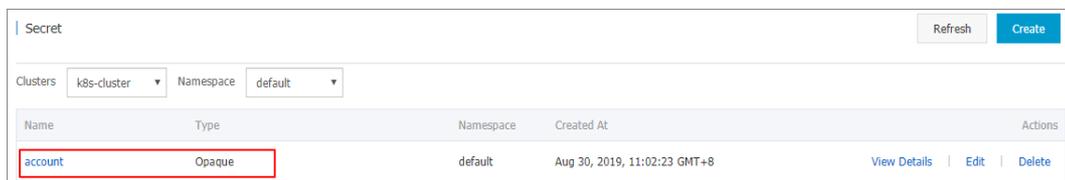
1. [Log on to the Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Configuration > Secrets**. The Secrets page appears.
3. Select the target cluster and namespace. Click **Create** in the upper-right corner of the page.
4. Configure the secret and click **OK**.

 **Note** To enter secret data in plain text, select **Encode Data Values Using Base64**.

Secret parameters

Parameter	Description
Name	The name of the secret that you want to create. The name must be 1 to 253 characters in length and can only contain lowercase letters, digits, hyphens (-), and periods (.).
Data	The data stored in the secret. Click Add , and in the dialog box that appears, enter the key and value as a key-value pair. In this example, two entries are entered: <code>username: admin</code> and <code>password: 1f2d1e2e67df</code> .

5. You can view the newly created secret on the Secrets page.



3.4.9.6. Edit a secret

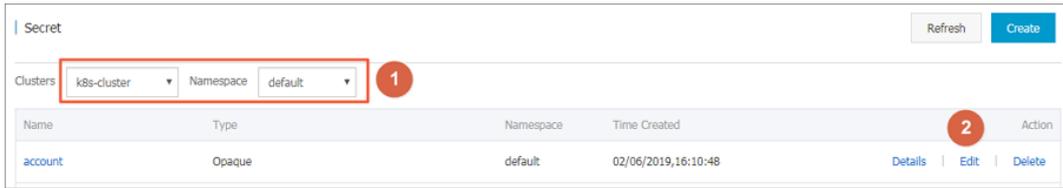
This topic describes how to edit a secret in the Container Service console.

Prerequisites

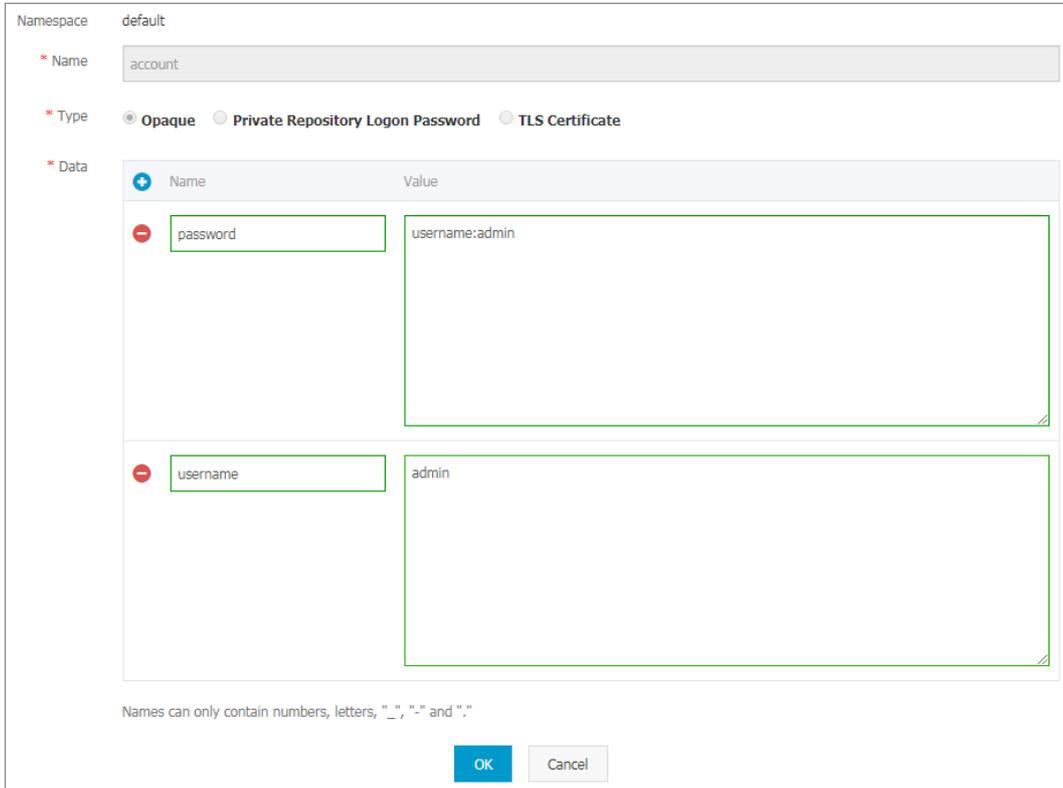
- You have created a Kubernetes cluster.
- You have created a secret. For more information, see [Create a secret](#).

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Configuration > Secrets** to go to the Secrets page.
3. Select the cluster and namespace. Choose the target secret and click **Edit** in the Actions column.



4. On the Edit Secret page, edit the secret based on your needs.



5. Click **OK** to save your edits.

3.4.9.7. Delete a secret

This topic describes how to delete a secret in the Container Service console.

Prerequisites

- You have created a Kubernetes cluster.
- You have created a secret. For more information, see [Create a secret](#).

Context

Note Do not delete secrets that were generated during the cluster creation process.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Configuration > Secrets** to go to the Secrets page.
3. Select the cluster and namespace. Choose the target secret and click **Delete** in the Actions column.
4. In the dialog box that appears, click **OK** to delete the secret.

3.4.10. Templates

3.4.10.1. Create an orchestration template

This topic describes how to use multiple methods to create orchestration templates through the Container Service console.

Procedure

1. Log on to the [Container Service console](#).
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates** and click **Create** in the upper-right corner.
3. In the dialog box that appears, configure the template, and then click **Save**. This example demonstrates how to create a Tomcat application template that contains a deployment and a service.
 - **Name:** The name of the template.
 - **Description:** Optional. The description of the template.
 - **Template:** Enter the template content based on YAML syntax. The template can contain multiple resource objects that are separated by `---`.

Create

Name:
The name should be 1-64 characters long, and can contain numbers, English letters, Chinese characters and hyphens.

Description:

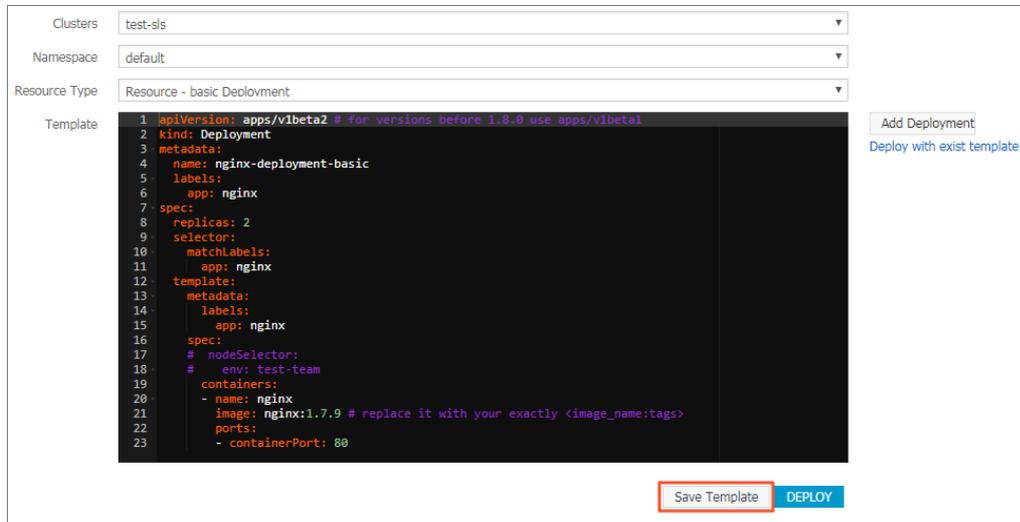
Template:

```
1 apiVersion: apps/v1beta2 # for versions before 1.8.0 use
2   apps/v1beta1
3   kind: Deployment
4   metadata:
5     name: tomcat-deployment
6     labels:
7       app: tomcat
8   spec:
9     replicas: 1
10    selector:
11      matchLabels:
12        app: tomcat
13    template:
14      metadata:
15        labels:
16          app: tomcat
17      spec:
18        containers:
19          - name: tomcat
20            image: tomcat # replace it with your exactly
21              <image_name;tags>
22              ports:
23                - containerPort: 8080
```

4. After the template is created, you are redirected to the **Templates** page by default. You can find the template on the **My Templates** tab.



5. (Optional) You can also choose **Applications > Deployments** in the left-side navigation pane, and click **Create from Template** to go to the **Create from Template** page. You can modify a built-in template provided by Container Service and save it as a custom template.
 - i. Select a built-in template and click **Save Template**.



- ii. In the dialog box that appears, specify the name, description, and content. Click **Save** to save the template.

Note You can modify the built-in template based on your needs.

- iii. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. You can find the newly created template on the **My Templates** tab.



What's next

You can use the orchestration templates on the **My Templates** tab to quickly create applications.

3.4.10.2. Update an orchestration template

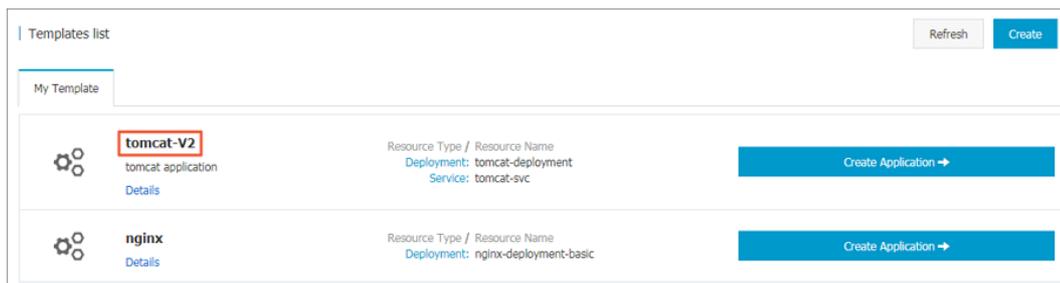
This topic describes how to edit and update an orchestration template.

Prerequisites

You have created an orchestration template. For more information, see [Create orchestration templates](#).

Procedure

1. **Log on to the Container Service console.**
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Edit** in the upper-right corner.
5. In the dialog box that appears, edit the name, description, and template content, and click **Save**.
6. Go to the **Templates** page. You can view the template that you have updated on the **My Templates** tab.



3.4.10.3. Save an orchestration template as a new one

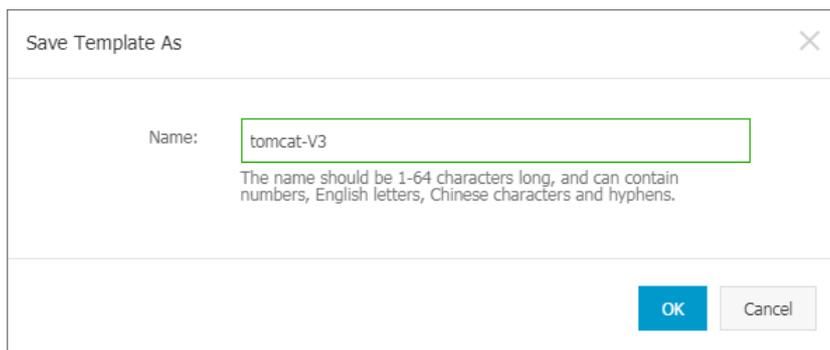
This topic describes how to save an orchestration template as a new one.

Prerequisites

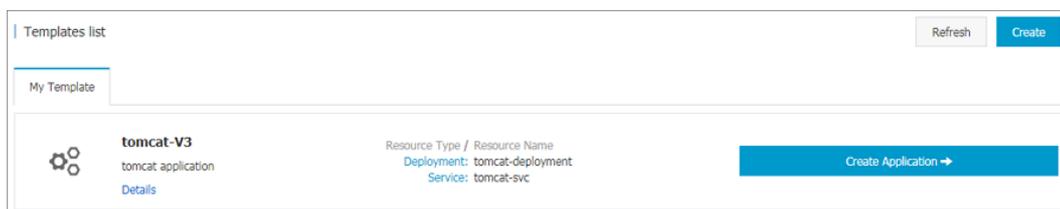
You have created an orchestration template. For more information, see [Create orchestration templates](#).

Procedure

1. Log on to the [Container Service for Kubernetes console](#).
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, modify the template and click **Save As** in the upper-right corner.
5. In the dialog box that appears, enter the template name and click **OK**.



6. Go to the **Templates** page. The newly saved template is displayed on the **My Templates** tab.



3.4.10.4. Download an orchestration template

This topic describes how to download an orchestration template.

Prerequisites

You have created an orchestration template. For more information, see [Create orchestration templates](#).

Procedure

1. [Log on to the Container Service console.](#)
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Download** in the upper-right corner to download the template as a YAML file.

3.4.10.5. Delete an orchestration template

Prerequisites

You have created an orchestration template. For more information, see [Create orchestration templates](#).

Procedure

1. [Log on to the Container Service console.](#)
2. In the left-side navigation pane, choose **Marketplace > Orchestration Templates**. The **Templates** page appears. You can view existing templates on the **My Templates** tab.
3. Select the target template and click **Details**.
4. On the template details page, click **Delete** in the upper-right corner.
5. In the dialog box that appears, click **OK**.

3.4.11. Images

3.4.11.1. Create an image repository

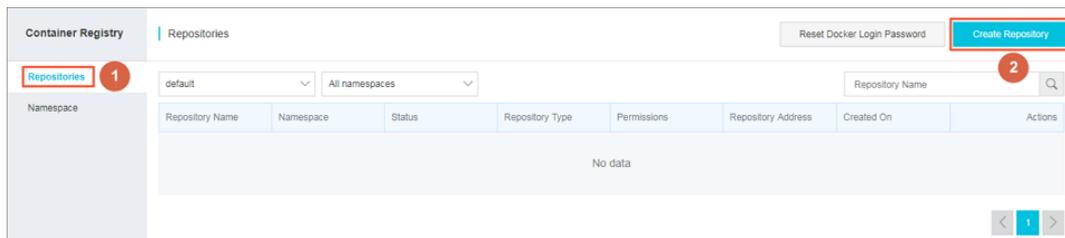
This topic describes how to create an image repository in the Container Registry console.

Prerequisites

The Container Registry service is activated and a namespace is created in the Container Registry service.

Procedure

1. [Log on to the Container Registry console.](#)
2. In the left-side navigation pane, click **Repositories** to go to the **Repositories** page. Click **Create Repository** in the upper-right corner of the page.



3. In the dialog box that appears, set the following parameters and click **Next**.
 - **Region:** Use the default value. The region of the repository must be the same as that of the cluster.
 - **Namespace:** Use the default value. The namespace must be the same as that of the organization that you select when you log on to the console.
 - **Repository Name:** The repository name must be 2 to 64 characters in length and can contain lowercase letters, digits, and special characters, including underscores (_), hyphens (-), and periods (.). It cannot start or end with a special character.
 - **Summary:** Enter the repository summary.
 - **Description:** Enter description information. The description must be 0 to 100 characters in length.

- **Repository Type:** The repository type can be public or private.

Create Repository

1 ————— 2

Repository Info Code Source

Region

* Namespace

* Repository Name

Repository name length: 2-64 characters. The name can contain lowercase English letters numbers and the separators _ - and . (separators cannot be the first or last character)

* Summary

Max. 100 characters

Description

Supports Markdown Format

Repository Type Public Private

Next Cancel

4. Set the code source and click Create Repository.

Note Currently, only local repositories are supported. You can push images to an image repository by using the command-line interface (CLI).

Create Repository

✓ ————— 2

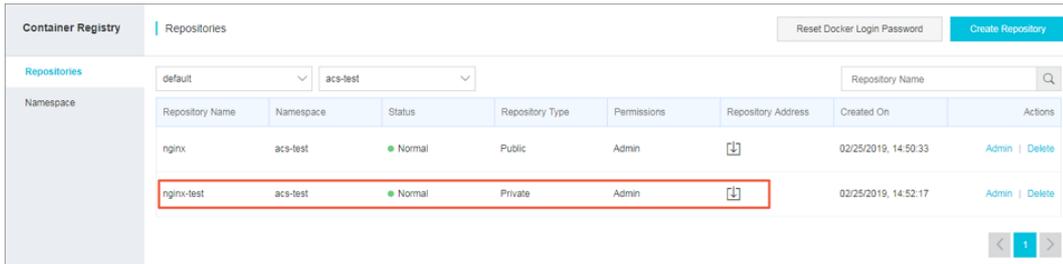
Repository Info Code Source

Code Source

You can use the command line to push this image to the image repository.

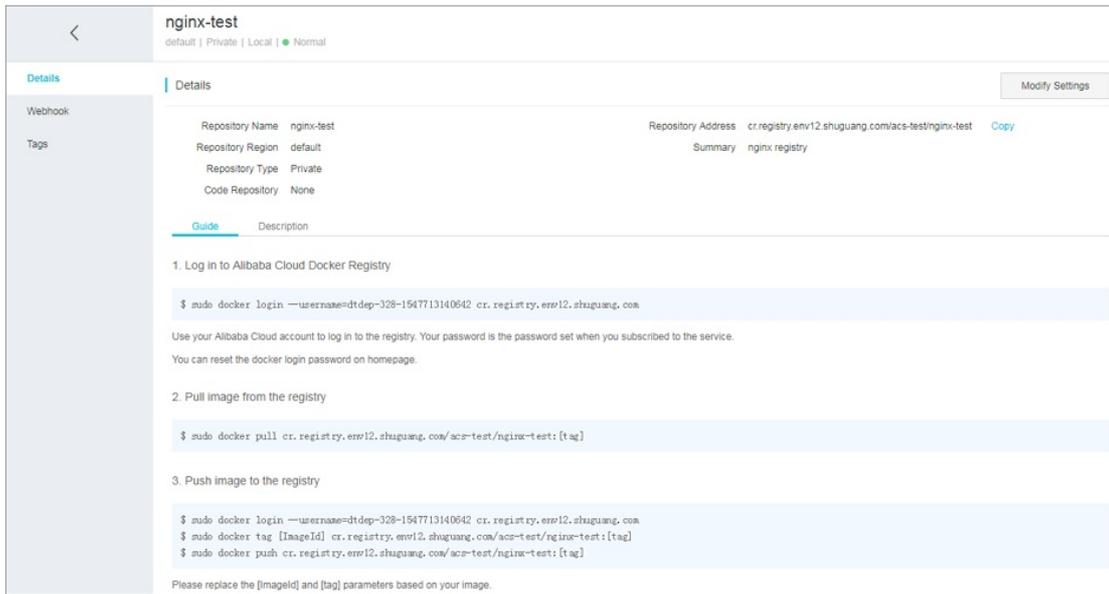
Previous **Create Repository** Cancel

5. Go to the Repositories page to view the newly created repository.



What's next

You can click **Admin** in the **Actions** column for the repository to go to the **Details** page and learn how to manage the repository.



3.4.11.2. Create a namespace

This topic describes how to create a namespace in the Container Registry console.

Context

A namespace is a collection of repositories. We recommend that you place the repositories of a company or an organization in the same namespace.

- You can enter a name that corresponds to a company name, such as aliyun or alibaba.
- You can enter a name that corresponds to a team or organization name for the namespace, such as misaka-team.

Procedure

1. **Log on to the Container Registry console**
2. In the left-side navigation pane, click **Namespace**. On the page that appears, click **Create Namespace** to create a namespace.

Create Namespace ✕

Each account can create at most 5 namespaces. [Tips](#)

* Namespace

Define your namespace. After being set it cannot be modified. The namespace must be 2-30 characters long and can contain lowercase English letters numbers and the separators _ and - (separators cannot be the first or last character)

[Confirm](#)

You can find the newly created namespace on the Namespace page.

? Note

- You can turn on or turn off **Automatically Create Repository** to specify whether to automatically create image repositories for the namespace.
- You can set the repository type to **Public** or **Private** for the namespace.

Container Registry	Namespace						Create Namespace
Repositories	Namespace	Permissions	Status	Automatically Create Repository	Default Repository Type	Actions	
Namespace	acs-test	Admin	● Normal	<input checked="" type="checkbox"/> On	<input type="radio"/> Public <input checked="" type="radio"/> Private	Delete	
	acs-registry	Admin	● Normal	<input checked="" type="checkbox"/> On	<input type="radio"/> Public <input checked="" type="radio"/> Private	Delete	

What's next

You can create more image repositories for the namespace.

3.4.11.3. Synchronize an image

Container Registry can be used to synchronize images across data centers. This allows you to retrieve images from the data center that is nearest to where your workloads are deployed.

Prerequisites

- A namespace is created. For more information, see [Create a namespace](#).
- An image repository for the namespace is created. For more information, see [Create a repository](#).

Procedure

1. [Log on to the Container Registry console](#).
2. In the left-side navigation pane, choose **Repositories** to go to the **Repositories** page.
3. Select the target repository and click **Admin** in the **Actions** column for the repository to go to the **Details** page.
4. In the left-side navigation pane, choose **Tags**. Find the target tag and click **Sync** in the **Actions** column for the tag.
5. On the **Sync** page, specify the tag and target repository, and click **OK**. A message appears to indicate that the synchronization request is submitted.

Result

In the left-side navigation pane of the target repository, click **Sync** to check the status of the synchronization

task.

You can click **Details** in the **Actions** column for the task to check the task status for the specified image.

3.4.11.4. Sign and verify an image

When you manage container images, you can use content trust to verify both the integrity and the publisher of images. Image publishers can encrypt images by using digital signatures that are stored in Container Registry. We recommend that you verify the signatures to ensure that only images signed by trusted authorities are deployed. This way, you can minimize or stop attacks that may occur when you run containers.

Install and configure signature tools

1. Install the aliyun client tool. For more information, see [Install Alibaba Cloud CLI](#).
2. On the command line, run the following command to configure aliyun client:

```
./aliyun configure set \
--profile akProfile \
--mode AK \
--region cn-qingdao-env17-d01 \
--access-key-id yourAK \
--access-key-secret yourAK
```

3. After you install and configure the GPG tool, run the following command to export the public key:

```
#Query the pubKeyId.
gpg --fingerprint

#Export the public key.
gpg --armor --output public-key.txt --export yourUser
```

 **Note** For Apsara Stack instances, instanceId is set to default.

Sign container images

1. On the command line, run the following command to determine the image tag URL:

```
echo image://region/instanceId/namespace/repo@digest | tee imageURL.txt

#e.g.
#echo image://cn-qingdao-env17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78d2ea5
b382841ca6569fd174bd24c14bfff3dca > imageURL.txt
```

2. On the command line, run the following command to use GPG to sign the unique URL of the image. By default, imageURL.txt.asc is generated.

```
gpg --armor --sign imageURL.txt
```

The imageURL.txt.asc file includes the following content:

```
-----BEGIN PGP MESSAGE-----  
  
owGbwMvMwMEo63vGqaX74wXGNXVJ3CmpaYmlOSV6JRulcZy7vmfmJqanWunr****  
  
.....  
=O2TP  
  
-----END PGP MESSAGE-----
```

3. On the command line, run the following command to create a metadata namespace:

```
./aliyun cr CreateMetadataNamespace --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --NameSpaceName kritis-test-2 --Description "for test"
```

4. On the command line, run the following command to create a signature note:

```
cat <<EOF > note.json| jq  
{  
  "name": "/namespaces/kritis-test-2/notes/image-sign",  
  "LongDescription": "long",  
  "ShortDescription": "short",  
  "ExpirationTime": "2021-01-01T00:00:00Z",  
  "Kind": "ATTESTATION",  
  "Attestation": {  
    "Hint": {  
      "HumanReadableName": "ACR"  
    }  
  }  
}  
EOF
```

5. On the command line, run the following command to create an occurrence for the image signature and save the occurrence to the metadata service:

```
cat <<EOF > occurrence.json | jq
{
  "Name": "/namespaces/kritis-test-2/occurrences/randomId1",
  "NoteName": "/namespaces/kritis-test-2/notes/image-sign",
  "ResourceUri": "image://cn-qingdao-env17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bff****",
  "Kind": "ATTESTATION",
  "Attestation": {
    "Signatures": [
      {
        "Signature": $(cat imageURL.txt.asc|jq -R --slurp),
        "PublicKeyId": "E5B5FF2AFC3A1D70FE3CE57C1D4DCC42848****"
      }
    ]
  }
}
EOF
```

```
./aliyun cr CreateMetadataOccurrence --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --Name
spaceName kritis-test-2 --OccurrenceName randomId1 --Occurrence "$(cat occurrence.json)"
```

6. On the command line, run the following command to query the image signature:

```
#Retrieve a list of occurrences for a specified note. In this example, one occurrence is created.
./aliyun cr ListMetadataOccurrences --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --Name
spaceName kritis-test-2 --NoteName image-sign --PageNo 1 --PageSize 5

#Retrieve a list of occurrences for a specified note that occurs on a resource. In this example, one occurrence is cr
eated.
./aliyun cr ListMetadataOccurrences --force --version 2018-12-01 --endpoint cr.inter.env17e.shuguang.com --Name
spaceName kritis-test-2 --NoteName image-sign --PageNo 1 --PageSize 5 --ResourceURIs '["image://cn-qingdao-e
nv17-d01/default/kritis-test/busybox2@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14b
fff****"]'
```

Verify an image signature

The following steps describe how to install an image signature verification component and enable the verification feature on Apsara Stack.

1. Install the image signature verification component.
 - i. [Log on to the Container Service console](#).
 - ii. Go to the Clusters page, find the cluster in which you want to install the image signature verification component, and in the Actions column for the cluster, choose **More** > **Install Kritis**.
2. Configure the signature verification policy.

- i. Set the following signature parameters to configure the signature verification policy:

```
$ export namespace=Actual value of namespace
$ export noteName=Actual value of noteName
$ export publicKeyData=Actual value of publicKeyData
```

- namespace: the ACR namespace setting used for signing an image.
- noteName: the noteName setting used for signing an image.
- publicKeyData: the Base64-encoded GPG public key.

- ii. On the command line, run the following command to set AttestationAuthority:

 **Note** The default namespace is used in the following example.

```
$ cat <<EOF > aa.yaml
apiVersion: kritis.grafeas.io/v1beta1
kind: AttestationAuthority
metadata:
  name: ${noteName}
spec:
  noteReference: namespaces/${namespace}
  publicKeyData: ${publicKeyData}
EOF

$ kubectl -n default apply -f aa.yaml
```

- iii. On the command line, run the following command to set GenericAttestationPolicy:

 **Note** The default namespace is used in the following example.

```
$ cat <<EOF > gap.yaml
apiVersion: kritis.grafeas.io/v1beta1
kind: GenericAttestationPolicy
metadata:
  name: my-gap
spec:
  attestationAuthorityNames:
  - ${noteName}
EOF

$ kubectl -n default apply -f gap.yaml
```

3. Test the signature verification feature.

In the test, a signed image is used: registry.acs.example.com/kritis-test/signed@sha256:2f122941b5850006dbb7adda78d2ea5b382841ca6569fd174bd24c14bfff****. An unsigned image is also used: registry.acs.example.com/kritis-test/not-sign@sha256:efc961b2b3499c25753d3c9f29977f494f49125cf1191071057aa68bffa7****. If the feature functions as expected, the signature verification policy enables the signed image and disables the unsigned image for the default namespace. The following example shows how to test the feature:

```
#A deployment is created based on the signed image.
$ kubectl -n default run test-signed --image=registry.acs.example.com/kritis-test/signed@sha256:2f122941b58500
06dbb7adda78d2ea5b382841ca6569fd174bd24c14bff****
deployment.apps/test-signed created

#The unsigned image fails a deployment.
$ kubectl -n default run test-not-signed --image=registry.acs.example.com/kritis-test/not-sign@sha256:efc961b2b349
9c25753d3c9f29977f494f49125cf1191071057aa68bffa7****
Error from server: admission webhook "kritis-validation-hook-deployments.grafeas.io" denied the request: image
registry.acs.example.com/kritis-test/not-sign@sha256:efc961b2b3499c25753d3c9f29977f494f49125cf1191071057aa68bffa
7**** is not attested
```

 **Note** After you enable image signature verification, when you create resources, the image ID must be a digest value in the format such as @sha256:<hash>.

Appendix 1: Use GPG commands to generate publicKeyData

1. On the command line, run the following command to find the local GPG key user that you want to use:

 **Note** The user in the following example is `abcdef@example.com`.

```
$ gpg --list-keys
pub rsa2048 2020-01-08 [SC] [Expired on: 2022-01-07]
    7726310BC6E11E9B57B9CC08E2932E4363F3***
uid [Absolute] abcdef <abcdef@example.com>
sub rsa2048 2020-01-08 [E] [Expired on: 2022-01-07]
```

2. Export the public key of this user and encode the content of the public key in the Base64 format to generate publicKeyData.

```
$ gpg --armor --export <user> |base64 | tr -d '\n'

#In the example, the user is abcdef@example.com.
#Run this command to generate publicKeyData: gpg --armor --export abcdef@example.com |base64 | tr -d '\n'
# export publicKeyData=$(gpg --armor --export abcdef@example.com |base64 | tr -d '\n')
```

Appendix 2: Retrieve an image digest value

You can use the following method to find an image digest value. In this example, the image URL is `registry.acs.example.com/kritis-test/alpine:3.11`.

```
$ docker pull registry.acs.example.com/kritis-test/alpine:3.11
$ docker images --digests | grep registry.acs.example.com/kritis-test/alpine
registry.acs.example.com/kritis-test/alpine 3.11 sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45
e7d92cdc71fe 2 months ago 5.59MB
```

The output shows that the image digest value is `sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45`. When you create resources, the following image ID must be used: `registry.acs.example.com/kritis-test/alpine:sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45`.

3.4.11.5. Synchronize images between instances

You can configure synchronization rules to automatically synchronize images from a source instance to a destination instance. By default, images are automatically synchronized for instances that are deployed in different regions.

Procedure

1. [Log on to the Container Registry console](#).
2. Go to the **Repositories** page, find the repository that you want to manage, and then click **Admin** in the **Actions** column for the repository.
3. The **Details** page is displayed. In the left-side navigation pane, click **Sync**.
4. In the upper-right corner of the **Sync** page, click **Create**. In the **Sync** dialog box, set the **Tag** and **Target** parameters, specify the namespace, repository name, and tag of the target repository, and then click **OK**. The synchronization rule is created.

Result

After the synchronization rule is created, a synchronization task is automatically triggered when a new image is uploaded to the repository whose name matches the specified synchronization rule.

3.4.12. Create a batch release

Context

Note In Kubernetes clusters of the latest version, alicloud-application-controller is installed by default. This component is only available in Kubernetes 1.9.3 and later. You can upgrade your cluster through the console.

Procedure

1. [Log on to the Container Service console](#).
2. In the left-side navigation pane, choose **Applications > Releases**. Click the **Batch Release** tab and click **Create Batch Release** in the upper-right corner.

Note If the button is dimmed, it indicates that you need to upgrade your cluster first.

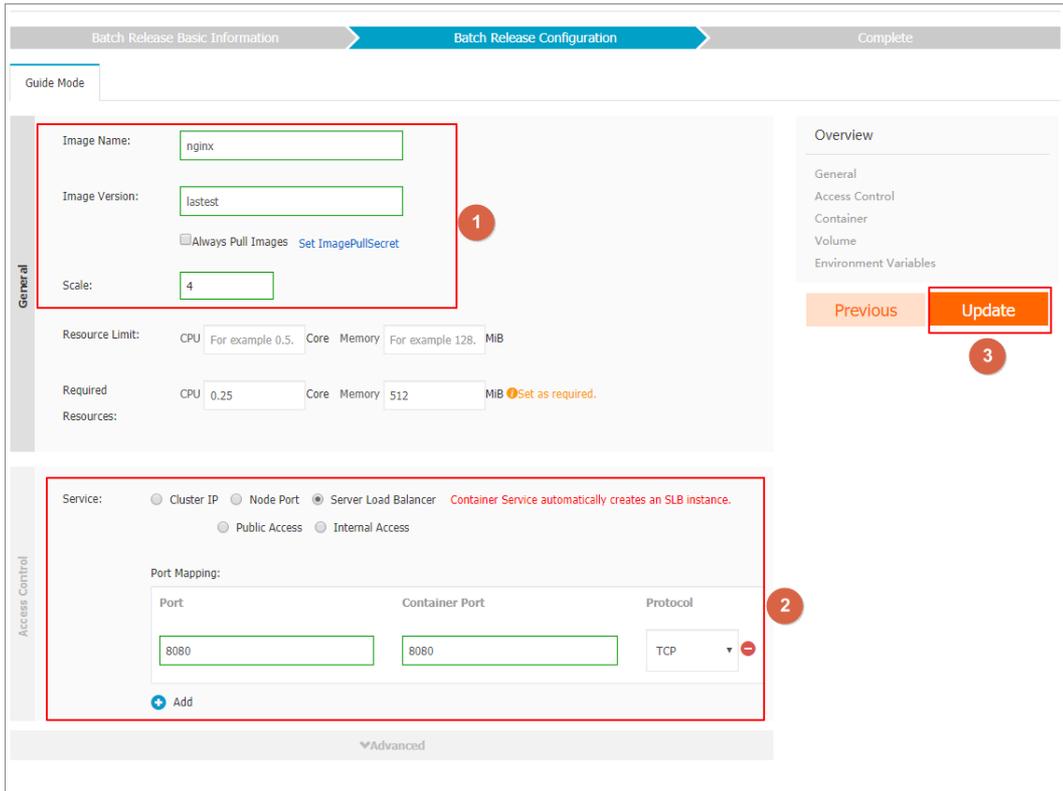
3. On the **Batch Release Basic Information** tab, set the following parameters: application name, cluster, namespace, and release option. Click **Next**.

The screenshot shows the 'Create Batch Release' interface. At the top, there is a breadcrumb 'Create Batch Release' and a link 'Return to Release List'. Below this is a progress bar with three steps: 'Batch Release Basic Information' (active), 'Batch Release Configuration', and 'Complete'. The main form area contains the following fields:

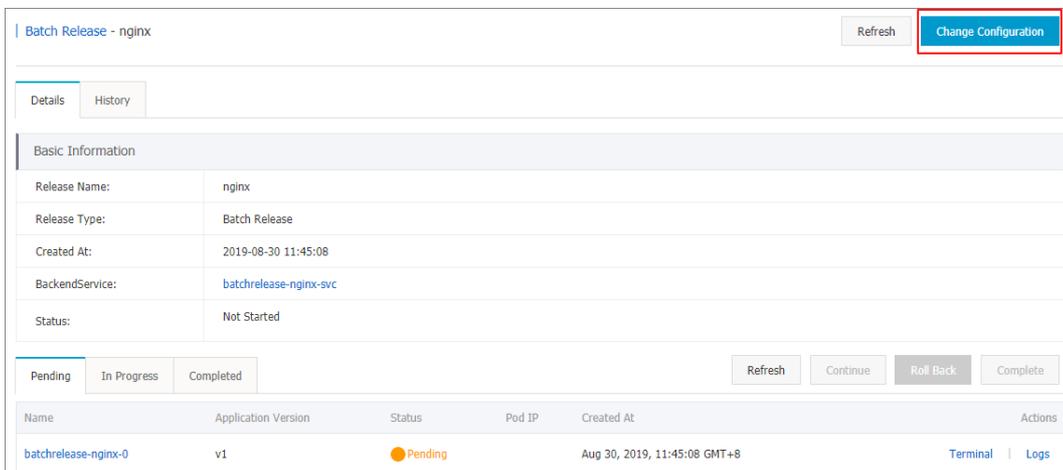
- Name:** A text input field containing 'nginx'. Below it is a note: 'The name must be 1 to 64 characters in length and can contain numbers letters and hyphens (-). The name cannot start with a hyphen (-).'.
- Cluster:** A dropdown menu with 'k8s-cluster' selected.
- Namespace:** A dropdown menu with 'default' selected.
- Release Option:** A dropdown menu with 'Release in two batches. Suspend the release of th' selected.

At the bottom right of the form, there are two buttons: 'Back' and 'Next'. The 'Next' button is highlighted with a red rectangular box.

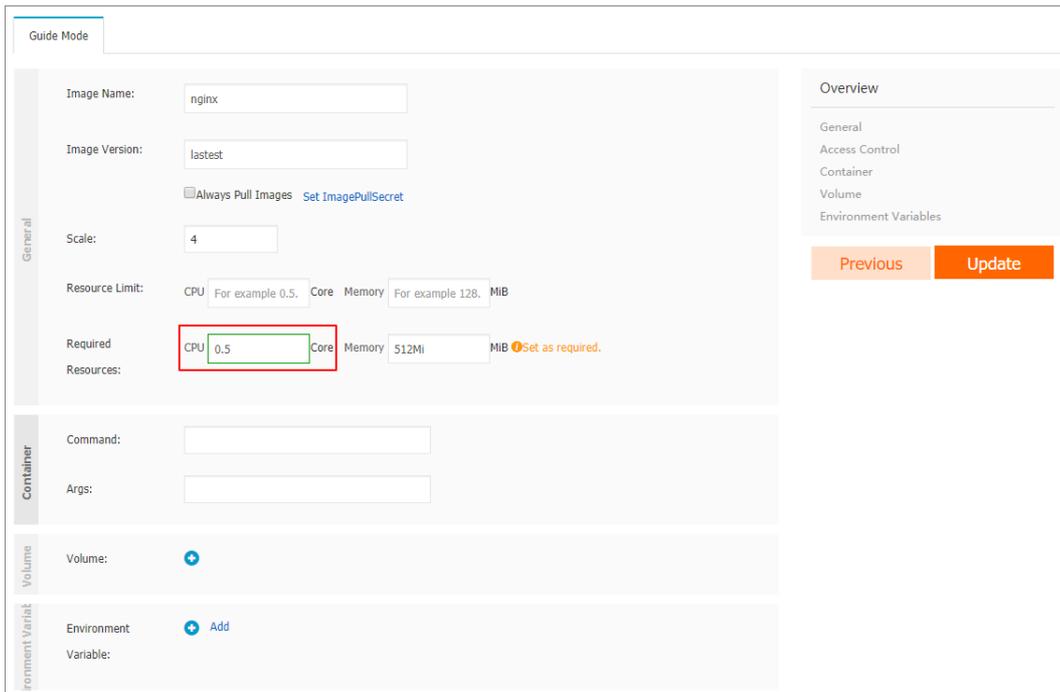
4. On the **Batch Release Configuration** tab, configure the pods and service, and then click **Update** to create an application.



5. Go to the Releases page and click the Batch Release tab. You can find the newly created application and its status is **Not Started**. Click **Details** in the Actions column.
6. On the Details tab, you can find more information about the application. Click **Change Configuration** in the upper-right corner to change the application configuration.



7. On the page that appears, change the configuration and then click **Update**.



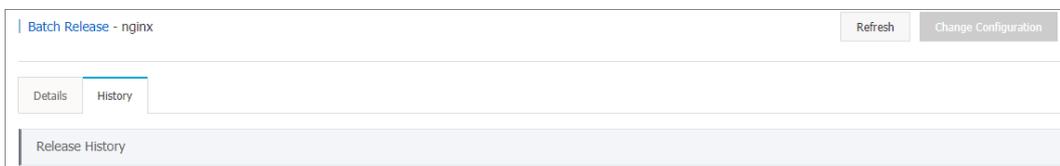
8. You are redirected to the Releases page by default. Click the Batch Release tab and you can find the application status. After the first batch is deployed, click Details.



9. On the details tab, two pods are listed in the Not Started list and two pods are listed in the Completed list. This indicates that the first batch has been released. Click Continue to release the second batch of pods. Click Roll Back to roll back to the previous version.



10. After the release is completed, click the History tab and you can choose to roll back to a previous version.



What's next

You can create a batch release to quickly verify the functionalities of a new application version while serving all production traffic. A batch release requires less resources than a blue-green release. Currently, you can only create batch releases through the wizard. Support for YAML configuration files will be available soon.

3.4.13. Use Log Service to collect Kubernetes logs

Container Service for Kubernetes is integrated with Log Service. When you create a cluster, you can enable Log Service to collect container logs, including standard outputs and text files.

Activate Log Service

To activate Log Service, perform the following steps:

1. Log on to the Apsara Stack Cloud Management (ASCM) console. In the top navigation bar, choose **Products > Log Service** to go to the Log Service page.
2. Select the target organization and region.
3. Click **SLS** to go to the SLS console.

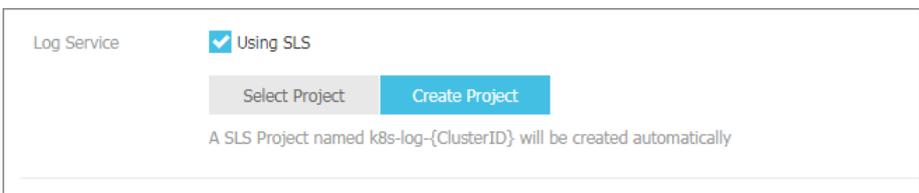
Create a cluster and enable Log Service

To create a Kubernetes cluster, perform the following steps:

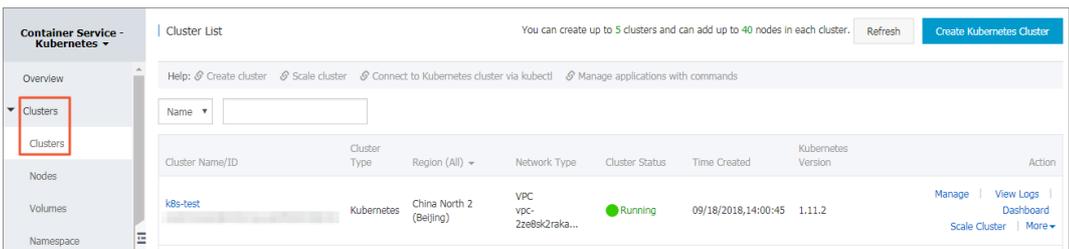
1. **Log on to the Container Service for Kubernetes console.**

Note The specified organization must be the same as the one you selected when you activate Log Service. For more information, see [Activate Log Service](#).

2. In the left-side navigation pane, choose **Clusters > Clusters**. The Clusters page appears.
3. Click **Create Kubernetes Cluster**. For more information about the configurations, see [Create a Kubernetes cluster](#).
4. Select **Enable Log Service** to install the logging agent.
5. After you select the check box, you must specify a project to store log data. You can click **Select Project** and select an existing project from the drop-down list that appears. You can also click **Create Project** to allow the system to automatically create a project for log management. The project is named `k8s-log-{ClusterID}`, where ClusterID is the unique identifier of the cluster.



6. After you set the parameters, click **Create Cluster** in the upper-right corner. In the dialog box that appears, click **OK** to create the cluster. You can find the created cluster on the Clusters page.



Install Log Service components in an existing cluster

If you have already created a Kubernetes cluster and activated Log Service, you can perform the following steps to use Log Service:

1. Connect to the Kubernetes cluster by using CloudShell. For more information, see [Connect to a Kubernetes cluster through kubectl](#).
2. Run the script `logtail-dedicated.sh` to install Log Service components in the cluster.

```
#!/env/bin/bash

yaml=$(cat <<-END
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: alibaba-log-config-file
  namespace: kube-system
data:
  ilogtail_config.json: |
    {
      "config_server_address": "http://logtail.$REGION.sls-pub.$INTERNET_DOMAIN",
      "data_server_address": "http://data.$REGION.sls-pub.$INTERNET_DOMAIN",
      "data_server_list":
      [
        {
          "cluster": "$REGION",
          "endpoint": "data.$REGION.sls-pub.$INTERNET_DOMAIN"
        }
      ],
      "shennong_unix_socket": false
    }
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: alibaba-log-configuration
  namespace: kube-system
data:
  log-project: "k8s-log-$CLUSTER_ID"
  log-endpoint: "data.$REGION.sls-pub.$INTERNET_DOMAIN"
  log-machine-group: "k8s-group-$CLUSTER_ID"
  log-config-path: "/etc/ilogtail/conf/apsara/ilogtail_config.json"
  log-ali-uid: "$ALI_UID"
  log-access-id: "" # just use blank string
  log-access-key: "" # just use blank string
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: alibaba-log-controller
```

```
name: alibaba-log-controller
namespace: kube-system
labels:
  k8s-app: alibaba-log-controller
annotations:
  component.version: "v0.1.3"
  component.revision: "v1"
spec:
  replicas: 1
  template:
    metadata:
      labels:
        k8s-app: alibaba-log-controller
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: "
    spec:
      serviceAccountName: alibaba-log-controller
      tolerations:
        - operator: "Exists"
      containers:
        - name: alibaba-log-controller
          image: $IMAGE_REPO_URL/acs/log-controller-$ARCH:v0.1.3.0-527ff4d-aliyun
          resources:
            limits:
              memory: 100Mi
            requests:
              cpu: 50m
              memory: 100Mi
          env:
            - name: "ALICLOUD_LOG_PROJECT"
              valueFrom:
                configMapKeyRef:
                  name: alibaba-log-configuration
                  key: log-project
            - name: "ALICLOUD_LOG_ENDPOINT"
              valueFrom:
                configMapKeyRef:
                  name: alibaba-log-configuration
                  key: log-endpoint
            - name: "ALICLOUD_LOG_MACHINE_GROUP"
              valueFrom:
                configMapKeyRef:
                  name: alibaba-log-configuration
                  key: log-machine-group
            - name: "ALICLOUD_ACS_K8S_FLAG"
              value: "ture"
            - name: "ALICLOUD_ACCESS_KEY_ID"
```

```
valueFrom:
  configMapKeyRef:
    name: alibaba-log-configuration
    key: log-access-id
- name: "ALICLOUD_ACCESS_KEY_SECRET"
  valueFrom:
    configMapKeyRef:
      name: alibaba-log-configuration
      key: log-access-key
nodeSelector:
  beta.kubernetes.io/os: linux
---
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: aliyunlogconfigs.log.alibabacloud.com
spec:
  group: log.alibabacloud.com
  version: v1alpha1
  names:
    kind: AliyunLogConfig
    plural: aliyunlogconfigs
  scope: Namespaced
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: alibaba-log-controller
subjects:
- kind: ServiceAccount
  name: alibaba-log-controller
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: alibaba-log-controller
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: alibaba-log-controller
  labels:
    k8s-app: alibaba-log-controller
rules:
- apiGroups: ["log.alibabacloud.com"]
  resources:
  - aliyunlogconfigs
```

```
verbs:
- update
- get
- watch
- list
- apiGroups: [""]
resources:
- configmaps
verbs:
- create
- update
- get
- apiGroups: [""]
resources:
- events
verbs:
- create
- patch
- update
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: alibaba-log-controller
  namespace: kube-system
  labels:
    k8s-app: alibaba-log-controller
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: logtail-ds
  namespace: kube-system
  labels:
    k8s-app: logtail-ds
  annotations:
    component.version: "v0.16.16"
    component.revision: "v0"
spec:
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        k8s-app: logtail-ds
      annotations:
        scheduler.alpha.kubernetes.io/critical-path:
```

```
scheduler.alpha.kubernetes.io/critical-pod: ""
spec:
  tolerations:
    - operator: "Exists"
  containers:
    - name: logtail
      image: $IMAGE_REPO_URL/acs/logtail-$ARCH:v0.16.24.0-c46cd2fe-aliyun
  resources:
    limits:
      memory: 512Mi
    requests:
      cpu: 100m
      memory: 256Mi
  livenessProbe:
    exec:
      command:
        - /etc/init.d/ilogtaild
        - status
    initialDelaySeconds: 30
    periodSeconds: 30
  securityContext:
    privileged: false
  env:
    - name: "ALIYUN_LOGTAIL_CONFIG"
      valueFrom:
        configMapKeyRef:
          name: alibaba-log-configuration
          key: log-config-path
    - name: "ALIYUN_LOGTAIL_USER_ID"
      valueFrom:
        configMapKeyRef:
          name: alibaba-log-configuration
          key: log-ali-uid
    - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
      valueFrom:
        configMapKeyRef:
          name: alibaba-log-configuration
          key: log-machine-group
    - name: "ALICLOUD_LOG_DOCKER_ENV_CONFIG"
      value: "true"
    - name: "ALICLOUD_LOG_ECS_FLAG"
      value: "ture"
    - name: "ALICLOUD_LOG_DEFAULT_PROJECT"
      valueFrom:
        configMapKeyRef:
          name: alibaba-log-configuration
          key: log-project
```

```
- name: "ALICLOUD_LOG_ENDPOINT"
  valueFrom:
    configMapKeyRef:
      name: alibaba-log-configuration
      key: log-endpoint
- name: "ALICLOUD_LOG_DEFAULT_MACHINE_GROUP"
  valueFrom:
    configMapKeyRef:
      name: alibaba-log-configuration
      key: log-machine-group
- name: "ALICLOUD_LOG_ACCESS_KEY_ID"
  valueFrom:
    configMapKeyRef:
      name: alibaba-log-configuration
      key: log-access-id
- name: "ALICLOUD_LOG_ACCESS_KEY_SECRET"
  valueFrom:
    configMapKeyRef:
      name: alibaba-log-configuration
      key: log-access-key
- name: "ALIYUN_LOG_ENV_TAGS"
  value: "_node_name_|_node_ip_"
- name: "_node_name_"
  valueFrom:
    fieldRef:
      fieldPath: spec.nodeName
- name: "_node_ip_"
  valueFrom:
    fieldRef:
      fieldPath: status.hostIP
volumeMounts:
- name: sock
  mountPath: /var/run/docker.sock
- name: root
  mountPath: /logtail_host
  readOnly: true
- name: alibaba-log-config-file-volume
  mountPath: /etc/ilogtail/conf/apsara
  readOnly: true
terminationGracePeriodSeconds: 30
nodeSelector:
  beta.kubernetes.io/os: linux
volumes:
- name: sock
  hostPath:
    path: /var/run/docker.sock
    type: Socket
```

```

- name: root
  hostPath:
    path: /
    type: Directory
- name: alibaba-log-config-file-volume
  configMap:
    name: alibaba-log-config-file
END
)

echo "$yaml" > logtail.yml

kubectl create -f logtail.yml

```

3. Set `<your_server_architecture>` , `<your_k8s_cluster_region_id>` , `<your_k8s_cluster_id>` , `<k8s_cluster_domain_suffix>` , `<your_ali_uid>` , and `<your_image_repo_url>` to the actual values. Then, run the following commands to set the environment variables for deploying the components.

```

export ARCH=<your_server_architecture>
export REGION=<your_k8s_cluster_region_id>
export CLUSTER_ID=<your_k8s_cluster_id>
export INTERNET_DOMAIN=<k8s_cluster_domain_suffix>
export IMAGE_REPO_URL=<your_image_repo_url>
export ALI_UID=<your_ali_uid>
bash logtail-dedicated.sh // Run the script to install the components.

```

Note

- `<your_server_architecture>` : the server architecture, for example, amd64.
- `<your_k8s_cluster_region_id>` : the region where the cluster is deployed, for example, cn-qingdao-apsara-d01.
- `<your_k8s_cluster_id>` : the ID of the cluster.
- `<k8s_cluster_domain_suffix>` : the domain suffix of the cluster, for example, env28.internet.com.
- `<your_ali_uid>` : the Apsara Stack tenant account ID, for example, 1234074238634394.
- `<your_image_repo_url>` : the URL of the image repository, for example, registry.cn-hangzhou.aliyuncs.com.

Create an application and configure Log Service

When you create an application in Container Service for Kubernetes, you can configure Log Service to collect container logs. Currently, you can only use YAML templates to configure Log Service.

1. In the left-side navigation pane, choose **Applications > Deployments**. In the upper-right corner of the Deployments page, click **Create from Template**.
2. Set environment variables in the ENV field to configure log collection and custom tags. To support the specified log collection, you must set the volumeMounts and volumes fields. YAML templates follow the Kubernetes syntax. The following example shows how to configure Log Service in a pod:

```

apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    app: logtail-test
  name: logtail-test
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: logtail-test
        name: logtail-test
    spec:
      containers:
      - name: logtail
        image: registry.acs.env28.intranet.com/acs/busybox:latest
        args:
        - ping
        - 127.0.0.1
        env:
        - name: aliyun_logs_log-stdout
          value: stdout
        - name: aliyun_logs_log-varlog
          value: /log/*.log
        - name: aliyun_logs_log_tags
          value: tag1=v1
        volumeMounts:
        - name: volumn-sls
          mountPath: /log
      volumes:
      - name: volumn-sls
        emptyDir: {}

```

- Specify the following configurations in sequence to fit your needs:
- Use environment variables to configure log collection and custom tags. All environment variables must be prefixed with `aliyun_logs_`.
- Log collection is configured in the following format:

```

- name: aliyun_logs_{Logstore name}
  value: {Log path}

```

In the preceding example, two environment variables are used to configure log collection. The environment variable `aliyun_logs_log-stdout` instructs the system to create a Logstore named `log-stdout`. The Logstore collects standard output logs from the container in the pod.

 **Note** A Logstore name cannot contain underscores (`_`). You can use hyphens (`-`) instead.

- Custom tags are configured in the following format:

```
- name: aliyun_logs_{Tag name without underscores (_)}_tags
  value: {Tag name}={Tag value}
```

After a custom tag is set, it is automatically appended to certain log fields when the Logstore collects logs from the container.

- If you specify a log path to collect log data other than standard output log data, you must add the volumeMounts field.

In the preceding example, the volumeMounts field is added and set to `/var/log`. This allows the specified Logstore to collect `/var/log/*.log` files.

3. After you edit the YAML template, click **Create** to submit the configurations.

Advanced configurations

You can set more environment variables to apply advanced configurations to log collection. The following table provides details of these variables.

Variable	Description	Example	Note
aliyun_logs_{key}	<ul style="list-style-type: none"> • Required. The key field can contain lowercase letters, digits, and hyphens (-), and cannot contain underscores (_). • If the environment variable aliyun_logs_{key}_logstore is not set, a Logstore named {key} is created to collect logs. • To collect standard output logs of the container, set the value to stdout. You can also set the value to another Logstore path. 	<pre>- name: aliyun_logs_catalina stdout - name: aliyun_logs_access-log /var/log/nginx/access.log</pre>	<ul style="list-style-type: none"> • By default, the simple mode is used to collect logs. To parse logs, we recommend that you use the Log Service console. • The key field must be set to a unique value in the cluster.
aliyun_logs_{key}_tags	Optional. This variable is used to add tags to log data. The value must be in the following format: {tag-key}={tag-value}.	<pre>- name: aliyun_logs_catalina_tags app=catalina</pre>	-
aliyun_logs_{key}_project	Optional. This variable specifies a project in Log Service. By default, the project that you specified when you create the cluster is used to manage related logs and resources in Log Service.	<pre>- name: aliyun_logs_catalina_project my-k8s-project</pre>	The region of the project must be the same as where your Logtail is located.

Variable	Description	Example	Note
aliyun_logs_{key}_logstore	Optional. This variable specifies a Logstore in Log Service. By default, the Logstore is named after {key}.	<pre>- name: aliyun_logs_ca talina_tags my-logstore</pre>	-
aliyun_logs_{key}_shard	Optional. This variable specifies the number of shards in the Logstore. Valid values: 1 to 10. Default value: 2.	<pre>- name: aliyun_logs_ca talina_shard 4</pre>	-
aliyun_logs_{key}_ttl	Optional. This variable specifies the number of days for which log data is retained. Valid values: 1 to 3650. <ul style="list-style-type: none"> To retain log data permanently, set the value to 3650. Default value: 90. 	<pre>- name: aliyun_logs_ca talina_ttl 3650</pre>	-
aliyun_logs_{key}_machine group	Optional. This variable specifies the machine group of the application. By default, the machine group is the one where your Logtail is located.	<pre>- name: aliyun_logs_ca talina_machinegroup my-machine-group</pre>	-

- Scenario 1: Collect logs from multiple applications and store them in the same Logstore

In this scenario, you can set the `aliyun_logs_{key}_logstore` variable. The following example shows how to collect standard output logs from two applications and store them in `stdout-logstore`.

Set the following environment variables for Application 1:

```
##### Set environment variables #####
- name: aliyun_logs_app1-stdout
  value: stdout
- name: aliyun_logs_app1-stdout_logstore
  value: stdout-logstore
```

Set the following environment variables for Application 2:

```
##### Set environment variables #####
- name: aliyun_logs_app2-stdout
  value: stdout
- name: aliyun_logs_app2-stdout_logstore
  value: stdout-logstore
```

- Scenario 2: Collect logs from different applications and store them separately in different projects

In this scenario, perform the following steps:

- i. Create a machine group in each project and set the machine group ID in the following format: k8s-group-{cluster-id}, where {cluster-id} is the ID of the cluster. You can customize machine group names.
- ii. Specify the project, Logstore, and machine group in the environment variables for each application.

```
##### Set environment variables #####  
- name: aliyun_logs_app1-stdout  
  value: stdout  
- name: aliyun_logs_app1-stdout_project  
  value: app1-project  
- name: aliyun_logs_app1-stdout_logstore  
  value: app1-logstore  
- name: aliyun_logs_app1-stdout_machinegroup  
  value: app1-machine-group
```

View log data

To view log data, perform the following steps:

1. Log on to the Log Service console. For more information, see [Activate Log Service](#).
2. Select the target project. By default, the project ID is k8s-log-{Cluster ID}.
3. In the list of Logstores, find the target Logstores and click Search in the Log Search column for each Logstore. In this example, the Logstores are log-stdout and log-varlog.
4. On the Raw Logs tab, you can view raw logs in log-stdout and log-varlog.

4.Auto Scaling (ESS)

4.1. What is ESS?

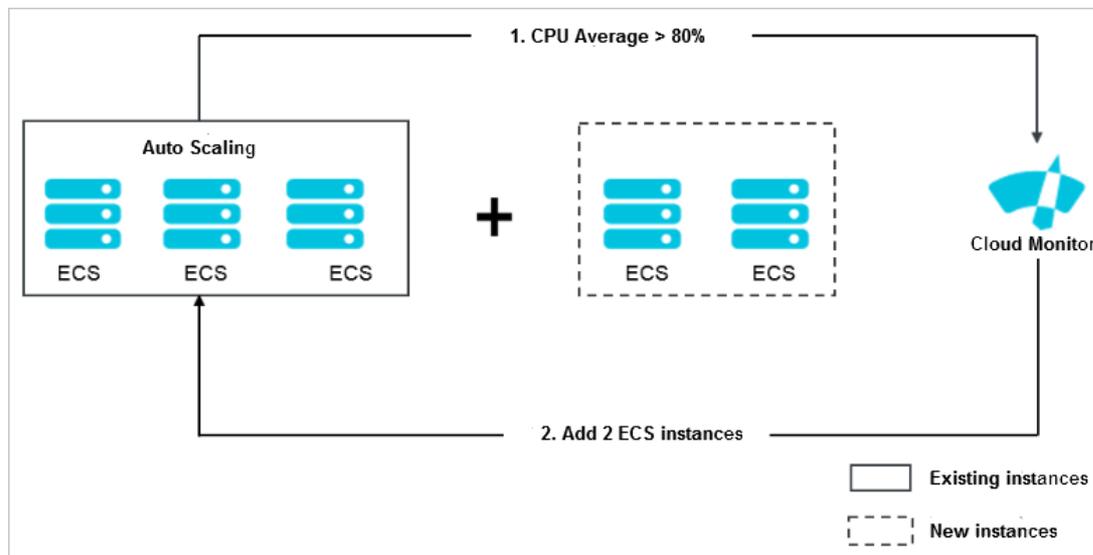
Auto Scaling (ESS) is a management service that automatically adjusts your elastic computing resources based on your business needs and policies.

When business loads increase, ESS automatically adds ECS instances based on the scaling rules that you configured to ensure sufficient computing capabilities. When business loads decrease, ESS automatically removes ECS instances to save costs.

ESS provides the following features:

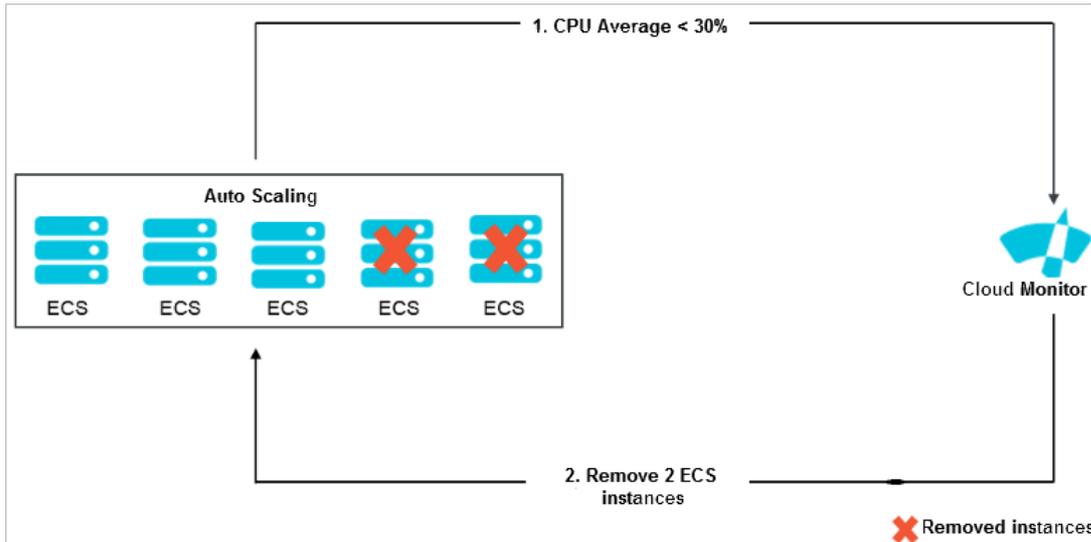
- Scale-out

When business loads surge above normal loads, ESS automatically increases underlying resources. This helps maintain access speed and ensure that resources are not overloaded. For example, if the CPU utilization of ECS instances exceeds 80%, ESS scales out ECS resources based on the rules that you configured. During the scale-out event, ESS automatically creates and adds ECS instances to the scaling group, and adds the new instances to the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB for RDS instances. The following figure shows the implementation of a scale-out event.



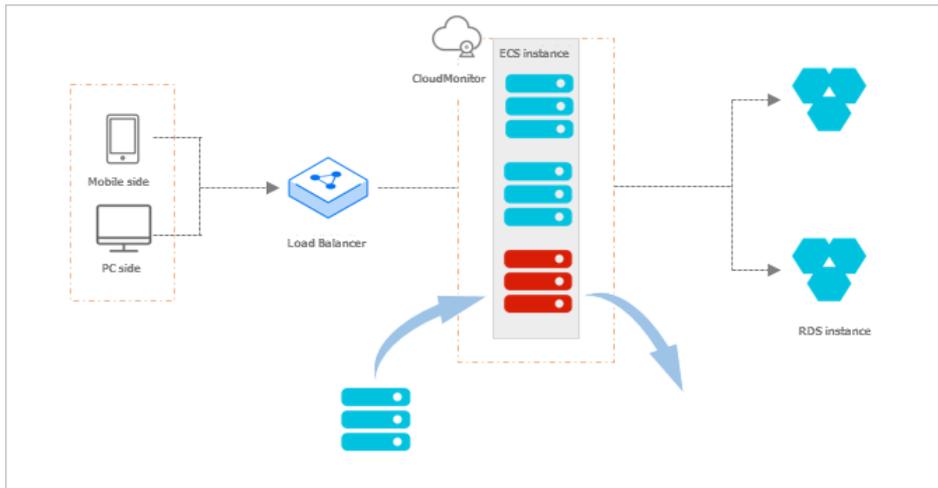
- Scale-in

When your business loads decrease, ESS automatically releases underlying resources to prevent resource wastes and reduce costs. For example, if the CPU utilization of ECS instances in a scaling group is less than 30%, ESS automatically scales in ECS instances based on the scaling rules that you specified. During the scale-in event, ESS removes ECS instances from the scaling group and also from the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB for RDS instances. The following figure shows the implementation of a scale-in event.



• Elastic recovery

If an ECS instance in a scaling group is not in the Running state, ESS considers the instance to be unhealthy. If an ECS instance is considered unhealthy, ESS automatically releases the instance and creates a new one. This process is called elastic recovery. It ensures that the number of healthy ECS instances in a scaling group will not fall below the minimum number of ECS instances that you specified for the scaling group. The following figure shows the implementation of elastic recovery.



4.2. Notes

4.2.1. Precautions

This topic describes the precautions when you use Auto Scaling (ESS).

Scaling rules

ESS uses scaling rules to scale ECS instances in a scaling group based on the minimum and maximum numbers of ECS instances specified for the scaling group. Assume that a scaling group can contain up to 45 ECS instances. If you configure a scaling rule to increase the number of ECS instances in the scaling group to 50, ESS only increases the number of ECS instances to 45 at most.

Scaling activities

- Only one scaling activity can be executed at a time in a scaling group.
- An ongoing scaling activity cannot be terminated. For example, if a scaling activity is being executed to

create 20 ECS instances but only five have been created, you cannot forcibly terminate the scaling activity.

- If some ECS instances fail to be added to a scaling group during a scaling activity, ESS considers that the scaling activity is complete without trying to add the failed instances to the scaling group. ESS rolls back the ECS instances that fails to be added but not the scaling activity. For example, if ESS has created 20 ECS instances for a scaling group, and 19 of the instances are added to SLB instances, only the one ECS instance that failed to be added is automatically released.

Cooldown period

- During the cooldown period, if you manually execute a scaling task, such as a scaling rule or scheduled task, the task is immediately executed without waiting for the cooldown period to expire.
- The cooldown period starts after the last ECS instance is added to or removed from a scaling group during a scaling activity.

4.2.2. Manual intervention

If you manually intervene with Auto Scaling (ESS) operations, ESS will process the intervention accordingly.

ESS does not prevent you from performing manual intervention, such as deleting automatically created ECS instances in the ECS console. The following table describes how ESS processes manual intervention.

Resource	Manual intervention type	Processing method
ECS	A user deletes an ECS instance from a scaling group by using the ECS console or calling API operations.	ESS performs health checks to determine whether the ECS instance is unhealthy. If the instance is unhealthy, ESS removes it from the scaling group. The internal IP address of the ECS instance is not automatically deleted from the whitelist of the associated ApsaraDB for RDS (RDS) instance. After the ECS instance is removed, if the number of instances (Total Capacity) in the scaling group is less than the minimum number of instances (MinSize), ESS automatically creates and adds ECS instances to the group until the number of instances is equal to the minimum number of instances (MinSize).
ECS	A user revokes the ECS API permissions granted to ESS.	ESS rejects all scaling activity requests.
SLB	A user manually removes an ECS instance from an SLB instance by using the SLB console or calling API operations.	ESS does not automatically detect this action or handle such exceptions. The ECS instance remains in the scaling group. If this instance is selected based on the removal policy during a scale-in event, the instance is released.
SLB	A user manually deletes an SLB instance or disables the health check feature for an SLB instance by using the SLB console or calling API operations.	ESS does not add ECS instances to scaling groups that are associated with this SLB instance. Scaling tasks can trigger scaling rules to remove ECS instances from the scaling groups. ECS instances that are considered unhealthy through the health check feature are also removed from the scaling groups.
SLB	An SLB instance is unavailable because of system-related reasons.	All scaling activities fail except for instance removal tasks that are manually executed.
SLB	A user revokes the SLB API permissions granted to ESS.	ESS rejects all scaling activity requests for scaling groups that are associated with SLB instances.

Resource	Manual intervention type	Processing method
RDS	A user manually removes the IP address of an ECS instance from the whitelist of the associated RDS instance by using the RDS console or calling API operations.	ESS does not automatically detect this action or handle such exceptions. The ECS instance remains in the scaling group. If this instance is selected based on the removal policy during a scale-in event, the instance is released.
RDS	A user manually deletes an RDS instance by using the RDS console or calling API operations.	ESS does not add ECS instances that are associated with this RDS instance to scaling groups. Scaling tasks can trigger scaling rules to remove ECS instances from the scaling groups. ECS instances that are considered unhealthy through the health check feature are also removed from the scaling groups.
RDS	An RDS instance is unavailable because of system-related reasons.	All scaling activities fail except for instance removal tasks that are manually executed.
RDS	A user revokes the RDS API permissions granted to ESS.	ESS rejects all scaling activity requests for the scaling groups associated with RDS instances.

4.2.3. Limits

This topic describes the limits of ESS.

- ECS instances that are created by ESS cannot be automatically added to whitelists of ApsaraDB for Memcache instances. For more information about ApsaraDB for Memcache instances, see *ApsaraDB for Memcache Product Introduction*.
- ESS does not support vertical scaling. It can only scale the number of ECS instances. The CPU, memory, and bandwidth configurations of ECS instances cannot be automatically adjusted.
- The following table describes the quantity limits that are applied to a scaling group.

Item	Quota
Scaling configuration	You can create a maximum of 10 scaling configurations for a scaling group.
Scaling rule	You can create a maximum of 50 scaling rules for a scaling group.
ECS instance	A scaling group can contain a maximum of 1,000 ECS instances.

4.2.4. Scaling group status

This topic describes the states of a scaling group in the console and in an API operation.

State in the console	State in an API operation
Creating	Inactive
Created	Inactive
Enabling	Inactive
Enabled	Active
Disabling	Inactive
Disabled	Inactive

State in the console	State in an API operation
Deleting	Deleting

4.2.5. Scaling activity process

Before you use ESS, you must understand the processes related to scaling activities.

Automatic scaling of a scaling group

- Automatic scale-out
 - i. Check the health status and boundary conditions of the scaling group.
 - ii. Assign the activity ID and execute the scaling activity.
 - iii. Create ECS instances.
 - iv. Modify Total Capacity.
 - v. Assign IP addresses to the created ECS instances.
 - vi. Add the ECS instances to the whitelist of the associated ApsaraDB for RDS instance.
 - vii. Start ECS instances.
 - viii. Associate the ECS instances with an SLB instance and set the weight to the SLB weight value that is specified when the scaling configuration is created.
 - ix. After the scaling activity is complete, the cooldown period starts.
- Automatic scale-in
 - i. Check the health status and boundary conditions of the scaling group.
 - ii. Assign the activity ID and execute the scaling activity.
 - iii. Remove ECS instances from the associated SLB instance.
 - iv. Stop the ECS instances.
 - v. Remove the ECS instances from the whitelist of the associated ApsaraDB for RDS instance.
 - vi. Release the ECS instances.
 - vii. Modify Total Capacity.
 - viii. After the scaling activity is complete, the cooldown period starts.

Manually add or remove existing ECS instances

- Manually add instances
 - i. Check the health status and boundary conditions of the scaling group, and check the status and type of ECS instances.
 - ii. Assign the activity ID and execute the scaling activity.
 - iii. Add ECS instances.
 - iv. Modify Total Capacity.
 - v. Add the ECS instances to the whitelist of the associated ApsaraDB for RDS instance.
 - vi. Associate the ECS instances with an SLB instance and set the weight to the SLB weight value that is specified in the active scaling configuration.

 **Note** If you want to manually add an instance to a scaling group, its instance type must be the same as that specified in the active scaling configuration of the scaling group. Therefore, you must set the weight to the SLB weight value that is specified in the active scaling configuration.

- vii. After the scaling activity is complete, the cooldown period starts.
- Manually remove instances

- i. Check the health status and boundary conditions of the scaling group.
- ii. Assign the activity ID and execute the scaling activity.
- iii. SLB stops forwarding traffic to ECS instances.
- iv. Remove the ECS instances from SLB after 60 seconds.
- v. Remove the ECS instances from the whitelist of the associated ApsaraDB for RDS instance.
- vi. Modify Total Capacity.
- vii. Remove the ECS instances from the scaling group.
- viii. After the scaling activity is complete, the cooldown period starts.

4.2.6. Remove unhealthy ECS instances

Before you use ESS, you must understand information about the removal of unhealthy ECS instances.

After an ECS instance is added to a scaling group, ESS checks the status of the instance on a regular basis. If the ECS instance is not in the Running state, ESS removes the ECS instance from the scaling group. The removal method depends on how the ECS instance is added:

- If an ECS instance is automatically created, ESS immediately removes and releases it.
- If an ECS instance is manually added, ESS immediately removes it, but does not stop or release it.

The removal of unhealthy ECS instances is not limited by the MinSize value. After the unhealthy ECS instances are removed, the number of ECS instances (Total Capacity) may fall below the MinSize value. In this case, ESS automatically creates ECS instances based on the difference between the actual instance number and MinSize value to ensure that the total number of ECS instances is equal to the MinSize value.

4.2.7. Instance rollback after a failed scaling activity

Before you use ESS, you must understand the mechanism of instance rollback after a failed scaling activity.

If some ECS instances fail to be added to a scaling group during a scaling activity, ESS considers that the scaling activity is complete without trying to add the failed instances to the scaling group. ESS rolls back ECS instances, not the scaling activity.

For example, if a scaling group has created 20 ECS instances, and 19 of the instances are added to SLB instances, only the one ECS instance that failed to be added is automatically released.

4.2.8. Instance lifecycle management

Before you use Auto Scaling (ESS), you must understand concepts related to the instance lifecycle.

Automatically created ECS instances

ECS instances are automatically created by ESS based on user-defined scaling configurations and rules.

ESS manages the entire lifecycle of automatically created ECS instances. ESS creates ECS instances during scale-out events, and stops and releases them during scale-in events.

Manually added ECS instances

ECS instances are manually added to a scaling group.

ESS does not manage the entire lifecycle of manually added ECS instances. These instances are not automatically created by ESS, but are manually added by a user to a scaling group. If the ECS instances are manually or automatically removed from the scaling group, ESS removes the instances but does not stop or release them.

Instance status

An ECS instance in a scaling group goes through the following states during its lifecycle:

- **Pending:** The ECS instance is being added to the scaling group. The instance is being created, added to an SLB instance, or added to the whitelist of the associated ApsaraDB RDS instance.

- **InService:** The ECS instance is added to the scaling group and is providing services normally.
- **Removing:** The ECS instance is being removed from the scaling group.

Instance health status

An ECS instance in a scaling group has the following health states:

- **Healthy**
- **Unhealthy**

If an ECS instance is not in the Running state, ESS considers the instance to be unhealthy and automatically removes it from the scaling group.

- ESS stops and releases automatically created ECS instances.
- ESS does not stop or release manually added ECS instances.

4.3. Quick start

4.3.1. Overview

This topic describes how to create a scaling group and how to add or remove ECS instances.

You can perform the following steps to create a scaling group, and add or remove ECS instances.

1. **Create a scaling group**

Set the parameters for the scaling group, such as the Maximum Capacity and Minimum Capacity of ECS instances.

2. **Create a scaling configuration**

Set the parameters for the scaling configuration, such as Instance Type and Image.

3. **Enable a scaling group**

Enable the scaling group after creating the scaling configuration.

4. **Create a scaling rule**

Specify how to add or remove ECS instances. For example, add an ECS instance to a scaling group.

5. **Create a scheduled task**

Create scheduled tasks to add or remove instances at a specified time point. Auto Scaling executes the scheduled tasks and scaling rules at the specified time. For example, Auto Scaling can trigger a task to execute a specific scaling rule at 08:00 everyday.

4.3.2. Log on to the Auto Scaling console

This topic describes how to log on to the Auto Scaling console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Elastic Computing > Auto Scaling**.

4.3.3. Create a scaling group

This topic describes how to create a scaling group. A scaling group is a group of ECS instances that is dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group.

Prerequisites

- A VPC and a VSwitch are created. For more information, see *Create a VPC and a VSwitch in VPC User Guide*.
- To associate a scaling group with SLB instances, make sure that the following requirements are met:
 - You have one or more SLB instances in the **Running** state.
 - The SLB instances and the scaling group are in the same organization, resource set, and region.
- To associate a scaling group with ApsaraDB for RDS (RDS) instances, make sure that the following requirements are met:
 - You have one or more RDS instances in the **Running** state.
 - The RDS instances and the scaling group are in the same organization, resource set, and region.

Procedure

1. **Log on to the Auto Scaling console.**
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Click **Create Scaling Group**.
4. Configure parameters for the scaling group.

Parameter	Required	Description
Scaling Group	Yes	The name of the scaling group. It must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Organization/Resource Group	Yes	The organization and resource set to which the scaling group belongs.
Maximum Capacity	Yes	The maximum number of instances that a scaling group can contain. This helps control costs within an expected amount. Valid values: 0 to 1000.

Parameter	Required	Description
Minimum Capacity	Yes	<p>The minimum number of instances that a scaling group must contain. Set the value based on business needs to ensure service availability. When the scaling group is enabled, Auto Scaling automatically creates this number of ECS instances.</p> <p>Valid values: 0 to 1000.</p>
Cooldown (seconds)	Yes	<p>The period during which Auto Scaling cannot execute any new scaling activities. This occurs after the scaling group executes a successful scaling activity. During the cooldown period, Auto Scaling rejects all scaling activity requests triggered by event-triggered tasks from Cloud Monitor. However, scaling activities triggered by other types of tasks, such as manually triggered tasks and scheduled tasks are not limited by the cooldown period. These tasks are immediately executed.</p> <p>The value must be an integer that is greater than or equal to zero. Unit: seconds.</p>
Scale-In Policy	No	<p>The policy for automatically removing ECS instances from the scaling group. This parameter contains two steps.</p> <p>Valid values for the first step:</p> <ul style="list-style-type: none"> ◦ Oldest Instance ◦ Newest Instances ◦ Instance with Oldest Scaling Configuration <p>Valid values for the second step:</p> <ul style="list-style-type: none"> ◦ None ◦ Oldest Instance ◦ Newest Instances <p>For example, you can select Instance with Oldest Scaling Configuration for the first step and select Oldest Instance for the second step. This indicates that Auto Scaling filters ECS instances to find the ones that were created based on the earliest scaling configuration, and then filters instances to find the ones that were added to the scaling group at the earliest point in time.</p>
Region/VPC	Yes	The region and VPC to which the scaling group belongs.
VSwitch	Yes	The ID of the VSwitch with which the scaling group is associated.

Parameter	Required	Description
Associate SLB Instances	No	<p>After you associate SLB instances with the scaling group, ECS instances that are added to the scaling group will be automatically added as SLB backend servers. You can specify a server group for the ECS instances. The following section describes two available server groups:</p> <ul style="list-style-type: none"> Default server group: the group of ECS instances that are used to receive requests. If the listener is not configured with a VServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group. VServer group: If you want to distribute different requests to different backend servers or configure domain name- or URL-based routing methods, you can use VServer groups.
Associate RDS Instances	No	<p>After you associate RDS instances with the scaling group, the internal IP addresses of ECS instances that are added to the scaling group will be automatically added to the whitelists of the RDS instances to allow internal communication.</p>

5. Click OK.

Result

The created scaling group is displayed in the scaling group list but is in the **Disabled** state. To enable the scaling group, you must create scaling configurations. For more information, see [Create a scaling configuration](#).

4.3.4. Create a scaling configuration

This topic describes how to create a scaling configuration for a scaling group.

Prerequisites

At least one security group is available. If you do not have any security groups, create a security group. For more information, see [Create a security group](#) in *ECS User Guide*.

Context

You can create only a limited number of scaling configurations for a scaling group. For more information, see the [Limits](#) topic in *Auto Scaling Product Introduction*.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. Choose **Create > Create Scaling Configuration**.
6. Configure parameters for the scaling configuration.

Section	Parameter	Required	Description
Region	Region	Yes	The region where the ECS instance is located.

Region Section	Parameter	Required	Description
	Zone	Yes	The zone where the ECS instance is located.
Security Group	Security Group	Yes	The security group to which the ECS instance belongs.
Instance	Instance Family	Yes	The instance family to which the ECS instance belongs.
	Instance Type	Yes	The instance type of the ECS instance.
Image	Image Type	Yes	<ul style="list-style-type: none"> ◦ Public Image: Public images provided by Alibaba Cloud are fully licensed to offer a secure and stable operating environment for applications on ECS instances. ◦ Custom Image: You can create custom images to install software or deploy projects that have special requirements.
Storage	System Disk	Yes	Specify the category and size of the system disk. The operating system is installed on the system disk. You can select Ultra Disk or Standard SSD .
	Data Disk	No	Specify the category and size of the data disk. You can select Ultra Disk or Standard SSD . You can add a maximum of 16 data disks. The maximum capacity of each data disk is 32 TiB. You can set Release with Instance and Encrypt for each data disk.
Password	Set Password	Yes	<p>Select when to set password. You can select Now or Later.</p> <p>If you select Later, you can use the Change Password feature in the console to set the password. For more information, see the Change Password topic in <i>ECS User Guide</i>.</p>

Section	Parameter	Required	Description
	Logon Password	No	The password used to log on to the ECS instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: digits, uppercase letters, lowercase letters, and special characters.  Note The password is used to log on to the operating system and is not the VNC password.
	Confirm Password	No	Enter the password again.
Deployment Set	Deployment Set	No	The deployment set to which the instance belongs.
Instance Name	Configuration Name	No	The name of the scaling configuration.
	Instance Name	No	The name of the ECS instance.
User Data	User Data	No	Windows supports two formats: Bat and Powershell. Before you perform Base64 encoding, make sure to include <code>[bat]</code> or <code>[powershell]</code> as the first line. You can run shell scripts for Linux ECS instances.
Quantity	Quantity	No	The number of instances to purchase.

7. Click **Submit**.

Result

After the scaling configuration is created, it is in the **Disabled** state and is displayed in your scaling configuration list. To automatically create ECS instances, you must apply a scaling configuration. For more information, see [Apply a scaling configuration](#).

4.3.5. Enable a scaling group

This topic describes how to enable a scaling group. You can enable a scaling group to trigger scaling activities.

Prerequisites

- The scaling group is in the **Disabled** state.
- The scaling group has a scaling configuration that is in the **Enable** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click **Enable** in the **Actions** column.

4. Click **OK**.

Result

In the **Status** column, the state of the scaling group is changed from **Disabled** to **Enable**.

4.3.6. Create a scaling rule

This topic describes how to create a scaling rule. You can create scaling rules to add or remove ECS instances. For example, you can add an ECS instance to a scaling group.

Context

- You can create only a limited number of scaling rules for a scaling group. For more information, see the **Limits** topic in *Auto Scaling Product Introduction*.
- After a scaling rule is executed, the resulting number of ECS instances in the scaling group may fall outside of the specified range. In this case, Auto Scaling automatically adjusts the number of ECS instances to ensure that the number of ECS instances in the scaling group is within the specified range.

Procedure

1. **Log on to the Auto Scaling console.**
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. Click **Create Scaling Rule**.
6. Configure parameters for the scaling rule.

Parameter	Required	Description
Rule Name	Yes	The name of the scaling rule. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Scaling Activity	Yes	The operation that is performed when the scaling rule is triggered. The operations include: <ul style="list-style-type: none"> ◦ Change to N instances: After the scaling rule is executed, the number of instances in the scaling group is changed to N. ◦ Add N instances: After the scaling rule is executed, N instances are added to the scaling group. ◦ Remove N instances: After the scaling rule is executed, N instances are removed from the scaling group.
Default Cooldown (Seconds)	No	The cooldown period. If this parameter is not specified, the default value is used.

7. Click **OK**.

4.3.7. Create a scheduled task

This topic describes how to create a scheduled task to scale computing resources in response to predictable business changes in the future. Scheduled tasks enable the system to automatically obtain sufficient computing resources before business peaks and release idle computing resources after the peaks.

Context

A scheduled task is preconfigured to execute the specified scaling rule at the specified time in the future. When the specified time arrives, the scheduled task automatically scales computing resources. This allows you to reduce costs and also meet business requirements. You can also specify the recurrence for scheduled tasks if business changes are regular.

If multiple scheduled tasks need to be executed in one minute, Auto Scaling executes the most recently created scheduled task.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **Create Scheduled Task**.
5. In the dialog box that appears, configure parameters for the scheduled task.

Parameter	Required	Description
Task Name	Yes	The name of the scheduled task. The name must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	Yes	The description of the scheduled task.
Organization/Resource Group	Yes	The organization and resource set to which the scheduled task belongs.
Start Time	Yes	The time when the scheduled task is executed.
Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Retry Expiry Time	No	The period of time during which the system retries to execute the scheduled task. Unit: seconds. If a scaling activity fails to be executed at the specified time, Auto Scaling will execute the scheduled task again within the period of time that is specified by the Retry Expiry Time parameter.
Recurrence Settings	No	Specifies whether to execute the scheduled task repeatedly. Select Recurrence Settings and set the Recurrence and Expire parameters. The recurrence options include Daily , Weekly , and Monthly .

6. Click **OK**.

Result

The scheduled task that you created is displayed in the scheduled task list.

4.3.8. Create an event-triggered task

This topic describes how to create an event-triggered task associated with monitoring metrics in response to emergent or unpredictable business changes. After you create and enable an event-triggered task, Auto Scaling collects data for the specified metric in real time, and triggers an alert when the specified condition is met. Then, Auto Scaling executes the scaling rule to dynamically scale ECS instances in the scaling group.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the left-side navigation pane, choose **Event-driven Tasks > Alert Tasks**.

3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **Create Alert**.
5. In the dialog box that appears, configure parameters for the event-triggered task.

Parameter	Required	Description
Task Name	Yes	The name of the event-triggered task. It must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	No	The description of the event-triggered task.
Organization/Resource Group	Yes	The organization and resource set in which to create the event-triggered task.
Monitoring Metrics/Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Monitoring Type	Yes	System-Level Monitoring is selected by default.
Monitoring Metrics	Yes	The metrics that you want to monitor. Valid values: <ul style="list-style-type: none"> ◦ Average CPU Utilization ◦ Memory Usage ◦ Outbound Traffic ◦ Inbound Traffic
Monitoring Period	Yes	The period during which data is aggregated and analyzed. The shorter the period, the higher the frequency that the alert is triggered. Unit: minutes. Valid values: <ul style="list-style-type: none"> ◦ 1 ◦ 2 ◦ 5 ◦ 15
Statistic	Yes	The rule that determines whether to trigger an alert. Select Average , Max Capacity , or Min Capacity , and specify a threshold value. For example, alerts are triggered when the CPU utilization exceeds 80%: <ul style="list-style-type: none"> ◦ Average: Alerts are triggered when the average CPU utilization of all ECS instances in the scaling group exceeds 80%. ◦ Max Capacity: Alerts are triggered when the highest CPU utilization among the ECS instances in the scaling group exceeds 80%. ◦ Min Capacity: Alerts are triggered when the lowest CPU utilization among the ECS instances in the scaling group exceeds 80%.
Alert after occurrences	Yes	The consecutive number of times that the threshold must be exceeded before the alert is triggered. Valid values: <ul style="list-style-type: none"> ◦ 1 ◦ 2 ◦ 3 ◦ 5

- Click **OK**.

4.4. Scaling groups

4.4.1. Create a scaling group

This topic describes how to create a scaling group. A scaling group is a group of ECS instances that is dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group.

Prerequisites

- A VPC and a VSwitch are created. For more information, see *Create a VPC and a VSwitch* in *VPC User Guide*.
- To associate a scaling group with SLB instances, make sure that the following requirements are met:
 - You have one or more SLB instances in the **Running** state.
 - The SLB instances and the scaling group are in the same organization, resource set, and region.
- To associate a scaling group with ApsaraDB for RDS (RDS) instances, make sure that the following requirements are met:
 - You have one or more RDS instances in the **Running** state.
 - The RDS instances and the scaling group are in the same organization, resource set, and region.

Procedure

- [Log on to the Auto Scaling console](#).
- In the top navigation bar, select an organization, a resource set, and a region.
- Click **Create Scaling Group**.
- Configure parameters for the scaling group.

Parameter	Required	Description
Scaling Group	Yes	The name of the scaling group. It must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Organization/Resource Group	Yes	The organization and resource set to which the scaling group belongs.
Maximum Capacity	Yes	The maximum number of instances that a scaling group can contain. This helps control costs within an expected amount. Valid values: 0 to 1000.
Minimum Capacity	Yes	The minimum number of instances that a scaling group must contain. Set the value based on business needs to ensure service availability. When the scaling group is enabled, Auto Scaling automatically creates this number of ECS instances. Valid values: 0 to 1000.

Parameter	Required	Description
Cooldown (seconds)	Yes	<p>The period during which Auto Scaling cannot execute any new scaling activities. This occurs after the scaling group executes a successful scaling activity. During the cooldown period, Auto Scaling rejects all scaling activity requests triggered by event-triggered tasks from Cloud Monitor. However, scaling activities triggered by other types of tasks, such as manually triggered tasks and scheduled tasks are not limited by the cooldown period. These tasks are immediately executed.</p> <p>The value must be an integer that is greater than or equal to zero. Unit: seconds.</p>
Scale-In Policy	No	<p>The policy for automatically removing ECS instances from the scaling group. This parameter contains two steps.</p> <p>Valid values for the first step:</p> <ul style="list-style-type: none"> ◦ Oldest Instance ◦ Newest Instances ◦ Instance with Oldest Scaling Configuration <p>Valid values for the second step:</p> <ul style="list-style-type: none"> ◦ None ◦ Oldest Instance ◦ Newest Instances <p>For example, you can select Instance with Oldest Scaling Configuration for the first step and select Oldest Instance for the second step. This indicates that Auto Scaling filters ECS instances to find the ones that were created based on the earliest scaling configuration, and then filters instances to find the ones that were added to the scaling group at the earliest point in time.</p>
Region/VPC	Yes	The region and VPC to which the scaling group belongs.
VSwitch	Yes	The ID of the VSwitch with which the scaling group is associated.
Associate SLB Instances	No	<p>After you associate SLB instances with the scaling group, ECS instances that are added to the scaling group will be automatically added as SLB backend servers. You can specify a server group for the ECS instances. The following section describes two available server groups:</p> <ul style="list-style-type: none"> ◦ Default server group: the group of ECS instances that are used to receive requests. If the listener is not configured with a VServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group. ◦ VServer group: If you want to distribute different requests to different backend servers or configure domain name- or URL-based routing methods, you can use VServer groups.

Parameter	Required	Description
Associate RDS Instances	No	After you associate RDS instances with the scaling group, the internal IP addresses of ECS instances that are added to the scaling group will be automatically added to the whitelists of the RDS instances to allow internal communication.

5. Click **OK**.

Result

The created scaling group is displayed in the scaling group list but is in the **Disabled** state. To enable the scaling group, you must create scaling configurations. For more information, see [Create a scaling configuration](#).

4.4.2. Enable a scaling group

This topic describes how to enable a scaling group. You can enable a scaling group to trigger scaling activities.

Prerequisites

- The scaling group is in the **Disabled** state.
- The scaling group has a scaling configuration that is in the **Enable** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click **Enable** in the **Actions** column.
4. Click **OK**.

Result

In the **Status** column, the state of the scaling group is changed from **Disabled** to **Enable**.

4.4.3. View scaling groups

This topic describes how to view the scaling group list and the details of a specific scaling group.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3. Select a filter option, enter the corresponding information, and then click **Search**.

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

Option	Description
Scaling Group	Enter a scaling group name to search for the scaling group.
Scaling Group ID	Enter a scaling group ID to search for the scaling group.

4. Click the name of the scaling group in the **Scaling Group Name/ID** column.
5. View the details of the specified scaling group.

Parameter	Description
Basic Information	The configurations of the scaling group, such as the scaling group ID, scaling group name, total instances, minimum number of instances, maximum number of instances, and scale-in policy.
ECS Instances	The details of ECS instances, such as the list of automatically created ECS instances, the list of manually added ECS instances, and the number of ECS instances that are in service.
Scaling Activities	All the scaling activities that have been executed in the scaling group.
Scaling Configuration	The information of scaling configurations in the scaling group.
Scaling Rules	The information of scaling rules.

4.4.4. Modify a scaling group

This topic describes how to modify a scaling group. You can modify the parameters of a specific scaling group, such as the minimum and maximum numbers of ECS instances.

Context

After you modify the minimum or maximum number of ECS instances that a scaling group can have, if the number of instances in the scaling group is outside this range, Auto Scaling automatically creates or removes ECS instances until the number of instances are within the range.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click **Edit** in the **Actions** column.
4. Modify the parameters of the scaling group. You can modify the scaling configuration and other parameters, but not the organization and resource set. For more information about other parameters, see [Create a scaling group.](#)
5. Click **OK**.

4.4.5. Disable a scaling group

This topic describes how to disable a scaling group.

Prerequisites

- The scaling group does not have scaling activities in progress.
- The scaling group is in the **Enable** state.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click **Disable** in the **Actions** column.
4. Click **OK**.

Result

The status of the scaling group is changed from **Enable** to **Disabled** in the **Status** column.

4.4.6. Delete a scaling group

This topic describes how to delete a scaling group. When you delete a scaling group, Auto Scaling removes and releases ECS instances that are automatically created, removes ECS instances that are manually added, and deletes the scaling configurations and rules in the scaling group. However, the scheduled tasks and event-triggered tasks that are associated with the scaling group are not deleted.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click **Delete** in the **Actions** column.
4. Click **OK**.

4.4.7. View ECS instances

You can view all ECS instances in a scaling group and their states.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. View the details of ECS instances.

Category	Description
Automatically created ECS instances	The ECS instances that are automatically created based on the active scaling configuration when the scaling rule is triggered.
Manually added ECS instances	The ECS instances that are manually added to the specified scaling group.
States of ECS instances in the scaling group	<p>The number of ECS instances in various states. The states include:</p> <ul style="list-style-type: none"> ◦ Total: all ECS instances in the scaling group ◦ In Service: the ECS instances that are in normal use ◦ On Standby: the ECS instances that are on standby ◦ Protected: the ECS instances that are protected ◦ Disabled: the ECS instances that are stopped ◦ Adding: the ECS instances that are being added to the scaling group ◦ Removing: the ECS instances that are being removed from the scaling group

4.4.8. Put an ECS instance into the Standby state

This topic describes how to put an ECS instance into the Standby state. Auto Scaling does not perform health checks on or release ECS instances in the Standby state.

Context

After an ECS instance is put into the Standby state:

- The ECS instance stays in the Standby state until you change its status.
- Auto Scaling stops managing the lifecycle of the ECS instance. You must manually manage the lifecycle of the ECS instance.
- If a scale-in event is triggered, Auto Scaling will not remove the ECS instance.
- When the ECS instance is stopped or restarted, its health check status is not affected.
- To release the ECS instance, you must first remove it from the scaling group.
- If you delete the scaling group, the ECS instance is automatically put out of the Standby state and is released.
- You can also perform other operations on the ECS instance, such as stopping, restarting, changing the instance type of, or changing the operating system of the ECS instance.

Procedure

1. **Log on to the Auto Scaling console.**
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Select the source of the ECS instance.
 - To select an automatically created ECS instance, click the **Auto Created** tab.
 - To select a manually added ECS instance, click the **Manually Added** tab.
6. Find the target ECS instance and choose **Actions > Switch to Standby** in the **Actions** column.
7. Click **OK**.

4.4.9. Remove an ECS instance from the Standby state

This topic describes how to remove an ECS instance from the Standby state. You can remove an instance from the Standby state to reuse it.

Context

After an ECS instance is removed from the Standby state:

- The ECS instance enters the In Service state.
- When the ECS instance is stopped or restarted, its health status is updated.
- Auto Scaling continues to manage the lifecycle of the ECS instance, and can remove the ECS instance from the scaling group during a scale-in event.

Procedure

1. **Log on to the Auto Scaling console.**
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Select the source of the ECS instance.
 - To select an automatically created ECS instance, click the **Auto Created** tab.
 - To select a manually added ECS instance, click the **Manually Added** tab.
6. Find the target ECS instance and choose **Actions > Move Out Of Standby** in the **Actions** column.
7. Click **OK**.

4.4.10. Put an ECS instance into the Protected state

This topic describes how to put an ECS instance into the Protected state. Auto Scaling does not perform health checks on or release ECS instances that are in the Protected state.

Context

After an ECS instance is put into the Protected state:

- The ECS instance stays in the Protected state until you change its status.
- If a scale-in event is triggered, Auto Scaling will not remove the ECS instance. To release the ECS instance, you must remove the ECS instance from the Protected state and then remove it from the scaling group.
- When the ECS instance is stopped or restarted, its health check status is not affected.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Select the source of the ECS instance.
 - To select an automatically created ECS instance, click the **Auto Created** tab.
 - To select a manually added ECS instance, click the **Manually Added** tab.
6. Find the target ECS instance and choose **Actions > Switch to Protection** in the **Actions** column.
7. Click **OK**.

4.4.11. Remove an ECS instance from the Protected state

This topic describes how to remove an ECS instance from the Protected state. After an ECS instance is removed from the Protected state, Auto Scaling continues to manage the lifecycle of the ECS instance.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Select the source of the ECS instance.
 - To select an automatically created ECS instance, click the **Auto Created** tab.
 - To select a manually added ECS instance, click the **Manually Added** tab.
6. Find the target ECS instance and choose **Actions > Move Out Of Protection** in the **Actions** column.
7. Click **OK**.

4.5. Scaling configurations

4.5.1. Create a scaling configuration

This topic describes how to create a scaling configuration for a scaling group.

Prerequisites

At least one security group is available. If you do not have any security groups, create a security group. For more information, see [Create a security group](#) in *ECS User Guide*.

Context

You can create only a limited number of scaling configurations for a scaling group. For more information, see

the Limits topic in *Auto Scaling Product Introduction*.

Procedure

1. Log on to the Auto Scaling console.
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. Choose **Create > Create Scaling Configuration**.
6. Configure parameters for the scaling configuration.

Section	Parameter	Required	Description
Region	Region	Yes	The region where the ECS instance is located.
	Zone	Yes	The zone where the ECS instance is located.
Security Group	Security Group	Yes	The security group to which the ECS instance belongs.
Instance	Instance Family	Yes	The instance family to which the ECS instance belongs.
	Instance Type	Yes	The instance type of the ECS instance.
Image	Image Type	Yes	<ul style="list-style-type: none"> ◦ Public Image: Public images provided by Alibaba Cloud are fully licensed to offer a secure and stable operating environment for applications on ECS instances. ◦ Custom Image: You can create custom images to install software or deploy projects that have special requirements.
Storage	System Disk	Yes	Specify the category and size of the system disk. The operating system is installed on the system disk. You can select Ultra Disk or Standard SSD .
	Data Disk	No	Specify the category and size of the data disk. You can select Ultra Disk or Standard SSD . You can add a maximum of 16 data disks. The maximum capacity of each data disk is 32 TiB. You can set Release with Instance and Encrypt for each data disk.

Section	Parameter	Required	Description
Password	Set Password	Yes	Select when to set password. You can select Now or Later . If you select Later , you can use the Change Password feature in the console to set the password. For more information, see the Change Password topic in <i>ECS User Guide</i> .
	Logon Password	No	The password used to log on to the ECS instance. The password must be 8 to 30 characters in length and must contain at least three of the following character types: digits, uppercase letters, lowercase letters, and special characters.  Note The password is used to log on to the operating system and is not the VNC password.
	Confirm Password	No	Enter the password again.
Deployment Set	Deployment Set	No	The deployment set to which the instance belongs.
Instance Name	Configuration Name	No	The name of the scaling configuration.
	Instance Name	No	The name of the ECS instance.
User Data	User Data	No	Windows supports two formats: Bat and Powershell . Before you perform Base64 encoding, make sure to include <code>[bat]</code> or <code>[powershell]</code> as the first line. You can run shell scripts for Linux ECS instances.
Quantity	Quantity	No	The number of instances to purchase.

7. Click **Submit**.

Result

After the scaling configuration is created, it is in the **Disabled** state and is displayed in your scaling configuration list. To automatically create ECS instances, you must apply a scaling configuration. For more information, see [Apply a scaling configuration](#).

4.5.2. View scaling configurations

This topic describes how to view scaling configurations.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. View the list of scaling configurations.

4.5.3. Modify a scaling configuration

This topic describes how to modify a scaling configuration. You can modify the parameters of a scaling configuration based on your actual needs.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. Find the target scaling configuration and click its name in the **Scaling Configuration Name/ID** column.
6. Modify the parameters of the scaling configuration. For more information about parameters of the scaling configuration, see [Create a scaling configuration](#).
7. Click **OK**.

4.5.4. Apply a scaling configuration

This topic describes how to apply a scaling configuration. You can create multiple scaling configurations for a scaling group and apply one.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. Find the target scaling configuration and click **Select** in the **Actions** column. Only one scaling configuration can be in the **Enabled** state in a scaling group. After a scaling configuration is applied, other scaling configurations are put into the **Disabled** state.
6. Click **OK**.

Result

The status of the scaling configuration changes from **Disabled** to **Enable** in the **Status** column.

4.5.5. Delete a scaling configuration

This topic describes how to delete a scaling configuration that is no longer needed. After you delete a scaling configuration, existing ECS instances that are created from the scaling configuration are not removed.

Prerequisites

The scaling configuration is in the **Disabled** state.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.

3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Configuration**.
5. Find the target scaling configuration and click **Delete** in the **Actions** column.
6. Click **OK**.

4.6. Scaling rules

4.6.1. Create a scaling rule

This topic describes how to create a scaling rule. You can create scaling rules to add or remove ECS instances. For example, you can add an ECS instance to a scaling group.

Context

- You can create only a limited number of scaling rules for a scaling group. For more information, see the *Limits* topic in *Auto Scaling Product Introduction*.
- After a scaling rule is executed, the resulting number of ECS instances in the scaling group may fall outside of the specified range. In this case, Auto Scaling automatically adjusts the number of ECS instances to ensure that the number of ECS instances in the scaling group is within the specified range.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. Click **Create Scaling Rule**.
6. Configure parameters for the scaling rule.

Parameter	Required	Description
Rule Name	Yes	The name of the scaling rule. It must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Scaling Activity	Yes	The operation that is performed when the scaling rule is triggered. The operations include: <ul style="list-style-type: none"> ◦ Change to N instances: After the scaling rule is executed, the number of instances in the scaling group is changed to N. ◦ Add N instances: After the scaling rule is executed, N instances are added to the scaling group. ◦ Remove N instances: After the scaling rule is executed, N instances are removed from the scaling group.
Default Cooldown (Seconds)	No	The cooldown period. If this parameter is not specified, the default value is used.

7. Click **OK**.

4.6.2. View scaling rules

This topic describes how to view scaling rules.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
The scaling groups that correspond to the specified organization, resource set, and region are displayed.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. View the list of scaling rules.

4.6.3. Modify a scaling rule

This topic describes how to modify a scaling rule. You can modify the following parameters for a scaling rule: Rule Name, Scaling Activity, and Default Cooldown.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. Find the scaling rule that you want to modify and click **Edit** in the **Actions** column.
6. Modify the Rule Name, Scaling Activity, and Default Cooldown parameters.
7. Click **OK**.

4.6.4. Delete a scaling rule

This topic describes how to delete a scaling rule that is no longer needed.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. Find the target scaling rule and click **Delete** in the **Actions** column.
6. In the message that appears, click **OK**.

4.7. Scaling tasks

4.7.1. Manually execute a scaling rule

This topic describes how to manually execute a scaling rule to add or remove ECS instances.

Prerequisites

- The scaling group to which the scaling rule belongs is in the **Enable** state.
- No scaling activity is in progress in the scaling group to which the scaling rule belongs.

Context

After the scaling rule is executed, if the number of ECS instances is greater than the maximum number or less than the minimum number, Auto Scaling automatically adjusts the number of ECS instances to be within the valid range.

Auto Scaling enables you to manually execute scaling rules. You can also associate an event-triggered task or scheduled task with the scaling rule to automatically adjust the number of ECS instances. For more information, see [Create a scheduled task](#) and [Create an event-triggered task](#).

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **Scaling Rules**.
5. Find the scaling rule that you want to execute and click **Run** in the **Actions** column.
6. In the message that appears, click **OK**.

Result

The **Scaling Activities** page appears. You can view the details of your scaling activity.

4.7.2. Manually add an ECS instance

This topic describes how to manually add an ECS instance to a scaling group. You can add existing ECS instances to a scaling group to take full advantage of the computing resources.

Prerequisites

The ECS instance to be added must meet the following conditions:

- The ECS instance and the scaling group to which to add the instance share the same region, organization, and resource set.
- The ECS instance is in the **Running** state.
- The ECS instance does not belong to any scaling groups.
- The ECS instance and the scaling group are in the same VPC.

The scaling group to which to add the ECS instance must meet the following conditions:

- The scaling group is in the **Enable** state.
- No scaling activity is in progress in the scaling group.

Context

- When no scaling activity is being executed in the scaling group, you can add an ECS instance to the scaling group without the need to wait for the cooldown time to expire.
- If the number of instances in the scaling group is greater than the maximum number of instances after an ECS instance is added to the scaling group, the ECS instance cannot be added.
- The ECS instances that are manually added to a scaling group are not limited by scaling configurations. The instance types of the manually added instances can be different from that of the scaling configuration in the **Enable** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Click **Add Instance**.
6. Select the ECS instance to be added and click **OK**.

Result

The manually added instance is displayed on the **Manually Added** tab.

4.7.3. Manually remove an ECS instance

This topic describes how to manually remove an ECS instance that is no longer needed from a scaling group.

Prerequisites

The scaling group must meet the following conditions:

- The scaling group is in the **Enable** state.
- No scaling activity is in progress in the scaling group.

Context

- When no scaling activity is being executed in the scaling group, you can immediately remove an ECS instance from the scaling group without the need to wait for the cooldown time to expire.
- After an ECS instances is removed from a scaling group, the number of instances in the scaling group must be greater than or equal to the minimum number of instances. Otherwise, the ECS instance cannot be removed.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the top navigation bar, select an organization, a resource set, and a region.
3. Find the target scaling group and click the name of the scaling group in the **Scaling Group Name/ID** column.
4. In the left-side navigation pane, click **ECS Instances**.
5. Select the source of the ECS instance.
 - To select an automatically created ECS instance, click the **Auto Created** tab.
 - To select a manually added ECS instance, click the **Manually Added** tab.
6. Use one of the following methods to remove one or more ECS instances from a scaling group: **Manually added ECS instances can only be removed, but cannot be released.**
 - Find the ECS instance that you want to remove and choose **Actions > Remove from Scaling Group** in the **Actions** column.
 - Find the ECS instance that you want to remove and release, and choose **Actions > Remove from Scaling Group and Release** in the **Actions** column.
7. In the message that appears, click **OK**.

4.8. Scheduled tasks

4.8.1. Create a scheduled task

This topic describes how to create a scheduled task to scale computing resources in response to predictable business changes in the future. Scheduled tasks enable the system to automatically obtain sufficient computing resources before business peaks and release idle computing resources after the peaks.

Context

A scheduled task is preconfigured to execute the specified scaling rule at the specified time in the future. When the specified time arrives, the scheduled task automatically scales computing resources. This allows you to reduce costs and also meet business requirements. You can also specify the recurrence for scheduled tasks if business changes are regular.

If multiple scheduled tasks need to be executed in one minute, Auto Scaling executes the most recently created scheduled task.

Procedure

1. [Log on to the Auto Scaling console.](#)
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks** .

3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **Create Scheduled Task**.
5. In the dialog box that appears, configure parameters for the scheduled task.

Parameter	Required	Description
Task Name	Yes	The name of the scheduled task. The name must be 2 to 64 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	Yes	The description of the scheduled task.
Organization/Resource Group	Yes	The organization and resource set to which the scheduled task belongs.
Start Time	Yes	The time when the scheduled task is executed.
Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Retry Expiry Time	No	The period of time during which the system retries to execute the scheduled task. Unit: seconds. If a scaling activity fails to be executed at the specified time, Auto Scaling will execute the scheduled task again within the period of time that is specified by the Retry Expiry Time parameter.
Recurrence Settings	No	Specifies whether to execute the scheduled task repeatedly. Select Recurrence Settings and set the Recurrence and Expire parameters. The recurrence options include Daily , Weekly , and Monthly .

6. Click **OK**.

Result

The scheduled task that you created is displayed in the scheduled task list.

4.8.2. View scheduled tasks

This topic describes how to view scheduled tasks.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The scheduled tasks that correspond to the specified organization, resource set, and region are displayed.
4. Select a filter option, enter the corresponding information, and then click **Search**.

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

Option	Description
Task Name	Enter a task name to search for the scheduled task.
Task ID	Enter a scheduled task ID to search for the scheduled task.

5. View the scheduled task list.

4.8.3. Modify a scheduled task

This topic describes how to modify a scheduled task. You can modify the following parameters for a scheduled task: Start Time, Scaling Rules, and Retry Expiry Time.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the scheduled task that you want to modify and click **Edit** in the **Actions** column.
5. Modify the parameters of the scheduled task. You can modify the **Recurrence** and **Expire** parameters if you have enabled the **Recurrence Settings** feature when you create the scheduled task, but the **Recurrence Settings** feature cannot be disabled. For more information about other parameters, see [Create a scheduled task](#).
6. Click **OK**.

4.8.4. Disable a scheduled task

This topic describes how to disable a scheduled task. You can disable a scheduled task if you do not want to use it to trigger scaling activities.

Prerequisites

The scheduled task is in the **Running** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the scheduled task that you want to disable and click **Disabled** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the scheduled task changes from **Running** to **Stop** in the **Status** column.

4.8.5. Enable a scheduled task

This topic describes how to enable a scheduled task. You can enable a scheduled task that has been disabled and use it to trigger a scaling activity at the specified time point.

Prerequisites

The scheduled task is in the **Stop** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the scheduled task that you want to enable and click **Enable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the scheduled task changes from **Stop** to **Running** in the **Status** column.

4.8.6. Delete a scheduled task

This topic describes how to delete a scheduled task that is no longer needed.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Scheduled Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the scheduled task that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

4.9. Event-triggered tasks

4.9.1. Create an event-triggered task

This topic describes how to create an event-triggered task associated with monitoring metrics in response to emergent or unpredictable business changes. After you create and enable an event-triggered task, Auto Scaling collects data for the specified metric in real time, and triggers an alert when the specified condition is met. Then, Auto Scaling executes the scaling rule to dynamically scale ECS instances in the scaling group.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Event-driven Tasks > Alert Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Click **Create Alert**.
5. In the dialog box that appears, configure parameters for the event-triggered task.

Parameter	Required	Description
Task Name	Yes	The name of the event-triggered task. It must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.
Description	No	The description of the event-triggered task.
Organization/Resource Group	Yes	The organization and resource set in which to create the event-triggered task.
Monitoring Metrics/Scaling Rules	Yes	The scaling group to be monitored and the scaling rule to be executed.
Monitoring Type	Yes	System-Level Monitoring is selected by default.
Monitoring Metrics	Yes	The metrics that you want to monitor. Valid values: <ul style="list-style-type: none"> ◦ Average CPU Utilization ◦ Memory Usage ◦ Outbound Traffic ◦ Inbound Traffic

Parameter	Required	Description
Monitoring Period	Yes	<p>The period during which data is aggregated and analyzed. The shorter the period, the higher the frequency that the alert is triggered. Unit: minutes. Valid values:</p> <ul style="list-style-type: none"> ○ 1 ○ 2 ○ 5 ○ 15
Statistic	Yes	<p>The rule that determines whether to trigger an alert. Select Average, Max Capacity, or Min Capacity, and specify a threshold value. For example, alerts are triggered when the CPU utilization exceeds 80%:</p> <ul style="list-style-type: none"> ○ Average: Alerts are triggered when the average CPU utilization of all ECS instances in the scaling group exceeds 80%. ○ Max Capacity: Alerts are triggered when the highest CPU utilization among the ECS instances in the scaling group exceeds 80%. ○ Min Capacity: Alerts are triggered when the lowest CPU utilization among the ECS instances in the scaling group exceeds 80%.
Alert after occurrences	Yes	<p>The consecutive number of times that the threshold must be exceeded before the alert is triggered. Valid values:</p> <ul style="list-style-type: none"> ○ 1 ○ 2 ○ 3 ○ 5

6. Click **OK**.

4.9.2. View event-triggered tasks

This topic describes how to view event-triggered tasks.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Event-triggered Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
The event-triggered tasks that correspond to the specific organization, resource set, and region are displayed.
4. Select a filter option, enter the corresponding information, and then click **Search**.

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

Option	Description
Alert Name	Enter an event-triggered task name to search for the event-triggered task.
Scaling Group ID	Enter a scaling group ID to search for the event-triggered task.

4.9.3. Modify an event-triggered task

This topic describes how to modify an event-triggered task. You can modify one or more of the following parameters: Scaling Rules, Monitoring Type, and Statistic.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Event-triggered Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the event-triggered task that you want to modify and click **Edit** in the **Actions** column.
5. Modify the parameters of the event-triggered task. You cannot modify the following parameters: Organization and Resource Group, Monitoring Metrics, and Monitoring Period. For more information about other parameters, see [Create an event-triggered task](#).
6. Click **OK**.

4.9.4. Disable an event-triggered task

This topic describes how to disable an event-triggered task. You can disable an event-triggered task if you no longer want to use it to trigger scaling activities.

Prerequisites

The event-triggered task is in the **Normal**, **Alerts**, or **Insufficient Data** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Event-triggered Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the event-triggered task that you want to disable and click **Disable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the event-triggered task is changed to **Stopped** in the **Status** column.

4.9.5. Enable an event-triggered task

This topic describes how to enable an event-triggered task. You can enable an event-triggered task that has been disabled to continue to monitor metrics and trigger scaling activities of a scaling group.

Prerequisites

The event-triggered task is in the **Stopped** state.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Event-triggered Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the event-triggered task that you want to enable and click **Enable** in the **Actions** column.
5. In the message that appears, click **OK**.

Result

The status of the event-triggered task changes from **Stopped** to **Normal** in the **Status** column.

4.9.6. Delete an event-triggered task

This topic describes how to delete an event-triggered task that is no longer needed.

Procedure

1. [Log on to the Auto Scaling console](#).
2. In the left-side navigation pane, choose **Scaling Tasks > Event-triggered Tasks**.
3. In the top navigation bar, select an organization, a resource set, and a region.
4. Find the event-triggered task that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

5.Resource Orchestration Service (ROS)

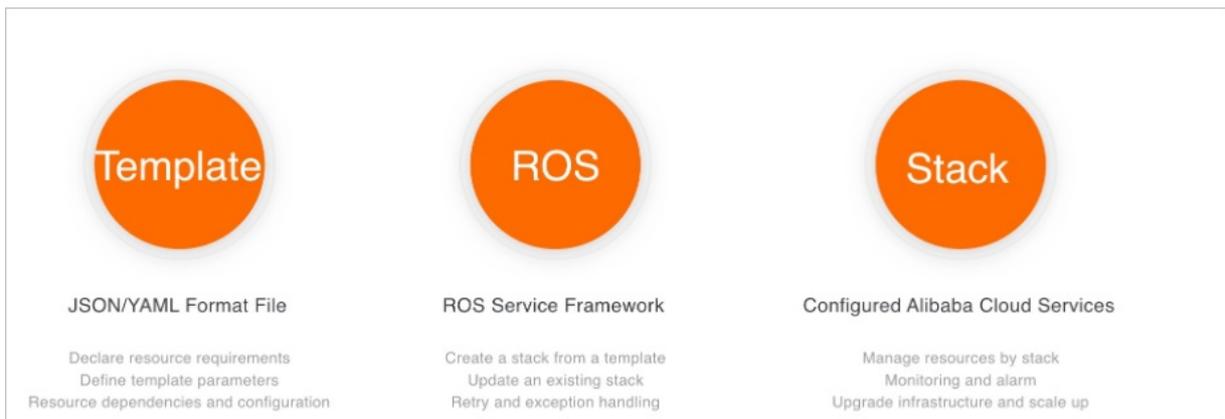
5.1. What is ROS?

Resource Orchestration Service (ROS) is a service provided by Alibaba Cloud to simplify the management of cloud computing resources. You can author stack templates based on the template specifications defined in ROS. Within a template, you can define required cloud computing resources such as ECS and ApsaraDB for RDS instances, and the dependencies between resources. The ROS engine automatically creates and configures all resources in a stack based on a template, making automatic deployment and O&M possible.

An ROS template is a readable, easy-to-author text file. You can directly edit a JSON-formatted template or use the Visual Editor available in the ROS console to edit the template. You can modify templates at any time. You can use version control tools such as SVN and Git to control the template and infrastructure versions. You can use APIs and SDKs to integrate the orchestration capabilities of ROS with your own applications to implement infrastructure as code.

ROS templates are also a standardized way to deliver resources and applications. If you are an independent software vendor (ISV), you can use ROS templates to deliver a holistic system and solution encompassing cloud resources and applications. ISVs can use this method to integrate Alibaba Cloud resources with their own software systems for centralized delivery.

ROS manages a group of cloud resources as a single unit called a stack. A stack is a group of Alibaba Cloud resources. You can create, delete, and clone cloud resources by stack. In DevOps practices, you can use ROS to clone the development, testing, and production environments, as well as migrate and scale out applications.



5.2. Log on to the ROS console

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

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Elastic Computing > Resource Orchestration Service**.

5.3. Create a stack

This topic describes how to use Resource Orchestration Service (ROS) to create a stack.

Prerequisites

Log on to the ROS console. For more information, see [Log on to the ROS console](#).

Procedure

1. In the upper-right corner of the page, click **Create Stack**.
2. In the **Select Template** step, set **Organization**, **Resource Set**, and **regionId**.
3. In the **Prepare Template** section, enter template content in the JSON format. Click **Next**.
4. In the **Configure Template Parameters** step, enter the stack name and parameters, and then click **Next**.
5. In the **Configure Stack** step, set **Rollback on Failure** and **Timeout Period**.
6. In the **Confirm** step, ensure the configurations of template and stack are correct and click **Create Stack**.

What's next

On the **Stacks** page in the ROS console:

- To delete a stack, click **Delete** in the **Actions** column corresponding to the target stack.
- To update a stack, click **Update** in the **Actions** column corresponding to the target stack.
- To recreate a stack, click **Recreate** in the **Actions** column corresponding to the target stack.

Note

- If you only need to modify the current template and configurations of a specified stack but do not need to change the region where the stack resides, update the stack.
- If you need to modify the current template and configurations of a specified stack and change the region where the stack resides, recreate the stack.

5.4. Template syntax

5.4.1. Template structure

A template is a UTF-8 encoded JSON file that is used to create stacks. Templates serve as the blueprint for underlying infrastructure and architecture. Templates define the configurations and dependencies of Alibaba Cloud resources.

ROS template structure

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",

  "Description" : "The template description used to provide information such as application scenarios and stack architecture.",
  "Metadata" : {
    // The template metadata that provides information such as layout for visualizations.
  },
  "Parameters" : {
    // The parameters you can specify when you create a stack.
  },

  "Mappings" : {
    // The mapping tables. Mapping tables are nested tables.
  },

  "Conditions": {
    // The conditions defined using internal condition functions. These conditions determine when to create associated resources.
  },

  "Resources" : {
    // The detailed information of resources such as configurations and dependencies.
  },

  "Outputs" : {
    // The outputs used to provide information such as resource properties. You can use the ROS console or API to obtain the information.
  }
}
```

ROSTemplateFormatVersion

Required. The template versions supported by Resource Orchestration Service (ROS). Current version: 2015-09-01.

Description

Optional. The description of the template, which is used to provide information such as the application scenarios and architecture of the template.

A detailed description can help users better understand the content of the template.

Metadata

Optional. The metadata of the template, in the JSON format.

Parameters

Optional. The parameters that you can specify when you create a stack. An ECS instance type is often defined as a parameter. Parameters have default values. Parameters can improve the flexibility and reusability of the template. When you create a stack, select appropriate specifications.

Mappings

Optional. Mappings are defined as nested mapping tables. You can use `Fn::FindInMap` to retrieve values corresponding to keys. You can also use parameter values as keys. For example, you can search the region-image mapping table for desired images by region.

Conditions

Optional. The conditions defined using `Fn::And`, `Fn::Or`, `Fn::Not`, and `Fn::Equals`. Separate multiple conditions with commas (,). The system will evaluate all conditions in the template before creating or updating a stack. All resources associated with true conditions are created, and all resources associated with false conditions are ignored.

Resources

Optional. The detailed information of resources in the stack created based on the template. The information includes resource dependencies and configurations.

Outputs

Optional. The outputs that are used to provide information such as resource properties. You can use the ROS console or API to obtain the information.

5.4.2. Parameters

The Parameters section improves the flexibility and reusability of a template. When you create a stack, you can replace parameter values in the template.

For example, assume that you have a web application requiring a stack that contains one SLB instance, two ECS instances, and one ApsaraDB for RDS instance. If the web application has a heavy workload, you can select an ECS instance with high specifications when you create the stack. Otherwise, you can select an ECS instance with low specifications. The following example shows how to define the `InstanceType` parameter for an ECS instance:

```

"Parameters" : {
  "InstanceType" : {
    "Type" : "String",
    "AllowedValues":["ecs.t1.small","ecs.s1.medium","ecs.m1.medium","ecs.c1.large"],
    "Default": "ecs.t1.small",
    "Label": "The ECS instance type",
    "Description" : "The type of the ECS instance you want to create. Default value: ecs.t1.small. Valid values: ecs.t1.small, ecs.s1.medium, ecs.m1.medium, and ecs.c1.large."
  }
}

```

You can assign a value to the `InstanceType` parameter when you create stacks based on templates. If this parameter is not specified, the default value `ecs.t1.small` is used.

The following example shows how to reference the `InstanceType` parameter when you define a resource:

```

"Websserver" : {
  "Type" : "ALIYUN::ECS::Instance",
  "InstanceType": {
    "Ref": "InstanceType"
  }
}

```

Syntax

Each parameter consists of a name and properties. The parameter name can only contain letters and digits and must be unique within the template. You can use the Label field to define the alias of the parameter.

The following table describes the parameters:

Parameter	Required	Description
Type	Yes	<p>The data type of the parameter.</p> <ul style="list-style-type: none"> String: a string value. Example: <code>"ecs.s1.medium"</code> . Number: an integer or floating-point number. Example: 3.14. CommaDelimitedList: a set of strings or numbers separated with commas (,), which can be indexed by using Fn::Select. Example: <code>"80, foo, bar"</code> . Json: a JSON-formatted string. Example: <code>{ "foo": "bar" }</code> . Boolean: a Boolean value. Example: <code>true</code> or <code>false</code> .
Default	No	If you do not specify a value when you create a stack, Resource Orchestration Service (ROS) checks whether a default value is defined in the template. If a default value is found, it is used. Otherwise, an error is returned.
AllowedValues	No	The list of valid parameter values.
AllowedPattern	No	The regular expression that is used to check whether the specified parameter value is a string. If the input is not a string, an error is returned.
MaxLength	No	The integer value that determines the longest string allowed for a String type parameter.
MinLength	No	The integer value that determines the shortest string allowed for a String type parameter.
MaxValue	No	The numeric value that determines the maximum value allowed for a Number type parameter.
MinValue	No	The numeric value that determines the minimum value allowed for a Number type parameter.
NoEcho	No	Specifies whether to mask the parameter value when the GetStack operation is called. If this parameter is set to true, only asterisks (*) are returned.
Description	No	The string that describes the parameter.
ConstraintDescription	No	The string that explains the parameter constraint.
Label	No	The alias of the parameter, encoded in UTF-8. When web forms are generated using templates, labels can be mapped to parameter names.

Parameter	Required	Description
AssociationProperty	No	<p>Automatically verifies the validity of the parameter value and provides a list of valid values.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ALIYUN::ECS::Instance::ImageId ALIYUN::ECS::Instance::ZoneId ALIYUN::ECS::VPC::VPCId ALIYUN::ECS::VSwitch::VSwitchId <p>When AssociationProperty is set to ALIYUN::ECS::Instance::ImageId, the ROS console verifies whether the specified image ID is available and lists other values in a drop-down list.</p>
Confirm	No	<p>Specifies whether to enter the parameter value a second time when the NoEcho parameter is set to true. Default value: false.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Notice This parameter can be set to true only when it is used with a String type parameter and the NoEcho parameter is set to true.</p> </div>

Examples

In the following example, two parameters are defined in the Parameters section.

- `username`
 - Type: String
 - Default value: anonymous. Valid values:
 - anonymous
 - user-one
 - user-two
 - Length: 6 to 12

 **Notice** The default value must also meet the length and valid value requirements.

- `password`
 - Type: String
 - Length: 1 to 41
 - The password can contain uppercase letters, lowercase letters, and digits.
 - If you set the NoEcho parameter to true, the GetStack operation does not return any parameter values.

```

"Parameters" : {
  "username" : {
    "Label": "Username",
    "Description": "Enter the username",
    "Default": "anonymous",
    "Type" : "String",
    "MinLength" : "6",
    "MaxLength" : "12",
    "AllowedValues": ["anonymous", "user-one", "user-two"]
  },
  "password" : {
    "Label": "Password",
    "NoEcho" : "True",
    "Description": "Enter the password",
    "Type" : "String",
    "MinLength" : "1",
    "MaxLength" : "41",
    "AllowedPattern" : "[a-zA-Z0-9]*"
  }
}

```

Pseudo parameters

Pseudo parameters are internal parameters provided by the ROS engine. They can be referenced in the same manner as user-defined parameters, and their values are determined while ROS is running. The following pseudo parameters are supported:

- ALIYUN::StackName: the name of the stack.
- ALIYUN::StackId: the ID of the stack.
- ALIYUN::Region: the region where the stack resides.
- ALIYUN::AccountId: the user ID of the stack.
- ALIYUN::NoValue: specifies whether the specific resource property is deleted when the resource is created or updated.

5.4.3. Resources

This topic describes the properties of each resource and dependencies of resources in a stack. A resource can be referenced by other resources and output items.

Syntax

Each resource consists of an ID and a description. All resource descriptions are enclosed in braces {}. Separate multiple resources with commas (.). The following sample code shows the Resources syntax:

```

"Resources" : {
  "Resource1 ID" : {
    "Type": "The resource type",
    "Condition": "The condition that specifies whether to create the resource",
    "Properties" : {
      The description of the resource properties
    }
  },
  "Resource2 ID" : {
    "Type": "The resource type",
    "Condition": "The condition that specifies whether to create the resource",
    "Properties" : {
      The description of the resource properties
    }
  }
}

```

Parameter description:

- The resource ID must be unique within the template. You can use the resource ID to reference the resource in other parts of the template.
- The Type parameter specifies the type of resource that is being declared. For example, ALIYUN::ECS::Instance indicates that the resource is an Elastic Cloud Service (ECS) instance.
- The Properties section provides additional options that you can specify for a resource. For example, you must specify an image ID for each Alibaba Cloud ECS instance. The image ID is one of the resource properties.

Examples

```

"Resources" : {
  "ECSInstance" : {
    "Type" : "ALIYUN::ECS::Instance",
    "Properties" : {
      "ImageId" : "m-25l0r****"
    }
  }
}

```

If a resource does not need properties to be declared, omit the Properties section of that resource.

Property values can be text strings, string lists, Boolean values, referenced parameters, or return values of functions.

The following example shows how to declare different types of property values:

```

"Properties" : {
  "String" : "string",
  "LiteralList" : [ "value1", "value2" ],
  "Boolean" : "true"
  "ReferenceForOneValue" : { "Ref" : "ResourceID" },
  "FunctionResultWithFunctionParams" : {
    "Fn::Join" : [ "%", [ "Key=", { "Ref" : "SomeParameter" } ] ] }
}

```

DeletionPolicy

The `DeletionPolicy` parameter specifies whether to retain a resource when its stack is deleted. The following sample code shows how to use the `DeletionPolicy` parameter to retain an ECS instance when its stack is deleted:

```

"Resources" : {
  "ECSInstance" : {
    "Type" : "ALIYUN::ECS::Instance",
    "Properties" : {
      "ImageId" : "m-25l0r****"
    },
    "DeletionPolicy" : "Retain"
  }
}

```

DependsOn

The `DependsOn` parameter allows you to create a specific resource after you create its dependent resource. If you specify the `DependsOn` parameter for a resource, the resource is created only after its dependent resource specified by the `DependsOn` parameter is created.

In the following example, `WebServer` is created only after `DatabaseServer` is created:

```

{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "DependsOn": "DatabaseServer"
    },
    "DatabaseServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
      }
    }
  }
}

```

Condition

The Condition parameter specifies whether to create the resource. The resource can be created only when the Condition parameter is set to true.

In the following example, WebServer is created only if the condition determined by the MaxAmount parameter is true:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters": {
    "MaxAmount": {
      "Type": "Number",
      "Default": 1
    }
  },
  "Conditions": {
    "CreateWebServer": {"Fn::Not": {"Fn::Equals": [0, {"Ref": "MaxAmount"}]}}
  }
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Condition": "CreateWebServer",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
        "MaxAmount": {"Ref": "MaxAmount"}
      }
    },
    "DatabaseServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
      }
    }
  }
}
```

Resource declaration example

The following example shows how to declare a resource:

```

"Resources" : {
  "WebServer": {
    "Type": "ALIYUN::ECS::Instance",
    "Properties": {
      "ImageId": "m-25l0r****",
      "InstanceType": "ecs.t1.small",
      "SecurityGroupId": "sg-25zwc****",
      "ZoneId": "cn-beijing-b",
      "Tags": [{
        "Key": "Department1",
        "Value": "HumanResource"
      }{
        "Key": "Department2",
        "Value": "Finance"
      }
    ]
  },
  "ScalingConfiguration": {
    "Type": "ALIYUN::ESS::ScalingConfiguration",
    "Properties": {
      "ImageId": "ubuntu1404_64_20G_aliaegis_2015****.vhd",
      "InstanceType": "ecs.t1.small",
      "InstanceId": "i-25xhh****",
      "InternetChargeType": "PayByTraffic",
      "InternetMaxBandwidthIn": 1,
      "InternetMaxBandwidthOut": 20,
      "SystemDisk_Category": "cloud",
      "ScalingGroupId": "bwhtvpcBcKYac9fe3vd0****",
      "SecurityGroupId": "sg-25zwc****",
      "DiskMappings": [
        {
          "Size": 10
        },
        {
          "Category": "cloud",
          "Size": 10
        }
      ]
    }
  }
}

```

5.4.4. Outputs

The **Outputs** section is used to define the values returned when the `GetStack` operation is called. For example, if you define an ECS instance ID as an output item, the ECS instance ID is returned when the `GetStack` operation is called.

Syntax

Each output item consists of an ID and a description. All output descriptions are enclosed in braces `{}`. Separate multiple output items with commas `,`. Each output item can have multiple values in the array format. The following example shows the **Outputs** syntax:

```
"Outputs" : {
  "Output1 ID" : {
    "Description": "The description of the output item",
    "Condition": "The condition that specifies whether to provide resource properties",
    "Value": "The output value expression"
  },
  "Output2 ID" : {
    "Description": "The description of the output item",
    "Condition": "The condition that specifies whether to provide resource properties",
    "Value" : [
      "Output value expression 1",
      "Output value expression 2",
      ...
    ]
  }
}
```

- **Output ID:** the ID of the output item. Duplicate IDs are not allowed within a template.
- **Description:** optional. The description of the output item.
- **Value:** required. The value returned when the `GetStack` operation is called.
- **Condition:** optional. The condition that specifies whether to create a resource and provide its information. The resource is created and its information is provided only when the specified condition is true.

In the following example, `WebServer` is created only if the condition determined by the `MaxAmount` parameter is true:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters": {
    "MaxAmount": {
      "Type": "Number",
      "Default": 1
    }
  },
  "Conditions": {
    "CreateWebServer": {"Fn::Not": {"Fn::Equals": [0, {"Ref": "MaxAmount"}]}}
  }
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Condition": "CreateWebServer",
      "Properties": {
        "ImageId" : "m-25l0r****",
        "InstanceType": "ecs.t1.small"
        "MaxAmount": {"Ref": "MaxAmount"}
      }
    }
  }
  "Outputs": {
    "WebServerIP": {
      "Condition": "CreateWebServer",
      "Value": {
        "Fn::GetAtt": ["WebServer", "PublicIps"]
      }
    }
  }
}
```

Examples

The following example contains two output items.

- The value of the InstanceId parameter of WebServer.
- The values of the PublicIps and PrivateIps parameters of WebServer.

```

"Outputs": {
  "InstanceId": {
    "Value": {"Fn::GetAtt": ["WebServer", "InstanceId"]}
  },
  "PublicIp & PrivateIp": {
    "Value": [
      {"Fn::GetAtt": ["WebServer", "PublicIp"]},
      {"Fn::GetAtt": ["WebServer", "PrivateIp"]}
    ]
  }
}

```

5.4.5. Functions

Resource Orchestration Service (ROS) provides several built-in functions to help you manage stacks. You can use built-in functions to define Resources and Outputs.

Fn::Base64Encode

The Fn::Base64Encode function is used to return the Base64 representation of the input string.

Declaration

```

{"Fn::Base64Encode": "stringToEncode"}

```

Parameters

`stringToEncode` : the string decoded from the Base64-encoded string.

Return value

The Base64 representation of the input string.

Examples

```

{"Fn::Base64Encode": "string to encode"}

```

`c3RyaW55IHRvIGVudY29kZQ==` is returned in this example.

Fn::Base64Decode

The Fn::Base64Decode function is used to return a string decoded from a Base64-encoded string.

Declaration

```

{"Fn::Base64Decode": "stringToEncode"}

```

Parameters

`stringToDecode` : the string decoded from the Base64-encoded string.

Return value

The string decoded from the Base64-encoded string.

Examples

```
{"Fn::Base64Decode": "c3RyaW55IHRvIGVudY29kZQ=="}
```

string to encode is returned in this example.

Fn::Base64

The Fn::Base64 function returns the Base64 representation of the input string.

- Declaration

```
"Fn::Base64": stringToEncode
```

- Parameters

valueToEncode: the string to be encoded in Base64.

- Return value

The Base64 representation of the input string.

- Examples

```
"Fn::Base64": "string to encode"
```

Fn::FindInMap

The Fn::FindInMap function is used to return the values based on keys in a two-level mapping that is declared in the Mappings section.

Declaration

```
"Fn::FindInMap": ["MapName", "TopLevelKey", "SecondLevelKey"]
```

Parameters

- **MapName** : the ID of a mapping declared in the Mappings section that contains keys and values.
- **TopLevelKey** : the top-level key name. The value is a list of key-value pairs.
- **SecondLevelKey** : the second-level key name. The value is a string or a number.

Return value

The value that is assigned to the SecondLevelKey parameter.

Examples

The ImageId property must be specified when you create a WebServer instance. The Mappings section describes the ImageId mappings by region. The Parameters section describes the regions that must be specified by template users. Fn::FindInMap finds the corresponding ImageId mapping in RegionMap based on the region specified by a user, and then finds the corresponding ImageId in the mapping.

- MapName can be set to a custom value, which is "RegionMap" in this example.
- TopLevelKey is set to the region where the stack is created, which is {"Ref": "regionParam"} in this example.
- SecondLevelKey is set to the required architecture, which is "32" in this example.

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou"
      ]
    }
  }
}
```


- Fn::FindInMap
- Ref

Fn::GetAtt

The Fn::GetAtt function is used to return the value of a property from a resource in a template.

Declaration

```
"Fn::GetAtt": ["resourceID", "attributeName"]
```

Parameters

- `resourceID` : the ID of the resource.
- `attributeName` : the name of the resource property.

Return value

The value of the resource property.

Examples

The ImageId property of MyEcsInstance is returned in this example.

```
{"Fn::GetAtt" : ["MyEcsInstance", "ImageID"]}
```

Fn::Join

The Fn::Join function is used to append a set of values into a single value that is separated by a specified delimiter.

Declaration

```
{"Fn::Join": ["delimiter", ["string1", "string2", ... ]]}
```

Parameters

- `delimiter` : the value used to divide the string. The delimiter value can be left blank so that all the values are directly combined.
- `["string1", "string2", ...]` : the list of values that are combined into a string.

Return value

The combined string.

Examples

```
{"Fn::Join": [ " ", ["a", "b", "c"]]}
```

`"a,b,c"` is returned in this example.

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Select

The Fn::Select function is used to return a single data element from a list of data elements by an index.

Declaration

- The following example assumes that the list of data elements is an array:

```
"Fn::Select": ["index", ["value1", "value2", ... ]]
```

- The following example assumes that the list of data elements is a mapping table:

```
"Fn::Select": ["index", {"key1": "value1", ... }]
```

Parameters

`index` : the index of the object data element. If the list of data elements is an array, the index must be an integer ranging from 0 to N-1, where N indicates the number of elements in the array. If the list of data elements is a mapping table, the index must be a key in the mapping table.

If the corresponding value of the index cannot be found, the system returns an empty string.

Return value

The object data element.

Examples

- The following example assumes that the list of data elements is an array:

```
{"Fn::Select": ["1", ["apples", "grapes", "oranges", "mangoes"]]}
```

`"grapes"` is returned in this example.

- The following example assumes that the list of data elements is a mapping table:

```
{"Fn::Select": ["key1", {"key1": "grapes", "key2": "mangoes"}]}
```

`"grapes"` is returned in this example.

- The following example assumes that the list of data elements is a comma-delimited list:

```
"Parameters": {
  "userParam": {
    "Type": "CommaDelimitedList",
    "Default": "10.0.100.0/24, 10.0.101.0/24, 10.0.102.0/24"
  }
}

"Resources": {
  "resourceID": {
    "Properties": {
      "CidrBlock": {"Fn::Select": ["0", {"Ref": "userParam"}]}
    }
  }
}
```

Supported functions

For the `Fn::Select` index value, you can use the `Ref` function.

For the `Fn::Select` list of data elements, you can use the following functions:

- `Fn::Base64Encode`
- `Fn::FindInMap`
- `Fn::GetAtt`

- Fn::Join
- Fn::Select
- Ref

Ref

The Ref function is used to return the value of a specified parameter or resource.

If the specified parameter is a resource ID, the value of the resource is returned. Otherwise, the system will return the value of the specified parameter.

Declaration

```
"Ref": "logicalName"
```

Parameters

`logicalName` : the logical name of the resource or parameter that you want to reference.

Return value

The value of the resource or parameter.

Examples

The following example uses the Ref function to specify `regionParam` as the region parameter for `RegionMap` of `WebServer`:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou",
        "beijing"
      ]
    }
  },
  "Mappings": {
    "RegionMap": {
      "hangzhou": {
        "32": "m-25l0rcfjo",
        "64": "m-25l0rcfj1"
      },
      "beijing": {
        "32": "m-25l0rcfj2",
        "64": "m-25l0rcfj3"
      }
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
```

```

"Properties": {
  "ImageId": {
    "Fn::FindInMap": [
      "RegionMap",
      {"Ref": "regionParam"},
      "32"
    ]
  },
  "InstanceType": "ecs.t1.small",
  "SecurityGroupId": "sg-25zwc****",
  "ZoneId": "cn-beijing-b",
  "Tags": [
    {
      "Key": "tiantt",
      "Value": "ros"
    },
    {
      "Key": "tiantt1",
      "Value": "ros1"
    }
  ]
}
}
}
}
}
}

```

Supported function

When you use Ref function, you cannot use other functions in it at the same time. You must specify a string value for the resource logical ID.

Fn::GetAZs

The Fn::GetAZs function is used to return a list of zones for a specified region.

Declaration

```
"Fn::GetAZs": "region"
```

Parameters

region : the ID of the region.

Return value

The list of zones for the specified region.

Examples

The following example demonstrates how to create an ECS instance in the first zone of a specified region:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": "centos7u2_64_40G_cloudinit_2016****.raw",
        "InstanceType": "ecs.n1.tiny",
        "SecurityGroupId": "sg-2zedcm7ep5quses0****",
        "Password": "Ros1****",
        "AllocatePublicIP": true,
        "InternetChargeType": "PayByTraffic",
        "InternetMaxBandwidthIn": 100,
        "InternetMaxBandwidthOut": 100,
        "SystemDiskCategory": "cloud_efficiency",
        "IoOptimized": "optimized",
        "ZoneId": {"Fn::Select": ["0", {"Fn::GetAZs": {"Ref": "ALIYUN::Region"}}]}
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Value": {"Fn::GetAtt": ["WebServer", "InstanceId"]}
    },
    "PublicIp": {
      "Value": {"Fn::GetAtt": ["WebServer", "PublicIp"]}
    }
  }
}

```

Supported functions

- Fn::Base64Encode
- Fn::FindInMap
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Replace

The Fn::Replace function is used to replace a specified substring contained in a string with a new substring.

Declaration

```

{"Fn::Replace": [{"object_key": "object_value"}, "object_string"]}

```

Parameters

- `object_key` : the substring to be replaced.

- `object_value` : the new substring to replace the previous substring.
- `object_string` : the string whose `object_key` is replaced.

Return value

The string after replacement.

Examples

The following example demonstrates how to replace "print" with "echo" in the specified script:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId" : "centos_7_2_64_40G_base_2017****.vhd",
        "InstanceType": "ecs.n1.medium",
        "SecurityGroupId": "sg-94q49****",
        "Password": "MytestPassword****",
        "IoOptimized": "optimized",
        "VSwitchId": "vsw-94v8v8****",
        "VpcId": "vpc-949uz****",
        "SystemDiskCategory": "cloud_oss",
        "UserData": {"Fn::Replace": [{"print": "echo"},
          {"Fn::Join": ["", [
            "#! /bin/sh\n",
            "mkdir ~/test_ros\n",
            "print hello > ~/1.txt\n"
          ]]}]}
      }
    }
  },
  "Outputs": {
    "Instanceid": {
      "Value" : {"Fn::GetAtt": ["WebServer","Instanceid"]}
    },
    "Publicip": {
      "Value" : {"Fn::GetAtt": ["WebServer","Publicip"]}
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref

Fn::Split

The Fn::Split function is used to split a string into a list of values separated by a specified delimiter and return the list.

Declaration

```
"Fn::Split": ["delim", "original_string"]
```

Parameters

- `delim` : the specified delimiter, which can be commas (,), semicolons (;), line breaks (\n), and indents (\t).
- `original_string` : the string to be split.

Return value

A list of string values.

Examples

- The following example assumes that the list of data elements is an array:

```
{"Fn::Split": [";", "foo; bar; achoo"]}
```

`["foo", " bar", "achoo "]` is returned in this example.

- The following example uses Fn::Split to split InstanceIds:

```
{
  "Parameters": {
    "InstanceIds": {
      "Type": "String",
      "Default": "instane1_id,instance2_id,instance2_id"
    }
  },
  "Resources": {
    "resourceID": {
      "Type": "ALIYUN::SLB::BackendServerAttachment",
      "Properties": {
        "BackendServerList": {
          "Fn::Split": [
            ",",
            {
              "Ref": "InstanceIds"
            }
          ]
        }
      }
    }
  }
}
```

Supported functions

- Fn::Base64Encode
- Fn::FindInMap

- Fn::GetAtt
- Fn::Join
- Fn::Select
- Fn::Replace
- Fn::GetAZs
- Fn::If
- Ref

Fn::Equals

The Fn::Equals function is used to compare whether two values are equal. If the two values are equal, true is returned. If the two values are not equal, false is returned.

Declaration

```
{"Fn::Equals": ["value_1", "value_2"]}
```

Parameters

value : the values to be compared.

Return value

true or false.

Examples

The following example uses Fn::Equals to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "EnvType": {
      "Default": "pre",
      "Type": "String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {
      "Fn::Equals": [
        "prod",
        {"Ref": "EnvType"}
      ]
    }
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And

- Ref

Fn::And

The Fn::And function is used to represent the AND operator, and must contain at least two conditions. If all the specified conditions are evaluated as true, true is returned. If any condition is evaluated as false, false is returned.

Declaration

```
{"Fn::And": ["condition", {...]}
```

Parameters

`condition` : the condition to be evaluated.

Return value

true or false.

Examples

The following example uses Fn::And to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {"Fn::Equals": ["prod", {"Ref": "EnvType"}]},
    "TestAndCond": {"Fn::And": ["TestEqualsCond", {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}]}
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::Or

The Fn::Or function is used to represent the OR operator, and must contain at least two conditions. If any specified condition is evaluated as true, true is returned. If all the conditions are evaluated as false, false is returned.

Declaration

```
{"Fn::Or": ["condition", {...]}
```

Parameters

`condition` : the condition to be evaluated.

Return value

true or false.

Examples

The following example uses `Fn::Or` to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestEqualsCond": {"Fn::Equals": ["prod", {"Ref": "EnvType"}]},
    "TestOrCond": {"Fn::And": ["TestEqualsCond", {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}]}
  }
}
```

Supported functions

- `Fn::Or`
- `Fn::Not`
- `Fn::Equals`
- `Fn::FindInMap`
- `Fn::And`
- `Ref`

Fn::Not

The `Fn::Not` function is used to represent the NOT operator. If a condition is evaluated as false, true is returned. If a condition is evaluated as true, false is returned.

Declaration

```
{"Fn::Not": "condition"}
```

Parameters

`condition` : the condition to be evaluated.

Return value

true or false.

Examples

The following example uses `Fn::Not` to define a condition in the Conditions section:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
  "Conditions": {
    "TestNotCond": {"Fn::Not": {"Fn::Equals": ["pre", {"Ref": "EnvType"}]}}
  }
}
```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::If

This function returns one of two possible values. If a specified condition is evaluated as true, one value is returned. If the specified condition is evaluated as false, the other value is returned. The property values of Resources and Outputs in templates support the Fn::If function. You can use the `ALIYUN::NoValue` pseudo parameter as the return value to delete the corresponding property.

Declaration

```
{"Fn::If": ["condition_name", "value_if_true", "value_if_false"]}
```

Parameters

- `condition_name` : the name of the condition in the Conditions section. A condition is referenced by using the condition name.
- `value_if_true` : If the specified condition is evaluated as true, this value is returned.
- `value_if_false` : If the specified condition is evaluated as false, this value is returned.

Examples

The following example demonstrates how to determine whether to create a data disk based on input parameters:

```
{
  "ROSTemplateFormatVersion":"2015-09-01",
  "Parameters":{
    "EnvType":{
      "Default":"pre",
      "Type":"String"
    }
  },
}
```

```

"Conditions":{
  "CreateDisk":{
    "Fn::Equals":[
      "prod",
      {
        "Ref":"EnvType"
      }
    ]
  }
},
"Resources":{
  "WebServer":{
    "Type":"ALIYUN::ECS::Instance",
    "Properties":{
      "DiskMappings":{
        "Fn::If":[
          "CreateDisk",
          [
            {
              "Category":"cloud_efficiency",
              "DiskName":"FirstDataDiskName",
              "Size":40
            },
            {
              "Category":"cloud_ssd",
              "DiskName":"SecondDataDiskName",
              "Size":40
            }
          ],
          {
            "Ref":"ALIYUN::NoValue"
          }
        ]
      },
      "VpcId":"vpc-2zew9pxh2yirtzqxd****",
      "SystemDiskCategory":"cloud_efficiency",
      "SecurityGroupId":"sg-2zece6wcqriejf1v****",
      "SystemDiskSize":40,
      "ImageId":"centos_6_8_64_40G_base_2017****.vhd",
      "IoOptimized":"optimized",
      "VSwitchId":"vsw-2zed9txvy7h2srqo6****",
      "InstanceType":"ecs.n1.medium"
    }
  }
},
"Outputs":{
  "InstanceId":f

```

```

instances: {
  "Value": {
    "Fn::GetAtt": [
      "WebServer",
      "Instanceid"
    ]
  }
},
"Zoneid": {
  "Value": {
    "Fn::GetAtt": [
      "WebServer",
      "Zoneid"
    ]
  }
}
}
}
}

```

Supported functions

- Fn::Or
- Fn::Not
- Fn::Equals
- Fn::FindInMap
- Fn::And
- Ref

Fn::ListMerge

The Fn::ListMerge function is used to merge multiple lists into one list.

Declaration

```
{"Fn::ListMerge": [{"list_1_item_1", "list_1_imte_2", ...}, {"list_2_item_1", "list_2_imte_2", ...}]}
```

Parameters

- ["list_1_item_1", "list_1_imte_2", ...] : the first list to merge.
- ["list_2_item_1", "list_2_imte_2", ...] : the second list to merge into the first list.

Examples

The following example demonstrates how to attach two ECS instance groups to an SLB instance:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "ros",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth",
      }
    },
    "BackendServer1": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId": "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3****",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    },
    "BackendServer2": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId": "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3iu****",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    },
    "Attachment": {
      "Type": "ALIYUN::SLB::BackendServerAttachment",
      "Properties": {
        "LoadBalancerId": {"Ref": "LoadBalancer"},
        "BackendServerList": { "Fn::ListMerge": [
          {"Fn::GetAtt": ["BackendServer1", "InstanceIds"]},
          {"Fn::GetAtt": ["BackendServer2", "InstanceIds"]}
        ]
      }
    }
  }
}

```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join
- Fn::If

Fn::GetJsonValue

The Fn::GetJsonValue function is used to resolve a JSON string and obtain its key value from the first layer.

Declaration

```
{"Fn::GetJsonValue": ["key", "json_string"]}
```

Parameters

- `key` : the key value.
- `json_string` : the specified JSON string to be resolved.

Examples

In the following example, the WebServer instance executes UserData and returns a JSON string, and the WebServer2 instance then obtains the corresponding key value from the string.

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": "m-2ze45uwova5fedlu****",
        "InstanceType": "ecs.n1.medium",
        "SecurityGroupId": "sg-2ze7pxymaix640qr****",
        "Password": "Wenqiao****",
        "IoOptimized": "optimized",
        "VSwitchId": "vsw-2zei67xd9nhcqxe****",
        "VpcId": "vpc-2zevx9ios1rszqv0a****",
        "SystemDiskCategory": "cloud_ssd",
        "UserData": {"Fn::Join": ["", [
          "#! /bin/sh\n",
          "mkdir ~/test_ros\n",
          "print hello > ~/1.txt\n",
          "Fn::GetAtt": ["WaitConHandle", "CurlCli"],
          "\n",
          "Fn::GetAtt": ["WaitConHandle", "CurlCli"],
          "-d '{\"id\": \"1\", \"data\": [\"1111\", \"2222\"]}'\n"
        ]]},
        "PrivateIpAddress": "192.168.XX.XX",
        "HostName": "userdata-1
      }
    }
  }
}
```

```

},
"WaitConHandle": {
  "Type": "ALIYUN::ROS::WaitConditionHandle"
},
"WaitCondition": {
  "Type": "ALIYUN::ROS::WaitCondition",
  "Properties": {
    "Handle": {"Ref": "WaitConHandle"},
    "Timeout": 900
  }
},
"WebServer2": {
  "Type": "ALIYUN::ECS::Instance",
  "Properties": {
    "ImageId": "m-2ze45uwova5fedlu****",
    "InstanceType": "ecs.n1.medium",
    "SecurityGroupId": "sg-2ze7pxymaix640qr****",
    "Password": "Wenqiao****",
    "IoOptimized": "optimized",
    "VSwitchId": "vsw-2zei67xd9nhcqzxec****",
    "VpcId": "vpc-2zevx9ios1rszqv0a****",
    "SystemDiskCategory": "cloud_ossd",
    "UserData":
      {"Fn::Join": ["", [
        "#! /bin/sh\n",
        "mkdir ~/test_ros\n",
        "echo hello > ~/1.txt\n",
        "server_1_token=",
        {"Fn::GetJsonValue": ["1", {"Fn::GetAtt": ["WaitCondition", "Data"]}]}],
        "\n"}
  ]},
  "PrivateIpAddress": "192.168.XX.XX",
  "HostName": "userdata-2"
}
},
"Outputs": {
  "InstanceId": {
    "Value": {"Fn::GetAtt": ["WebServer", "InstanceId"]}
  },
  "PublicIp": {
    "Value": {"Fn::GetAtt": ["WebServer", "PublicIp"]}
  }
}
}

```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join
- Fn::If

Fn::MergeMapToList

The Fn::MergeMapToList function is used to merge multiple mappings into a list of mapping elements.

Declaration

```
{Fn::MergeMapToList: [{"key_1": ["key_1_item_1", "key_1_item_2", ...]}, {"key_2": ["key_2_item_1", "key_2_item_2", ...]}, ...
]}
```

Parameters

- {"key_1": ["key_1_item_1", "key_1_item_2", ...]} : the first mapping to merge. The "key_1" value must be a list. "key_1" is the key for each mapping in the list of merged mappings. The "key_1" value is "key_1_item_1" for the first merged mapping and "key_1_item_2" for the second merged mapping. All values follow the same format. The length of the final list of merged mappings is the length of the longest list "key_x" from all mappings being merged. If a "key_y" list is shorter, the last element of the list is repeated until the list is the longest.
- {"key_2": ["key_2_item_1", "key_2_item_2", ...]} : the second mapping to merge into the first mapping. The "key_2" value must be a list. "key_2" is the key for each mapping in the merged list. The "key_2" value is "key_2_item_1" for the first merged mapping and "key_2_item_2" for the second merged mapping.

Examples

- The following example demonstrates how to merge three mappings. The length of the list based on the key values for each mapping is the same.

```
{
  "Fn::MergeMapToList": [
    {"key_1": ["k1e_1_item_1", "k1e_1_item_2"]},
    {"key_2": ["k2e_2_item_1", "k2e_2_item_2"]},
    {"key_3": ["k3e_3_item_1", "k3e_3_item_2"]}
  ]
}
```

The following code shows the merged result:

```
[
  {
    "key_1": "kye_1_item_1",
    "key_2": "kye_2_item_1",
    "key_3": "kye_3_item_1"
  },
  {
    "key_1": "kye_1_item_2",
    "key_2": "kye_2_item_2",
    "key_3": "kye_3_item_2"
  }
]
```

- The length of the list based on the key values for each mapping varies in the following example:

```
{
  "Fn::MergeMapToList": [
    {"key_1": ["kye_1_item_1", "kye_1_item_2"]},
    {"key_2": ["kye_2_item_1", "kye_2_item_2", "key_2_item_3"]},
    {"key_3": ["kye_3_item_1", "kye_3_item_2"]}
  ]
}
```

The following code shows the merged result:

```
[
  {
    "key_1": "kye_1_item_1",
    "key_2": "kye_2_item_1",
    "key_3": "kye_3_item_1"
  },
  {
    "key_1": "kye_1_item_2",
    "key_2": "kye_2_item_2",
    "key_3": "kye_3_item_2"
  },
  {
    "key_1": "kye_1_item_2",
    "key_2": "kye_2_item_3",
    "key_3": "kye_3_item_2"
  }
]
```

- In the following template example, all instances created in WebServer are added to the VServer group of an SLB instance:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroupClone",
      "Properties": {
        "SourceInstanceId": "i-xxxxx",
        "Password": "Hello****",
        "MinAmount": 1,
        "MaxAmount": 1
      }
    },
    "CreateVServerGroup": {
      "Type": "ALIYUN::SLB::VServerGroup",
      "Properties": {
        "LoadBalancerId": "lb-****",
        "VServerGroupName": "VServerGroup-****",
        "BackendServers": {
          "Fn::MergeMapToList": [
            {"Port": [6666, 9090, 8080]},
            {"ServerId": {"Fn::GetAtt": ["WebServer", "InstanceIds"]}},
            {"Weight": [20, 100]}
          ]
        }
      }
    }
  }
}

```

Supported functions

- Fn::Base64Encode
- Fn::GetAtt
- Fn::Join
- Fn::Select
- Ref
- Fn::Join
- Fn::If
- Fn::ListMerge
- Fn::GetJsonValue

Fn::Avg

The Fn::Avg function is used to return the average value of a set of numbers.

Declaration

```
{"Fn::Avg": [ndigits, [number1, number2, ... ]]}
```

Parameters

- `ndigits` : the number of decimal places to report. This parameter value must be an integer.
- `[number1, number2, ...]` : the set of numbers for which the average value will be calculated. Each element in the group must be either a number or a string that can be converted into a number.

Return value

The average value of the set of numbers.

Examples

```
{ "Fn::Avg": [ 1, [1, 2, 6.0] ] }
{ "Fn::Avg": [ 1, ['1', '2', '6.0'] ] }
```

3.0 is returned in this example.

Supported functions

- `Fn::GetAtt`
- `Ref`

Fn::SelectMapList

The `Fn::SelectMapList` function is used to return a list of map elements.

Declaration

```
{ "Fn::SelectMapList": ["key2", [{"key1": "value1-1", "key3": "value1-3"}, {"key1": "value2-1", "key2": "value2-2"}, {"key1": "value3-1", "key2": "value3-2"}, ...] ] }
```

Parameters

- `key2` : the key to be queried in the map.
- `[{"key1": "value1-1", "key3": "value1-3"}, ...]` : the list of maps.

Return value

A list of key values for all maps in the map list.

Examples

```
{
  "Fn::SelectMapList": [
    "key2",
    [
      {"key1": "value1-1", "key3": "value1-3"},
      {"key1": "value2-1", "key2": "value2-2"},
      {"key1": "value3-1", "key2": "value3-2"}
    ]
  ]
}
```

`["value2-2","value3-2"]` is returned in this example.

Fn::Add

The `Fn::Add` function is used to sum the values of parameters.

Declaration

```
{"Fn::Add": [{"Product": "ROS"}, {"Fn": "Add"}]}
```

Parameters

- The parameters must be arranged as a list.
- The parameters in the list can be of the Number, List, or Dictionary type. All the parameters must be of the same type. The list must contain at least two parameters.

Return value

If the parameter values are numbers, sum the parameter values. If the parameter values are lists, concatenate the values. If the parameter values are dictionaries, merge the values. If the two parameters have the same key, overwrite the former parameter value with the latter.

Examples

```
{
  "Fn::Add": [
    {"Product": "ROS"},
    {"Fn": "Add"}
  ]
}
```

`{"Fn": "Add", "Product": "ROS"}` is returned in this example.

5.4.6. Mappings

The Mappings section is a key-value mapping table. When mappings are used in Resources or Outputs definitions, use `Fn::FindInMap` to find their values by using corresponding keys.

Syntax

A mapping consists of key-value pairs, where both the keys and values can be strings or numbers. Multiple mappings are separated with commas (,). Each mapping name must be unique. Mappings must be pure data and cannot parse functions.

Examples

The following example shows a correct mapping definition:

```
"Mappings": {
  "ValidMapping": {
    "TestKey1": {"TestValu1": "value1"},
    "TestKey2": {"TestValu2": "value2"},
    1234567890: {"TestValu3": "value3"},
    "TestKey4": {"TestValu4": 1234}
  }
}
```

The following example shows an incorrect mapping definition:

```

"Mappings": {
  "InvalidMapping1": {
    "ValueList": ["foo", "bar"],
    "ValueString": "baz"
  },
  "InvalidMapping2": ["foo", {"bar" : "baz"}],
  "InvalidMapping3": "foobar"
}

```

The following example shows how to use Fn::FindInMap to find the return value:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "regionParam": {
      "Description": "The region where the ECS instance is created",
      "Type": "String",
      "AllowedValues": [
        "hangzhou",
        "beijing"
      ]
    }
  },
  "Mappings": {
    "RegionMap": {
      "hangzhou": {
        "32": "m-25l0rcfjo",
        "64": "m-25l0rcfj1"
      },
      "beijing": {
        "32": "m-25l0rcfj2",
        "64": "m-25l0rcfj3"
      }
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": {
          "Fn::FindInMap": [
            "RegionMap",
            {
              "Ref": "regionParam"
            }
          ],
          "32"

```


The following example shows how to use conditions in a resource definition:

In this example, a condition is used to determine whether to create a data disk and an OSS bucket for an ECS instance based on the EnvType value.

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "EnvType": {
      "Default": "pre",
      "Type": "String"
    }
  },
  "Conditions": {
    "CreateProdRes": {
      "Fn::Equals": [
        "prod",
        {
          "Ref": "EnvType"
        }
      ]
    }
  },
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "DiskMappings": {
          "Fn::If": [
            "CreateProdRes",
            [
              {
                "Category": "cloud_efficiency",
                "DiskName": "FirstDataDiskName",
                "Size": 40
              },
              {
                "Category": "cloud_ssd",
                "DiskName": "SecondDataDiskName",
                "Size": 40
              }
            ],
            {
              "Ref": "ALIYUN::NoValue"
            }
          ]
        }
      },
      "VpcId": "vpc-2zew9pxh2yirtzqxd****"
    }
  }
}
```

```

    "SystemDiskCategory": "cloud_efficiency",
    "SecurityGroupId": "sg-2zece6wcqiejf1v****",
    "SystemDiskSize": 40,
    "ImageId": "centos_6_8_64_40G_base_2017****.vhd",
    "IoOptimized": "optimized",
    "VSwitchId": "vsw-2zed9txvy7h2srqo6****",
    "InstanceType": "ecs.n1.medium"
  }
},
"OssBucket": {
  "Type": "ALIYUN::OSS::Bucket",
  "Condition": "CreateProdRes",
  "Properties": {
    "AccessControl": "private",
    "BucketName": "myprodbucket"
  }
}
},
"Outputs": {
  "InstanceId": {
    "Value": {
      "Fn::GetAtt": [
        "WebServer",
        "InstanceId"
      ]
    }
  },
  "OssDomain": {
    "Condition": "CreateProdRes",
    "Value": {
      "Fn::GetAtt": [
        "OssBucket",
        "DomainName"
      ]
    }
  }
}
}
}

```

5.5. Resource types

5.5.1. ECS

5.5.1.1. ALIYUN::ECS::AutoSnapshotPolicy

ALIYUN::ECS::AutoSnapshotPolicy is used to create an automatic snapshot policy.

Statement

```
{  
  "Type" : "ALIYUN::ECS::AutoSnapshotPolicy",  
  "Properties" : {  
    "TimePoints" : String,  
    "RepeatWeekdays" : String,  
    "RetentionDays" : Integer,  
    "DiskIds" : String,  
    "AutoSnapshotPolicyName" : String  
  }  
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
-----------	------	----------	----------	-------------	------------

Parameter	Type	Required	Editable	Description	Constraint
TimePoints	List	Retained	Yes	The points in time at which automatic snapshots are created. Unit: hours.	<p>Value range:[0, 23], represents 24 time points from 00:00 to 23:00. For example: 1 indicating 01:00. To schedule multiple automatic snapshot creation tasks in a day, you can set the TimePoints parameter as an array.</p> <ul style="list-style-type: none"> • The maximum number of time points allowed is 24. • Use one format for multiple time points like [0, 1,... 23]. Separate time points with commas (,).

Parameter	Type	Required	Editable	Description	Constraint
RepeatWeekdays	List	Retained	Yes	The days of a week on which automatic snapshots are created.	<p>Value range:[1, 7], 1 indicates Monday. To schedule multiple automatic snapshot creation tasks in a week, you can set the RepeatWeekdays parameter as an array.</p> <ul style="list-style-type: none"> You can specify up to 7 days over a one week period. Use one format for multiple time points like [1, 2,... 7]. Separate the time points with commas (,).
RetentionDays	Integer	Retained	Yes	The number of days for which you want to retain automatic snapshots.	<p>Default value: -1. Valid values:</p> <ul style="list-style-type: none"> -1: The automatic snapshots are retained indefinitely. [1, 65536]: The automatic snapshots are retained for the specified number of days. <p>Default value: -1.</p>

Parameter	Type	Required	Editable	Description	Constraint
DiskIds	List	Retained	Yes	The ID of the destination disk. When you want to apply the automatic snapshot policy to multiple disks, you can set the diskids "d-zzzzzzzz"]. Separate multiple disk IDs with commas (,).	None
AutoSnapshotPolicyName	String	Yes	True	The name of the automatic snapshot policy.	<ul style="list-style-type: none"> • The name must be 2 to 128 characters in length • It can contain letters, digits, colons (:), underscores (_), and hyphens (-). • It cannot start with http:// or https://. <p>This parameter is empty by default.</p>

Response parameters

Fn::GetAtt

AutoSnapshotPolicyId: the ID of the automatic snapshot policy.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AutoSnapshotPolicy": {
      "Type": "ALIYUN::ECS::AutoSnapshotPolicy",
      "Properties": {
        "TimePoints": ["0"],
        "RepeatWeekdays": ["1"],
        "RetentionDays": 10,
        "DiskIds": ["<DiskId>"],
        "AutoSnapshotPolicyName": "MyAutoSnapshotPolicy"
      }
    }
  }
}
```

5.5.1.2. ALIYUN::ECS::BandwidthPackage

ALIYUN::ECS::BandwidthPackage is used to create a service plan for a NAT gateway.

Syntax

```
{
  "Type": "ALIYUN::ECS::BandwidthPackage",
  "Properties": {
    "Description": String,
    "NatGatewayId": String,
    "ZoneId": String,
    "BandwidthPackageName": String,
    "Bandwidth": Integer,
    "IpCount": Integer
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
NatGatewayId	String	Yes	No	The ID of the NAT gateway to which you want to bind the service plan.	None
Bandwidth	Integer	Yes	No	The bandwidth.	Valid values: 5 to 5000. Unit: Mbit/s. Default value: 5.

Property	Type	Required	Editable	Description	Constraint
IpCount	Integer	Yes	No	The number of public IP addresses assigned to the NAT gateway.	Valid values: 1 to 5.
Description	String	No	No	The description of the service plan.	The description must be 2 to 256 characters in length.
Zoneld	String	No	No	The ID of the zone where the NAT gateway resides.	None
BandwidthPackageName	String	No	No	The name of the service plan.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter.

Response parameters

Fn::GetAtt

- **BandwidthPackageId**: the ID of the service plan.
- **BandwidthPackageIps**: all IP addresses included in the service plan.

Examples

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "BandwidthPackage": {
      "Type": "ALIYUN::ECS::BandwidthPackage",
      "Properties": {
        "BandwidthPackageName": "pkg_2",
        "Description": "my_bandwidth",
        "NatGatewayId": "ngw-h1xox****",
        "IpCount": 2,
        "Bandwidth": 5,
        "ZoneId": "cn-beijing-c"
      }
    }
  },
  "Outputs": {
    "BandwidthPackageId": {
      "Value": {"Fn::GetAttr": ["BandwidthPackage","BandwidthPackageId"]}
    },
    "BandwidthPackageIps": {
      "Value": {"Fn::GetAttr": ["BandwidthPackage","BandwidthPackageIps"]}
    }
  }
}

```

5.5.1.3. ALIYUN::ECS::Command

ALIYUN::ECS::Command is used to create a Cloud Assistant command.

Statement

```

{
  "Type": "ALIYUN::ECS::Command",
  "Properties": {
    "Name": String,
    "WorkingDir": String,
    "CommandContent": String,
    "Timeout": Integer,
    "Type": String,
    "Description": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Name	String	Yes	True	The name of the command, which supports all character sets. The name can be up to 30 characters in length.	None
WorkingDir	String	Yes	True	The working directory on the ECS instance where the command will be run.	None
CommandContent	String	Yes	Released	The Base64-encoded content of the command. When you specify request parameters <code>Type</code> you must also specify this parameter. The parameter value must be Base64-encoded and cannot exceed 16 KB in size after encoding.	None
Timeout	String	No.	True	The timeout period that is specified for the command to run on ECS instances. Unit: seconds. If the command fails to run within the specified period, the command execution will time out and the process will be forcibly terminated. Default value: 3600.	None
Type	String	No	No	The command type. Valid values: <ul style="list-style-type: none"> RunBatScript: Creates a Bat script for a Windows instance. RunPowerShellScript: Create a PowerShell script to run on a Windows instance. RunShellScript: Creates a Shell script for Linux-based instances. 	None
Description	String	Yes	True	The description of the command, which supports all character sets. The description can be up to 100 characters in length.	None

Response parameters

Fn::GetAtt

CommandId: the ID of the command.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "WorkingDir": {
      "Type": "String",
      "Description": "The path where command will be executed in the instance."
    },
    "CommandContent": {
      "Type": "String",
      "Description": "The content of command. Content requires base64 encoding. Maximum size support 16KB."
    },
    "Type": {
      "Type": "String",
      "Description": "The type of command."
    },
    "Description": {
      "Type": "String",
      "Description": "The description of command."
    },
    "Timeout": {
      "Type": "Number",
      "Description": "Total timeout when the command is executed in the instance. Input the time unit as second. Default is 3600s."
    },
    "Name": {
      "Type": "String",
      "Description": "The name of command."
    }
  },
  "Resources": {
    "Command": {
      "Type": "ALIYUN::ECS::Command",
      "Properties": {
        "WorkingDir": {
          "Ref": "WorkingDir"
        },
        "CommandContent": {
          "Ref": "CommandContent"
        },
        "Type": {
          "Ref": "Type"
        },
        "Description": {
          "Ref": "Description"
        }
      }
    }
  }
}
```

```

"Timeout": {
  "Ref": "Timeout"
},
"Name": {
  "Ref": "Name"
}
}
},
"Outputs": {
  "CommandId": {
    "Description": "The id of command created.",
    "Value": {
      "Fn::GetAtt": [
        "Command",
        "CommandId"
      ]
    }
  }
}
}
}

```

5.5.1.4. ALIYUN::ECS::CustomImage

ALIYUN::ECS::CustomImage is used to create a custom image.

Statement

```

{
  "Type": "ALIYUN::ECS::CustomImage",
  "Properties": {
    "Description": String,
    "InstanceId": String,
    "ImageName": String,
    "ImageVersion": String,
    "SnapshotId": String,
    "Tag": List,
    "ResourceGroupId": String,
    "Platform": String,
    "DiskDeviceMapping": List,
    "Architecture": String
  }
}

```

Properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Description	String	Yes	Released	The description of the image.	The description can be up to 256 characters in length. This parameter is empty by default. It cannot start with http:// or https://.
InstanceId	String	Yes	Released	The ID of the ECS instance.	If this parameter is specified, an ECS instance will be used to create the custom image.
ImageName	String	Yes	Released	The name of the image.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens(-). It must start with a letter but cannot start with http:// or https://.
ImageVersion	String	Yes	Released	The image version.	The image version must be 1 to 40 characters in length.
SnapshotId	String	Yes	Released	The ID of the snapshot.	<ul style="list-style-type: none"> If this parameter is specified, a snapshot will be used to create the custom image. If both this parameter and the InstanceId parameter are specified, this parameter will be ignored and an instance will be used to create the custom image.
Tags	List	Erased	Released	The tags of the image.	None
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the custom image belongs.	None

Parameter	Type	Required or Not	Editable	Description	Constraint
Platform	String	Yes	Released	If you specify a data disk snapshot to be used to create the system disk of the custom image, you must use the Platform parameter to determine the release version of the operating system for the system disk.	None
DiskDeviceMapping	List	Erased	Released	The mappings between images and snapshots.	None
Architecture	String	Yes	Released	If you specify a data disk snapshot to be used to create the system disk of the custom image, you must use the Architecture parameter to determine the architecture of the system disk. Default value: x86_64.	Valid values: <ul style="list-style-type: none"> i386 x86_64

Tag syntax

```
"Tag": [
  {
    "Key": String,
    "Value": String
  }
]
```

Tag properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Key	String	Yes	Released	The tag key of the image.	The tag key cannot be a null string. The key can be up to 64 characters in length. It cannot start with aliyun or acs: and cannot contain http:// or https://.

Parameter	Type	Required or Not	Editable	Description	Constraint
Value	String	Yes	Released	The tag value of the image.	The tag value can be an empty string. The value can be up to 128 characters in length. It cannot start with aliyun or acs: and cannot contain http:// or https://.

DiskDeviceMapping

```
"DiskDeviceMapping": [
  {
    "Device": String,
    "SnapshotId": String,
    "Size": Integer,
    "DiskType": String
  }
]
```

DiskDeviceMapping properties

Parameter	Type	Required or Not	Editable	Description	Constraint
Device	String	Yes	Released	The device name of disk N in the custom image.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot that is used to create the custom image.	None
Size	String	Optional	Released	The size of disk N. Unit: GiB.	<p>Valid values: 5 to 2000.</p> <ul style="list-style-type: none"> The default value is the size of the snapshot specified by the DiskDeviceMapping.N.S snapshotId parameter. If the DiskDeviceMapping.N.S snapshotId parameter is not specified, the default disk size is 5 GiB. The disk size must be greater than or equal to the size of the snapshot specified by the DiskDeviceMapping.N.S snapshotId parameter.

Parameter	Type	Required or Not	Editable	Description	Constraint
DiskType	String	Yes	Released	The type of disk N in the custom image. You can specify this parameter to create the system disk of the custom image from a data disk snapshot. If you do not specify this parameter, the disk type is determined by the corresponding snapshot.	Valid values: <ul style="list-style-type: none"> • system: indicates a system disk. • data: indicates a data disk.

Response parameters

Fn::GetAtt

ImageId: the ID of the custom image.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "VpcId": "vpc-2zevx9ios1rszqv0a****",
        "MinAmount": 1,
        "SecurityGroupId": "sg-2ze7pxymaix640qr****",
        "ImageId": {
          "Ref": "CustomImage"
        },
      },
      "IoOptimized": "optimized",
      "SystemDisk_Description": "SystemDisk.Description",
      "SystemDisk_DiskName": "SystemDisk.DiskName",
      "SystemDisk_Category": "cloud_ssd",
      "VSwitchId": "vsw-2zei67xd9nhcqzxec****",
      "Password": "Wenqiao****",
      "InstanceType": "ecs.n1.medium",
      "MaxAmount": 1
    },
    "CustomImage": {
      "Type": "ALIYUN::ECS::CustomImage",
      "Properties": {
        "InstanceId": "i-2zefq1f3ynnrr89q****",
        "SnapshotId": "s-2ze0ibk1pvak4mw6****",
      }
    }
  }
}
```

```
"ImageName": "image-test-****",
  "ImageVersion": "verison-6-1"
}
}
},
"Outputs": {
  "CustomImage": {
    "Value": {
      "Fn::GetAtt": [
        "CustomImage",
        "ImageId"
      ]
    }
  },
  "InstanceIds": {
    "Value": {
      "Fn::GetAtt": [
        "WebServer",
        "InstanceIds"
      ]
    }
  }
}
}
```

5.5.1.5. ALIYUN::ECS::DedicatedHost

ALIYUN::ECS::DedicatedHost is used to create a dedicated host.

Statement

```
{
  "Type": "ALIYUN::ECS::DedicatedHost",
  "Properties": {
    "DedicatedHostType": String,
    "DedicatedHostName": String,
    "AutoReleaseTime": String,
    "Description": String,
    "AutoPlacement": String,
    "Tags": List,
    "ActionOnMaintenance": String,
    "NetworkAttributesSlbUdpTimeout": Integer,
    "ChargeType": String,
    "ResourceGroupId": String,
    "ZoneId": String,
    "NetworkAttributesUdpTimeout": Integer,
    "Quantity": Integer
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DedicatedHostType	String	No	No	The dedicated host type.	None
DedicatedHostName	String	Yes	Released	The name of the dedicated host.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). Must start with an uppercase or lowercase letter, and cannot start with <code>http://</code> or <code>https://</code> the beginning. It can contain digits, colons (:), underscores (_), and hyphens (-).

Parameter	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	Yes	Released	<p>The time scheduled for the dedicated host to be automatically released. If you do not specify the AutoReleaseTime parameter, the dedicated host will not be automatically released.</p> <ul style="list-style-type: none"> The minimum release time must be at least 30 minutes after the current time. The maximum release time must be at most three years from the current time. If the value of <code>ss</code> is not <code>00</code>, the start time is automatically rounded down to the nearest minute based on the value of <code>mm</code>. 	None
Description	String	Yes	Released	The description of the dedicated host.	None
ZoneId	String	Yes	Released	<p>The ID of the zone where the dedicated host resides.</p> <p>This parameter is empty by default. If this parameter is not specified, the system will automatically select a zone.</p>	None
ChargeType	String	Yes	Released	The billing method of the dedicated host.	Valid values: PostPaid and pay-as-you-go.

Parameter	Type	Required	Editable	Description	Constraint
AutoPlacement	String	Yes	Released	Specifies whether to add the dedicated host to the resource pool for automatic deployment. If you do not specify a DedicatedHostId when you create an instance on a DDH, Alibaba Cloud automatically selects a DDH from the resource pool to host the instance.	<p>Valid values:</p> <ul style="list-style-type: none"> on off <p> Note</p> <p>If you do not specify this parameter, the dedicated host is added to the automatic deployment resource pool.</p> <p>If you do not want to add the dedicated host to the resource pool for automatic deployment, set the value to off.</p>
Tags	List	Erased	Released	The custom tags of the instance.	<p>A maximum of 20 tags are supported. The format is as follows:</p> <pre>[{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}]</pre>
ActionOnMaintenance	String	Yes	Released	The method used to migrate the instances on the DDH when the DDH fails or needs to be repaired online.	<p>Valid values:</p> <ul style="list-style-type: none"> Migrate: specifies that the instances are migrated to another physical server and restarted. Stop: specifies that all the instances on the DDH are stopped. If the DDH cannot be repaired, the instances are migrated to another physical server and restarted. <p>The default value is "Migrate" for a dedicated host and "Stop" for a local disk.</p>

Parameter	Type	Required	Editable	Description	Constraint
NetworkAttributesSlbUdpTimeout	String	Optional	Released	The timeout period for a UDP session.	Valid values: 15 to 310. Unit: seconds.
ResourceGroupID	String	Yes	Released	The ID of the resource group to which the dedicated host belongs.	None
NetworkAttributesUdpTimeout	String	Optional	Released	The timeout period for UDP sessions that users can access for cloud services running on the dedicated host.	Valid values: 15 to 310. Unit: seconds.
Quantity	String	Optional	Released	The number of DDHs that you want to create this time.	Valid values: 1 to 100. Default value: 1

Response parameters

Fn::GetAtt

- OrderId: the ID of the order.
- DedicatedHostIds: the list of host IDs.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "AutoRenewPeriod": {
      "Type": "Number",
      "Description": "The time period of auto renew. When the parameter InstanceChargeType is PrePaid, it will take effect. It could be 1, 2, 3, 6, 12. Default value is 1.",
      "AllowedValues": [
        1,
        2,
        3,
        6,
        12
      ],
      "Default": 1
    },
    "Description": {
      "Type": "String",
      "Description": "The description of host."
    },
    "ZoneId": {
      "Type": "String",
      "Description": "The zone to create the host."
    }
  }
}
```

```

},
"DedicatedHostName": {
  "Type": "String",
  "Description": "The name of the dedicated host, [2, 128] English or Chinese characters. It must begin with an uppercase/uppercase letter or a Chinese character, and may contain numbers, '_' or '-'. It cannot begin with http:// or https://."
},
"ChargeType": {
  "Type": "String",
  "Description": "Instance Charge type, allowed value: Prepaid and Postpaid. If specified Prepaid, please ensure you have sufficient balance in your account. Or instance creation will be failure. Default value is Postpaid.",
  "AllowedValues": [
    "PrePaid",
    "PostPaid"
  ],
  "Default": "PostPaid"
},
"AutoRenew": {
  "Type": "String",
  "Description": "Whether renew the fee automatically? When the parameter InstanceChargeType is PrePaid, it will take effect. Range of value:True: automatic renewal.False: no automatic renewal. Default value is False.",
  "AllowedValues": [
    "True",
    "False"
  ],
  "Default": "False"
},
"Period": {
  "Type": "Number",
  "Description": "Prepaid time period. Unit is month, it could be from 1 to 9 or 12, 24, 36, 48, 60. Default value is 1.",
  "AllowedValues": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8,
    9,
    12,
    24,
    36,
    48,
    60
  ],
  "Default": 1
}

```

```

    },
    "DedicatedHostType": {
      "Type": "String",
      "Description": "The instance type of host."
    },
    "PeriodUnit": {
      "Type": "String",
      "Description": "Unit of prepaid time period, it could be Week/Month. Default value is Month.",
      "AllowedValues": [
        "Week",
        "Month"
      ],
      "Default": "Month"
    },
    "AutoReleaseTime": {
      "Type": "String",
      "Description": "Auto release time for created host, Follow ISO8601 standard using UTC time. format is 'yyyy-MM-ddT
HH:mm:ssZ'. Not bigger than 3 years from this day onwards"
    }
  },
  "Resources": {
    "Host": {
      "Type": "ALIYUN::ECS::DedicatedHost",
      "Properties": {
        "Description": {
          "Ref": "Description"
        },
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "DedicatedHostName": {
          "Ref": "DedicatedHostName"
        },
        "ChargeType": {
          "Ref": "ChargeType"
        },
        "DedicatedHostType": {
          "Ref": "DedicatedHostType"
        },
        "PeriodUnit": {
          "Ref": "PeriodUnit"
        },
        "AutoReleaseTime": {
          "Ref": "AutoReleaseTime"
        }
      }
    }
  }
}

```

```

}
},
"Outputs": {
  "OrderId": {
    "Description": "The order id list of created instance.",
    "Value": {
      "Fn::GetAtt": [
        "Host",
        "OrderId"
      ]
    }
  },
  "DedicatedHostIds": {
    "Description": "The host id list of created hosts",
    "Value": {
      "Fn::GetAtt": [
        "Host",
        "DedicatedHostIds"
      ]
    }
  }
}
}
}
}
}

```

5.5.1.6. ALIYUN::ECS::Disk

ALIYUN::ECS::Disk is used to create an ECS Disk.

Statement

```

{
  "Type": "ALIYUN::ECS::Disk",
  "Properties": {
    "DiskName": String,
    "Description": String,
    "Tags": List,
    "AutoSnapshotPolicyId": String,
    "Encrypted": Boolean,
    "ZoneId": String,
    "ResourceGroupId": String,
    "SnapshotId": String,
    "DiskCategory": String,
    "PerformanceLevel": String,
    "DeleteAutoSnapshot": Boolean,
    "Size": Integer
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
ZoneId	String	No	No	The ID of the zone where the instance resides.	None
DiskName	String	Yes	Released	The name of the disk.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length And can contain letters, digits, periods, underscores (_), and hyphens (-). It cannot start with http:// or https://. The disk name will be displayed in the ECS console.
Description	String	Yes	Released	The description of the disk.	<ul style="list-style-type: none"> The description must be 2 to 256 characters in length. Cannot http:// or https:// the beginning. The disk description will be displayed in the ECS console.
Tags	List	Erased	Released	The custom tags of the instance.	Up to four tags are supported. Example values: [{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}].

Parameter	Type	Required	Editable	Description	Constraint
DiskCategory	String	Yes	Released	The type of the data disk.	Value range <ul style="list-style-type: none"> cloud: indicates a basic disk. cloud_efficiency: indicates an ultra disk. cloud_ssd: indicates a standard SSD. cloud_essd: enhanced SSD (ESSD) Default value: cloud.
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	<ul style="list-style-type: none"> If both this parameter and 'Size' are specified, the value of this parameter prevails. The actual size of the created disk is the size of the specified snapshot. Snapshots created on or before July 15, 2013 cannot be used to create disks.
PerformanceLevel	String	Yes	Released	Specifies the performance level of an ESSD when you create the ESSD.	Default value: PL1. Valid values: <ul style="list-style-type: none"> PL1: A single enhanced SSD delivers up to 50,000 random read/write IOPS. PL2: A single ESSD delivers up to 100,000 random read/write IOPS. PL3: maximum random read/write IOPS of 100,000 per disk.

Parameter	Type	Required	Editable	Description	Constraint
Size	String	Optional	Released	The size of the disk. Unit: GiB. The value of this parameter must be equal to or greater than the capacity of the specified snapshot.	Valid values: <ul style="list-style-type: none"> cloud: 5 to 2000 cloud_efficiency: 20 to 32768 cloud_ssd: 20 to 32768 cloud_essd: 20 to 32768
AutoSnapshotPolicyId	String	Yes	Released	The ID of each automatic snapshot policy.	None
Encrypted	Boolean	Erased	Released	Specifies whether to encrypt the disk.	Valid values: <ul style="list-style-type: none"> true false Default value: false.
DeleteAutoSnapshot	Boolean	Erased	Released	Specifies whether to delete the automatic snapshots of the disk when the disk is released.	Valid values: <ul style="list-style-type: none"> true false Default value: true.

Tags syntax

```
"Tags" : [
  {
    "Value" : String,
    "Key" : String
  }
]
```

Tags properties

Parameter	Type	Required	Editable
Key	String	No	No
Value	String	Yes	Released

Response parameters

Fn::GetAtt

- **DiskId:** the ID of the disk.
- **Status:** The Status of the disk.

Sample request

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "DataDisk": {
      "Type": "ALIYUN::ECS::Disk",
      "Properties": {
        "Size": 10,
        "ZoneId": "cn-beijing-a",
        "DiskName": "DataDisk",
        "Description": "ECSDataDisk"
      }
    }
  },
  "Outputs": {
    "DiskId": {
      "Value" : {"Fn::GetAtt": ["DataDisk","DiskId"]}
    },
    "Status": {
      "Value" : {"Fn::GetAtt": ["DataDisk","Status"]}
    }
  }
}
```

5.5.1.7. ALIYUN::ECS::DiskAttachment

ALIYUN::ECS::DiskAttachment is used to attach an ECS disk.

Statement

```
{
  "Type" : "ALIYUN::ECS::DiskAttachment",
  "Properties" : {
    "DiskId" : String,
    "InstanceId" : String,
    "Device" : String,
    "DeleteWithInstance" : String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	No	The ID of the instance.	None

Parameter	Type	Required	Editable	Description	Constraint
DiskId	String	No	No	The ID of the disk.	The disk and the ECS instance must belong to the same zone.
Device	String	Yes	Released	The name of the disk.	If you do not set this parameter, the system will automatically allocate a device name in alphabetical order from /dev/xvdb to /dev/xvdz.
DeleteWithInstance	Boolean	Erased	Released	Specifies whether the disk is to be released together with the instance.	Valid values: <ul style="list-style-type: none"> • true: The disk will be released when the instance is released. • false: The disk will be retained when the instance is released.

Response parameters

Fn::GetAtt

- DiskId: the ID of the disk.
- Status: The Status of the disk.
- The name of the Device: disk.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DiskAttachment": {
      "Type": "ALIYUN::ECS::DiskAttachment",
      "Properties": {
        "InstanceId": {
          "Ref": "InstanceId"
        },
        "Device": {
          "Ref": "Device"
        }
      },
      "DeleteWithInstance": {
        "Ref": "DeleteWithInstance"
      }
    }
  }
}
```

```

    },
    "DiskId": {
      "Ref": "DiskId"
    }
  }
},
"Parameters": {
  "InstanceId": {
    "Type": "String",
    "Description": "The ID of the instance to attach the disk."
  },
  "Device": {
    "Type": "String",
    "Description": "The device where the volume is exposed on the instance. The device name could be /dev/xvd[a-z]. If this parameter is not specified, the default value will be used."
  },
  "DeleteWithInstance": {
    "Type": "Boolean",
    "Description": "If this parameter is set to true, the disk will be deleted while the instance is deleted. If this parameter is set to false, the disk will be retained after the instance is deleted.",
    "AllowedValues": [
      "True",
      "true",
      "False",
      "false"
    ]
  },
  "DiskId": {
    "Type": "String",
    "Description": "The ID of the disk to be attached."
  }
},
"Outputs": {
  "Status": {
    "Description": "The disk status now.",
    "Value": {
      "Fn::GetAtt": [
        "DiskAttachment",
        "Status"
      ]
    }
  }
},
"Device": {
  "Description": "The device where the volume is exposed on the ECS instance.",
  "Value": {
    "Fn::GetAtt": [

```

```

        "DiskAttachment",
        "Device"
    ]
}
},
"DiskId": {
    "Description": "The ID of the created disk.",
    "Value": {
        "Fn::GetAtt": [
            "DiskAttachment",
            "DiskId"
        ]
    }
}
}
}
}
}
}

```

5.5.1.8. ALIYUN::ECS::ForwardEntry

ALIYUN::ECS::ForwardEntry is used to configure the DNAT table of a NAT Gateway.

Statement

```

{
    "Type": "ALIYUN::ECS::ForwardEntry",
    "Properties": {
        "ExternalIp": String,
        "ExternalPort": String,
        "ForwardTableId": String,
        "InternalIp": String,
        "IpProtocol": String,
        "InternalPort": String
    }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ExternalIp	String	No	No	The public IP address of the NAT gateway.	It must be an IP address that is included in the shared NAT Gateway of the bandwidth plan to which the DNAT table belongs.

Parameter	Type	Required	Editable	Description	Constraint
ExternalPort	String	No	No	The public port number.	Valid values: 1 to 65535.
ForwardTableId	String	No	No	The ID of the DNAT table.	None
InternalIp	String	No	No	The destination IP address to which the request is forwarded.	This IP address is a private IP address.
IpProtocol	String	No	No	The type of the protocol.	Valid values: TCP, UDP, and Any.
InternalPort	String	No	No	The destination private port.	Valid values: 1 to 65535.

Response parameters

`Fn::GetAtt`

`ForwardEntryId`: the ID of each entry in the DNAT table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ForwardEntry": {
      "Type": "ALIYUN::ECS::ForwardEntry",
      "Properties": {
        "ForwardTableId": "my_forwardtable",
        "ExternalIp": "101.201.XX.XX",
        "ExternalPort": "8080",
        "IpProtocol": "TCP",
        "InternalIp": "10.2.XX.XX",
        "InternalPort": "80"
      }
    }
  },
  "Outputs": {
    "ForwardEntryId": {
      "Value": {"Fn::GetAttr": ["ForwardEntry", "ForwardEntryId"]}
    }
  }
}
```

5.5.1.9. ALIYUN::ECS::Instance

`ALIYUN::ECS::Instance` is used to create an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::Instance",
  "Properties": {
    "RamRoleName": String,
    "IoOptimized": String,
    "PrivateIpAddress": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "Description": String,
    "Tags": List,
    "HostName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "Password": String,
    "InstanceType": String,
    "SystemDiskCategory": String,
    "UserData": String,
    "SystemDiskSize": Number,
    "ZoneId": String,
    "VpcId": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String,
    "DiskMappings": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
ImageId	String	No	Yes	The ID of the image used to start the ECS instance. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	<p>When editing a template, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID.</p> <p>You can use the wildcard character (*) to represent part of an image ID.</p> <p>Take all Ubuntu public images provided by Alibaba Cloud as an example. You can use one of the following methods to specify the public image ID for the ECS instance:</p> <ul style="list-style-type: none"> If you enter ubuntu, the system matches it with the following ID: ubuntu16_0402_64_20G_alibase_20170818.vhd If you enter ubuntu_14, the system matches it with the following ID: ubuntu_14_0405_64_20G_alibase_20170824.vhd If you enter ubuntu*14*32, the system matches it with the following ID: ubuntu_14_0405_32_40G_alibase_20170711.vhd If you enter ubuntu_16_0402_32, the system matches it with the following ID: ubuntu_16_0402_32_40G_alibase_20170711.vhd
InstanceType	String	No	No	The type of the ECS instance.	None
SecurityGroupId	String	No	No	The ID of the security group to which the created instance will belong.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).

Parameter	Type	Required	Editable	Description	Constraint
Password	String	Yes	Released	The password used to log on to the ECS instance.	<p>The characters in length is 8 to 30.</p> <p>And must contain at least one of the following character types: uppercase letters, lowercase letters, digits, and special character.</p> <p>Special characters include () ` ~ ! @ # \$ % ^ & * - + = { } [] ; ' < > , . ? / - / - If you specify the Password parameter in the API request, use HTTPS to secure the API and protect your password.</p>
HostName	String	Yes	Released	The hostname of the instance.	<p>The password must be at least 2 characters in length.</p> <p>It cannot And hyphens (-) cannot start or end the hostname and cannot be used consecutively.</p> <p>On Windows, the hostname can be up to 15 characters in length and can contain letters, digits, and hyphens (-). It cannot contain periods (.) and cannot be composed of only digits.</p> <p>On other OSs such as Linux, the hostname can contain a maximum of 30 characters, including periods (.), each segment can contain uppercase or lowercase letters, digits, and hyphens (-).</p>
PrivateIpAddress	String	Yes	Released	The private IP address of an ECS instance in a VPC. The specified IP address must not be used by other instances in the VPC.	None
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet.	<p>Valid values: 1 to 100.</p> <p>Default value: 100.</p> <p>Unit: Mbit/s.</p>

Parameter	Type	Required	Editable	Description	Constraint
IoOptimized	String	Yes	Released	Specifies whether an I/O optimized instance is created.	Valid values: <ul style="list-style-type: none"> • none (non-I/O optimized) • optimized Default value: none.
DiskMappings	List	Erased	Released	The data disks to be attached to the instance.	A maximum of 16 disks can be attached to each instance.
SystemDiskCategory	String	Yes	Released	The type of the system disk.	Valid values: <ul style="list-style-type: none"> • cloud • cloud_efficiency • cloud_ssd • ephemeral_ssd
SystemDiskDescription	String	Yes	Released	The description of the ECS instance system disk.	None
SystemDiskDiskName	String	Yes	Released	The name of the ECS instance system disk.	None
SystemDiskSize	Number	No.	True	The size of the system disk. Unit: GB.	Valid values: 40 to 500. If a custom image is used to create a system disk, make sure that the size of the system disk is greater than that of the custom image.
Tags	List	Erased	Released	The custom tags of the instance.	A maximum of 20 tags are supported. The format is as follows: <pre>[{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}]</pre>
UserData	String	Yes	Released	The user data that you provide when you create ECS instances.	The user data can be up to 16KB in size. You do not need to use Base64. you must use special characters. \ Escape.
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
HpcClusterId	String	Yes	Released	The ID of the HPC cluster to which the ECS instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	Yes	Released	The ID of the VPC to which the ECS instance belongs.	None
VSwitchId	String	Yes	Released	The ID of the VSwitch for the ECS instance.	None
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to created ECS instances.	<ul style="list-style-type: none"> For Windows-based instances, this parameter is empty by default. In the Linux, if this parameter is specified, the Password content is still set to the instance, but the Password logon method is disabled by default. The key pair is used to verify the logon.
RamRoleName	String	Yes	Released	The RAM role name of the instance. You can call the ListRoles operation to query the role name.	None
DeletionProtection	Boolean	Erased	Released	The release protection property of created instances. It specifies whether the instances can be released from the ECS console or through the DeleteInstance operation.	Valid values: <ul style="list-style-type: none"> True False
DeploymentSetId	String	Yes	True	Deployment Set ID.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: <ul style="list-style-type: none"> cloud cloud_efficiency cloud_ssd ephemeral_ssd Default value: cloud.
DiskName	String	Yes	Released	The name of data disk N.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	Valid values: 2 to 256. Default value: Null.
Device	String	Yes	Released	The device name of the data disk.	If you do not specify this parameter, the system automatically allocates a device name in alphabetical order from /dev/xvdb to /dev/xvdz.
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None

Parameter	Type	Required	Editable	Description	Constraint
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- **InstanceId:** the ID of the instance. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- **PrivateIp:** The private IP address of the instance in a VPC. This parameter takes effect only when the **NetworkType** parameter is set to VPC.
- **InnerIp:** The private IP address of the instance in a Classic network. This parameter takes effect only when the **NetworkType** parameter is set to Classic.
- **PublicIp:** the public IP address of the instance in a Classic network. This parameter takes effect only when the **NetworkType** parameter is set to Classic.
- **ZoneId:** the zone ID.
- **HostName:** the hostname of the instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::Instance",
      "Properties": {
        "ImageId": "m-25l0rc****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "Tags": [{
          "Key": "tiantt",
          "Value": "ros"
        }],
        "Key": "tiantt1",
        "Value": "ros1"
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Value": {"get_attr": ["WebServer", "InstanceId"]}
    },
    "PublicIp": {
      "Value": {"get_attr": ["WebServer", "PublicIp"]}
    }
  }
}
```

5.5.1.10. ALIYUN::ECS::InstanceClone

ALIYUN::ECS::InstanceClone is used to clone an ECS instance.

Statement

```

{
  "Type": "ALIYUN::ECS::InstanceClone",
  "Properties": {
    "DeletionProtection": Boolean,
    "DiskMappings": List,
    "LoadBalancerIdToAttach": String,
    "Description": String,
    "BackendServerWeight": Integer,
    "Tags": List,
    "SecurityGroupId": String,
    "RamRoleName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "SpotPriceLimit": String,
    "InstanceChargeType": String,
    "SourceInstanceId": String,
    "Period": Number,
    "SpotStrategy": String,
    "Password": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "ZoneId": String,
    "KeyPairName": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
SourceInstanceId	String	No	No	The ID of the ECS instance to be cloned.	The clone operation clones the specified instance, including its instance type, image, bandwidth billing method, bandwidth limit, and network type. If the source ECS instance belongs to multiple security groups, the cloned instance is added only to the first of these security groups.
BackendServerWeight	String	Optional	Released	The weight of the ECS instance in the Server Load Balancer instance.	Value range:[0, 100]. Default value: 100.

Parameter	Type	Required	Editable	Description	Constraint
LoadBalancerIdToAttach	String	Yes	Released	The ID of the SLB instance to which the ECS instance is to be attached.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.
ImageId	String	Yes	True	The ID of the image used to start created instances. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	<p>When editing a template, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID.</p> <p>You can use the wildcard character (*) to represent part of an image ID.</p> <p>Take all Ubuntu public images provided by Alibaba Cloud as an example. You can use one of the following methods to specify the public image ID for the ECS instance:</p> <p>If you enter ubuntu,</p> <p>the system matches it with the following ID: ubuntu16_0402_64_20G_alibase_20170818.vhd</p> <p>If you enter ubuntu_14,</p> <p>the system matches it with the following ID: ubuntu_14_0405_64_20G_alibase_20170824.vhd</p> <p>If you enter ubuntu*14*32,</p> <p>the system matches it with the following ID: ubuntu_14_0405_32_40G_alibase_20170711.vhd</p> <p>If you enter ubuntu_16_0402_32,</p> <p>the system matches it with the following ID: ubuntu_16_0402_32_40G_alibase_20170711.vhd</p>
SecurityGroupId	String	Yes	Released	The ID of the security group to which the created instance will belong.	None

Parameter	Type	Required	Editable	Description	Constraint
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Password	String	Yes	Released	The password used to log on to the ECS instance.	<p>The password must be 8 to 30 characters in length.</p> <p>The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</p> <p>Special characters include () ` ~ ! @ # \$ % ^ & * - + = { } [] : ; < > , . ? /</p> <p>If you specify the password parameter in the API request, use HTTPS to secure the API and protect your password.</p>
DiskMappings	List	Erased	Released	The disks to be attached to created instances.	A maximum of 16 disks can be attached to each instance.
Tags	List	Erased	Released	The custom tags of the instance.	A maximum of 20 tags can be specified in the <code>{{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}}</code> .
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
InstanceChargeType	String	Yes	Released	The billing method of the new ECS instance.	<p>Valid values: PrePaid and PostPaid.</p> <p>Default value: Postpaid. If you set this parameter to Prepaid, make sure that you have sufficient balance in your account. Otherwise, the instance fails to be created.</p>

Parameter	Type	Required	Editable	Description	Constraint
Period	Number	Erased	Released	The subscription period of the new ECS instance. This parameter is required when the InstanceChargeType parameter is set to PrePaid. This parameter is ignored when the InstanceChargeType parameter is set to PostPaid.	Valid values: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, and 36. Unit: month.
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to created ECS instances.	For Windows-based instances, this parameter is empty by default. In the Linux, if this parameter is specified, the Password content is still set to the instance, but the Password logon method is disabled by default. The key pair is used to verify the logon.
RamRoleName	String	Yes	Released	The RAM role name of the instance. You can call the ListRoles operation to query the role name.	None
SpotPriceLimit	String	Yes	Released	The maximum hourly price of the instance.	Parameter SpotStrategy this parameter takes effect only when the value is SpotWithPriceLimit.
SpotStrategy	String	Yes	Released	The bidding policy for the pay-as-you-go instance.	This parameter is valid only when the InstanceChargeType parameter is set to PostPaid. Valid values: NoSpot: applies to regular pay-as-you-go instances. SpotWithPriceLimit: applies to preemptible instances with a maximum hourly price. SpotAsPriceGo: applies to pay-as-you-go instances priced at the market price at the time of purchase. Default value: NoSpot.
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet. Unit: Mbit/s.	Valid values: 1 to 200.

Parameter	Type	Required	Editable	Description	Constraint
DeletionProtection	Boolean	Erased	Released	Specifies whether to enable instance release protection in the console or by calling an API operation. (DeleteInstance) Release instances.	Valid values: true and false.

DiskMappings syntax

```
"DiskMappings": [
{
  "Category": String,
  "DiskName": String,
  "Description": String,
  "Device": String,
  "SnapshotId": String,
  "Size": String
}
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: cloud, cloud_efficiency, cloud_ssd, and ephemeral_ssd Default value: cloud.
DiskName	String	Yes	Released	Disk name.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	The description must be 2 to 256 characters in length. This parameter is empty by default.
Device	String	Yes	Released	The device name of the data disk.	If you do not specify this parameter, the system automatically allocates a device name in alphabetical order from /dev/xvdb to /dev/xvdz.

Parameter	Type	Required	Editable	Description	Constraint
SnapshotId	String	Yes	Released	The ID of the snapshot used to create the data disk.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	The tag key, which cannot be an empty string. It can be up to 64 characters in length, cannot start with <code>acs:</code> or <code>aliyun</code> , and cannot contain <code>http://</code> or <code>https://</code> .	None
Value	String	Yes	Released	The tag value, which can be an empty string. It can be up to 128 characters in length and cannot start with <code>acs:</code> or <code>aliyun</code> . It cannot contain <code>http://</code> or <code>https://</code> .	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- **InstanceId:** the ID of the instance. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- **PrivateIp:** The private IP address of the instance in a VPC. This parameter takes effect only when the `NetworkType` parameter is set to `VPC`.
- **InnerIp:** The private IP address of the instance in a Classic network. This parameter takes effect only when the `NetworkType` parameter is set to `Classic`.
- **PublicIp:** the public IP address of the instance in a Classic network. This parameter takes effect only when the `NetworkType` parameter is set to `Classic`.
- **ZoneId:** the zone ID.
- **HostName:** the hostname of the instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceClone",
      "Properties": {
        "SourceInstanceId": "i-25zsk****",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "DiskMappings": [
          {"Size": 10, "Category": "cloud"},
          {"Size": 10, "Category": "cloud", "SnapshotId": "s-25wsw****"}
        ]
      }
    }
  },
  "Outputs": {
    "InstanceId": {
      "Value": {"get_attr": ["WebServer", "InstanceId"]}
    },
    "PublicIp": {
      "Value": {"get_attr": ["WebServer", "PublicIp"]}
    }
  }
}
```

5.5.1.11. ALIYUN::ECS::InstanceGroup

ALIYUN::ECS::InstanceGroup is used to create an ECS instance group.

Syntax

```
{
  "Type": "ALIYUN::ECS::InstanceGroup",
  "Properties": {
    "SystemDiskAutoSnapshotPolicyId": String,
    "DedicatedHostId": String,
    "LaunchTemplateName": String,
    "RamRoleName": String,
    "IoOptimized": String,
    "PrivateIpAddress": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "Description": String,
    "Tags": List,
    "HostName": String,
    "ImageId": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "EniMappings": List,
    "Password": String,
    "InstanceType": String,
    "MaxAmount": Integer,
    "AutoReleaseTime": String,
    "SystemDiskCategory": String,
    "UserData": String,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "SystemDiskSize": Number,
    "ZoneId": String,
    "VpcId": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "Ipv6AddressCount": Integer,
    "SecurityGroupId": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String,
    "Ipv6Addresses": List,
    "NetworkType": String,
    "DiskMappings": List,
    "SystemDiskPerformanceLevel": String
  }
}
```

Properties

Attribute	Type	Required	Editable	Description	Constraint
ResourceGroupID	String	No	Yes	The ID of the resource group to which a created instance belongs.	None.
HpcClusterId	String	No	Yes	The ID of the HPC cluster to which a created instance belongs.	None.
MaxAmount	Integer	Supported	Yes	The maximum number of ECS instances that can be created at a time.	Valid values: 1 to 100. The MaxAmount parameter must be set to a value greater than or equal to the value of MinAmount.
MinAmount	String	Yes	Yes	The minimum number of ECS instances that can be created at a time.	Valid values: 1 to 100. The MinAmount parameter must be set to a value less than or equal to the value of MaxAmount.
Description	String	No	Yes	The description of created instances.	The description can be up to 256 characters in length.
InstanceType	String	Yes	Yes	The type of the ECS instance.	None.
ImageId	String	Yes	Yes	The ID of the image used to start an ECS instance. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	You can specify a partial public image ID instead of providing the complete ID. The following example shows how to use a CNAME record: <ul style="list-style-type: none"> • Ubuntu is specified and ubuntu_16_0402_64_20G_alibase_20170818.vhd are matched. • If ubuntu1432 is specified, ubuntu_14_0405_32_40G_alibase_20170711.vhd is matched.

Attribute	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	No	No	The ID of the security group to which created instances belong.	None.
InstanceName	String	No	No	The name of an instance.	The names can be up to 128 characters in length. It can contain English letters, Chinese characters, digits, underscores (_), periods (.), and hyphens (-). By <code>name_prefix[begin_number,bits]name_suffix</code> format to specify different instance name for each ECS instance.
Password	String	No	Yes	The password used to log on to created ECS instances.	<ul style="list-style-type: none"> The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The following special characters are supported: <code>:()'~!@#\$%^&* - + = { } [] ; ' < > , . ? /</code> <p>If you specify the Password parameter in the API request, use HTTPS to secure the API and protect your password.</p>
HostName	String	No	No	The hostname of created ECS instances.	The hostname must contain at least two characters in length. The periods and hyphens (-) cannot start or end the hostname and cannot be used together.

Attribute	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	No	No	The time scheduled for created ECS instances to be automatically released.	The time format must comply with ISO8601 specifications, for example, "yyyy-MM-ddTHH:mm:ssZ". The maximum release time must be within three years from the current time.
PrivateIpAddress	String	No	No	The private IP address of an ECS instance in a VPC.	The specified IP address must not be used by other instances in the VPC.
DiskMappings	List	No	Yes	The data disks to be attached to created instances.	None.
InternetMaxBandwidthIn	Integer	No	No	The maximum inbound bandwidth from the Internet.	Unit: Mbit/s. Valid values: 1 to 100 Default value: 100.
IoOptimized	String	No	No	Specifies whether the created instances are I/O optimized.	Valid values: • none (non-I/O optimized) • optimized Default value: none.
SystemDiskCategory	String	No	Yes	The category of the system disk.	Valid values: cloud: basic disk cloud_efficiency: the ultra disk cloud_ssd: standard SSDs cloud_essd: enhanced SSD ephemeral_ssd: local SSDs
SystemDiskDescription	String	No	Yes	The description of the ECS instance system disk.	None.
SystemDiskDiskName	String	No	Yes	The name of the ECS instance system disk.	None.

Attribute	Type	Required	Editable	Description	Constraint
SystemDiskSize	Number	No	Yes	The size of the system disk.	Valid values: 40 to 500. If a custom image is used to create a system disk, make sure that the size of the system disk is greater than that of the custom image.
Tags	List	No	Yes	The custom tags of a created instance.	A maximum of 20 tags are supported. The format is as follows: [{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}].
UserData	String	No	Yes	The user data that you provide when you create ECS instances.	The user data can be up to 16 KB in size. You do not need to use Base64 for transcoding. Special characters need to be escaped.
ZoneId	String	No	No	The zone ID of the disk.	None.
VpcId	String	No	No	The ID of the VPC.	None.
VSwitchId	String	No	No	The ID of the VSwitch for the ECS instance.	None.
KeyPairName	String	No	Yes	The name of the key pair that is used to connect to created ECS instances.	For Windows-based ECS instances, this parameter is ignored and it is empty by default. For Linux-based ECS instances, the Password parameter still takes effect if this parameter is specified. However, logon by password is disabled, and the KeyPairName value is used.
RamRoleName	String	No	Yes	The name of the instance RAM role.	You can call the ListRoles operation to query the role name.
DedicatedHostId	String	No	No	The ID of the dedicated host.	None.

Attribute	Type	Required	Editable	Description	Constraint
LaunchTemplateName	String	No	Yes	The name of the launch template for the instance.	None.
EniMappings	List	No	Yes	The elastic network interfaces (ENIs) to be attached to created instances.	Only one ENI can be attached to each instance.
LaunchTemplateId	String	No	Yes	The ID of the launch template.	None.
LaunchTemplateVersion	String	No	Yes	The version of the launch template.	If you do not specify a version, the default version is used.
NetworkType	String	No	No	The network type of created ECS instances.	Valid values: <ul style="list-style-type: none"> vpc classic Default value: classic.
DeletionProtection	Boolean	No	No	The release protection attribute of the instance. It specifies whether you can use the ECS console or call the DeleteInstance operation to release the instance.	Valid values: <ul style="list-style-type: none"> true false
DeploymentSetId	String	No	Yes	Deployment Set ID.	None.

DiskMappings syntax

```

"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "PerformanceLevel": String,
    "AutoSnapshotPolicyId": String
  }
]

```

DiskMappings properties

Attribute	Type	Required	Editable	Description	Constraint
Size	String	Yes	No	The size of a data disk.	Unit: GB.
Category	String	No	No	The type of the data disk.	Valid values: <ul style="list-style-type: none"> cloud cloud_efficiency cloud_ssd cloud_essd ephemeral_ssd For I/O optimized instances, the default value is cloud_efficiency. For non-I/O optimized instances, the default value is cloud.
DiskName	String	No	No	The name of data disk N.	The name can be up to 128 characters in length. It can contain English letters, Chinese characters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	No	No	The description of data disk N.	The description must be 2 to 256 characters in length. The description cannot start with <code>http://</code> or <code>https://</code> .
Device	String	No	No	The device name of the data disk.	The system allocates a device name in alphabetical order from <code>/dev/xvda</code> to <code>/dev/xvdz</code> .

Attribute	Type	Required	Editable	Description	Constraint
SnapshotId	String	No	No	The ID of the snapshot.	None.
Encrypted	Boolean	No	No	Specifies whether to encrypt the data disk.	Valid values: <ul style="list-style-type: none"> • true • false Default value: false
KMSKeyId	String	No	No	The ID of the KMS key corresponding to the data disk.	None.
AutoSnapshotPolicyId	String	No	Yes	The ID of the automatic snapshot policy.	None.
PerformanceLevel	String	No	No	The performance level of the enhanced SSD used as the data disk.	<ul style="list-style-type: none"> • (Default): Maximum random read /write IOPS of 50,000 per disk • PL2: A single enhanced SSD delivers up to 100,000 random read/write IOPS. • PL3: A single enhanced SSD delivers up to 1,000,000 random read/write IOPS.

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Attribute	Type	Required	Editable	Description	Constraint
Key	String	Yes	No	The key of tag N.	It must be 1 to 128 characters in length, and cannot start with <code>aliyun</code> and <code>acs:</code> beginning, cannot contain <code>http://</code> or <code>https://</code> .

Attribute	Type	Required	Editable	Description	Constraint
Value	String	No	No	The value of tag N.	It must be 0 to 128 characters in length and cannot start with <code>aliyun</code> and <code>acs:</code> beginning, cannot contain <code>http://</code> or <code>https://</code> .

EniMappings syntax

```
"EniMappings": [
  {
    "SecurityGroupId": String,
    "VSwitchId": String,
    "Description": String,
    "NetworkInterfaceName": String,
    "PrimaryIpAddress": String
  }
]
```

EniMappings properties

Attribute	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	Yes	The ID of the security group to which an instance belongs.	The security group and the instance must be in the same VPC.
VSwitchId	String	Yes	No	The ID of the VSwitch to which the instance is connected.	None.
Description	String	No	Yes	The description of the ENI.	It can contain 2 to 256 English letters or Chinese character. It cannot start with <code>http://</code> or <code>https://</code> the beginning.

Attribute	Type	Required	Editable	Description	Constraint
NetworkInterfaceName	String	No	Yes	The ENI name.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length Must start with an uppercase or lowercase letter, and cannot start with <code>http://</code> or <code>https://</code> the beginning. It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.
PrimaryIpAddress	String	No	No	The primary private IP address of ENI.	The specified IP address must be available within the CIDR block of the VSwitch. If this parameter is not specified, an available IP address in the VSwitch CIDR block will be selected at random.

Return value

Fn::GetAtt

- InstanceIds:** the IDs of created instances in the ECS instance group. An ID is a system-generated globally unique identifier (GUID) for an instance.
- PrivateIps:** the list of private IP addresses of instances in a VPC. This parameter takes effect only when the **NetworkType** parameter is set to VPC. For example, a Json-formatted Array: `["172.16.XX.XX", "172.16.XX.XX", … "172.16.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- InnerIps:** the list of private IP addresses of instances in a classic network. This parameter takes effect only when the **NetworkType** parameter is set to Classic. For example, a Json-formatted Array: `["10.1.XX.XX", "10.1.XX.XX", … "10.1.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- PublicIps:** the list of public IP addresses of instances in a classic network. This parameter takes effect only when the **NetworkType** parameter is set to Classic. For example, a Json-formatted Array: `["42.1.XX.XX", "42.1.XX.XX", … "42.1.XX.XX"]` the maximum number of IP addresses that can be specified. Separate multiple IP addresses with commas (,).
- HostNames:** the list of hostnames of all instances.
- OrderId:** the list of order IDs of all instances.
- ZoneIds:** the IDs of the zones where created instances reside.
- RelatedOrderIds:** the list of related order IDs of created ECS instances.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId": "m-25l0r****",
        "InstanceType": "ecs.t1.small",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "MaxAmount": 1,
        "MinAmount": 1,
        "Tags": [
          {
            "Key": "tiantt",
            "Value": "ros"
          },
          {
            "Key": "tiantt1",
            "Value": "ros1"
          }
        ]
      }
    },
    "Outputs": {
      "InstanceIds": {
        "Value": {"get_attr": ["WebServer", "InstanceIds"]}
      },
      "PublicIps": {
        "Value": {"get_attr": ["WebServer", "PublicIps"]}
      }
    }
  }
}
```

5.5.1.12. ALIYUN::ECS::InstanceGroupClone

ALIYUN::ECS::InstanceGroupClone is used to clone an ECS instance group.

Statement

```
{
  "Type": "ALIYUN::ECS::InstanceGroupClone",
  "Properties": {
    "BackendServerWeight": Integer,
    "DiskMappings": List,
    "LaunchTemplateName": String,
    "SpotPriceLimit": String,
    "ResourceGroupId": String,
    "KeyPairName": String,
    "SystemDiskDiskName": String,
    "PeriodUnit": String,
    "Description": String,
    "Tags": List,
    "ImageId": String,
    "SpotStrategy": String,
    "SourceInstanceId": String,
    "EniMappings": List,
    "Password": String,
    "MaxAmount": Integer,
    "AutoReleaseTime": String,
    "SystemDiskCategory": String,
    "LoadBalancerIdToAttach": String,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "ZoneId": String,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "DeletionProtection": Boolean,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "RamRoleName": String,
    "HpcClusterId": String,
    "SystemDiskDescription": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the enterprise resource group to which a created instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
HpcClusterId	String	Yes	True	The ID of the E-HPC cluster to which a created instance belongs.	None
SourceInstanceId	String	No	No	The ID of the ECS instance to be cloned.	The clone operation clones the specified instance, including its instance type, image, bandwidth limit, and network type. If the source ECS instance belongs to multiple security groups, the cloned instance is added only to the first of these security groups.
MaxAmount	Integer	Retained	Yes	The maximum number of ECS instances to be created.	Valid values: 1 to 100. The MaxAmount parameter must be set to a value greater than or equal to the value of the MinAmount parameter.
MinAmount	String	No	Yes	The minimum number of ECS instances to be created.	Valid values: 1 to 100. The MinAmount parameter must be set to a value less than or equal to the value of the MaxAmount parameter.
BackendServerWeight	String	Optional	Released	The weight assigned to the ECS instance in the Server Load Balancer instance.	Valid values: 0 to 100. Default value: 100.
LoadBalancerIdToAttach	String	Yes	Released	The ID of the SLB instance to which the ECS instance is to be attached.	None
Description	String	Yes	Released	The description of created instances.	The description can be up to 256 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
ImageId	String	Yes	True	The ID of the image used to start created instances. You can use a public image, a custom image, or an Alibaba Cloud Marketplace image.	<p>You can specify a partial public image ID instead of providing the complete ID. When editing a template used to deploy an ECS instance, you can specify the image type and version or only the image type. ROS automatically selects an appropriate public image ID. You can use the wildcard character (*) to represent part of an image ID.</p> <p>You can use one of the following methods to specify the public image ID for the ECS instance:</p> <ul style="list-style-type: none"> If you set the parameter to ubuntu, ubuntu_16_0402_64_20G_alibase_20170818.vhd. If this parameter is set to ubuntu_14, ubuntu_14_0405_64_20G_alibase_20170824.vhd is returned. Specify: ubuntu1432, which will eventually match: ubuntu_14_0405_32_40G_alibase_20170711.vhd Specify: ubuntu_16_0402_32, which will eventually match: ubuntu_16_0402_32_40G_alibase_20170711.vhd
SecurityGroupId	String	Yes	Released	The ID of the security group to which created instances belong.	None
InstanceName	String	Yes	Released	The name of a created instance.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Password	String	Yes	Released	The password used to log on to the ECS instance.	The password must be 8 to 30 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special character. Special characters include () ` ~ ! @ # \$ % ^ & * - + = { } [] : ; ' < > , . ? / If the Password parameter is specified, you must use HTTPS to call the operation to ensure that the Password remains confidential.
DiskMappings	List	Erased	Released	The data disks to be attached to the instance.	A maximum of 16 disks can be attached to each instance.
Tags	List	Erased	Released	The custom tags of the instance.	You can specify a maximum of 20 tags. The format is as follows: <pre>[{"Key": "tagKey", "Value": "tagValue"}, {"Key": "tagKey2", "Value": "tagValue2"}]</pre>

Parameter	Type	Required	Editable	Description	Constraint
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None
KeyPairName	String	Yes	Released	The name of the key pair that is used to connect to the ECS instance. For Windows-based ECS instances, this parameter is ignored. Default value: empty. For Linux-based instances, the Password parameter still takes effect if this parameter is specified. However, logon by Password is disabled, and the KeyPairName value is used.	None
RamRoleName	String	Yes	Released	The RAM role name of the instance.	You can use RAM API ListRoles you can call this operation to query the RAM role name of an instance.
SpotPriceLimit	String	Yes	Released	The maximum hourly price of the instance.	This parameter supports up to three decimal places. Parameter SpotStrategy this parameter takes effect only when the value is SpotWithPriceLimit.
SystemDiskDiskName	String	Yes	True	The name of the system disk of created instances.	The name must be 2 to 128 characters in length Must start with an uppercase or lowercase letter, and cannot start with <code>http://</code> or <code>https://</code> and can contain digits, colons (:), underscores (_), and hyphens (-).
PeriodUnit	String	Yes	True	The billing cycle for created ECS instances.	Valid values: <ul style="list-style-type: none"> Week. <ul style="list-style-type: none"> 1, 2, 3, and 4} when the value of the PeriodUnit parameter is Week. AutoRenewPeriod values are 1, 2, "3". Month <ul style="list-style-type: none"> 1, 2, 3, 4, 5, 6, and 7 when the PeriodUnit parameter is set to Month "8", "9", "12", "24", "36", "48", "60"}, AutoRenewPeriod can be {"1", "2", "3", "6", "12"}. Default value: Month.
EniMappings	List	No.	True	The elastic network interfaces (ENIs) to be attached to a created instance.	Only a single ENI can be attached to each instance.

Parameter	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	Yes	Released	<p>The time scheduled for a created ECS instance to be automatically released. Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.</p> <ul style="list-style-type: none"> If the value of seconds (ss) is a value other than 00, the start time is automatically rounded down to the nearest minute based on the value of mm. The minimum release time must be at least 30 minutes later than the current time. The maximum release time must be at most three years from the current time. <p>If you do not specify the AutoReleaseTime it indicates that the auto release feature is disabled and the ECS instance will not be automatically released.</p>	None
SystemDiskCategory	String	Yes	True	The type of the system disk.	<p>Valid values:</p> <ul style="list-style-type: none"> cloud: basic disk cloud_efficiency: indicates an ultra disk. cloud_ssd: indicates a standard SSD. ephemeral_ssd: local SSD. cloud_essd: indicates an enhanced SSD (ESSD). ESSDs are still in public preview and only available in some regions. <p>For phased-out instance types that are not I/O optimized, the default value is cloud. For other instances, the default value is cloud_efficiency.</p>
LaunchTemplateName	String	Yes	True	The name of the launch template.	None

Parameter	Type	Required	Editable	Description	Constraint
LaunchTemplateVersion	String	Yes	True	The version of the launch template. If you do not specify this parameter, the default version is used.	None
InternetMaxBandwidthIn	String	Optional	Released	The maximum inbound bandwidth from the Internet. Unit: Mbit/s.	Valid values: 1 to 200. Default value: 200.
LaunchTemplateId	String	Yes	True	The ID of the launch template.	None
SystemDiskDescription	String	Yes	True	The description of the system disk.	The description must be 2 to 256 characters in length and cannot start with http:// or https://. This parameter is empty by default.
DeletionProtection	Boolean	Erased	Released	The release protection attribute of the instance. Specifies whether the ECS console or API DeleteInstance) to release the instance.	Valid values: <ul style="list-style-type: none"> • true • false
DeploymentSetId	String	Yes	True	Deployment Set ID.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String
  }
]
```

DiskMappings properties

Parameter	Type	Required	Editable	Description	Constraint
-----------	------	----------	----------	-------------	------------

Parameter	Type	Required	Editable	Description	Constraint
Encrypted	String	Yes	Released	Specifies whether to encrypt the data disk.	Valid values: <ul style="list-style-type: none"> • true • false Default value: false
KMSKeyId	String	Yes	Released	The KMS key ID for data disk N.	None
Size	String	No	No	The size of data disk N. Unit: GB.	None
Category	String	Yes	Released	The type of the data disk.	Valid values: <ul style="list-style-type: none"> • cloud • cloud_efficiency • cloud_ssd • ephemeral_ssdDefault
DiskName	String	Yes	Released	The name of data disk N.	The name can be up to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-).
Description	String	Yes	Released	The description of data disk N.	Valid values: 2 to 256. Default value: Null.
Device	String	Yes	Released	The device name of the data disk attached to an ECS instance.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	Yes	Released	Create a data disk by using a snapshot.	None

EniMappings syntax

```
"EniMappings": [
  {
    "SecurityGroupId": String,
    "VSwitchId": String,
    "Description": String,
    "NetworkInterfaceName": String,
    "PrimaryIpAddress": String
  }
]
```

EniMappings properties

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	No	Yes	The ID of the security group to which the ENI belongs.	None
VSwitchId	String	No	No	The ID of the VSwitch to which the ENI belongs.	None
Description	String	Yes	True	The description of the ENI.	It can contain 2 to 256 English letters or Chinese characters. It cannot start with <code>http://</code> and <code>https://</code> the beginning.
NetworkInterfaceName	String	Yes	True	The name of the ENI.	None
PrimaryIpAddress	String	Yes	Released	The primary IP address of the ENI.	None

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- **InstanceIds**: the IDs of instances in the ECS instance group. An ID is a globally unique identifier (GUID) generated by the system for an instance.
- **PrivateIps**: the private IP addresses of VPC-type instances. This parameter is valid only when the **NetworkType** parameter is set to VPC. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["172.16.XX.XX", "172.16.XX.XX", ... "172.16.XX.XX"].
- **InnerIps**: the private IP addresses of instances in the classic network. This parameter is valid only when the

NetworkType parameter is set to Classic. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["10.1.XX.XX", "10.1.XX.XX", ... "10.1.XX.XX"].

- PublicIps: the public IP addresses of instances in the classic network. This parameter is applicable only when the NetworkType parameter is set is Classic. The parameter value is a JSON-formatted array, containing up to 100 IP addresses separated by commas (,). Example: ["42.1.XX.XX", "42.1.XX.XX", ... "42.1.XX.XX"].
- HostNames: the hostnames of all instances. The parameter value is a JSON-formatted array. Example: ["host1", "host2", ... "host3"].
- OrderId: the order IDs of all instances.
- ZoneIds: the IDs of the zones where created instances reside.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroupClone",
      "Properties": {
        "SourceInstanceId": "i-25zsk****",
        "ImageId": "m-25l0r****",
        "SecurityGroupId": "sg-25zwc****",
        "ZoneId": "cn-beijing-b",
        "MaxAmount": 1,
        "MinAmount": 1
      }
    }
  },
  "Outputs": {
    "InstanceIds": {
      "Value": {"get_attr": ["WebServer", "InstanceIds"]}
    },
    "PublicIps": {
      "Value": {"get_attr": ["WebServer", "PublicIps"]}
    }
  }
}
```

5.5.1.13. ALIYUN::ECS::Invocation

ALIYUN::ECS::Invocation is used to invoke a Cloud Assistant command for one or more ECS instances.

Statement

```
{
  "Type": "ALIYUN::ECS::Invocation",
  "Properties": {
    "Timed": Boolean,
    "Frequency": String,
    "CommandId": String,
    "InstanceIds": List
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Timed	Boolean	Erased	Released	Specifies whether to invoke the command on a periodic basis. Default value: false.	None
Frequency	String	Yes	Released	The frequency at which the command is invoked.	None
CommandId	String	No	No	The ID of the script.	None
InstanceIds	List	Yes	No	The IDs of the instances for which you want to invoke the command. A maximum of 20 instance IDs can be specified.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

The execution ID of the Invokeld: command.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Timed": {
      "Type": "Boolean",
      "Description": "Whether it is timed execution. Default is False.",
      "AllowedValues": [
        "True",
        "False"
      ]
    }
  }
}
```

```

    true ,
    "False",
    "false"
  ]
},
"Frequency": {
  "Type": "String",
  "Description": "The frequency of timing execution (the shortest frequency is performed every 1 minute). It is mandatory when Timing is True.The value rule follows the rules of the cron expression. "
},
"CommandId": {
  "Type": "String",
  "Description": "The id of command."
},
"InstanceIds": {
  "Type": "CommaDelimitedList",
  "Description": "The instance id list. Select up to 20 instances at a time.Instances selected network type must be VPC network, status must be running"
}
},
"Resources": {
  "Invocation": {
    "Type": "ALIYUN::ECS::Invocation",
    "Properties": {
      "Timed": {
        "Ref": "Timed"
      },
      "Frequency": {
        "Ref": "Frequency"
      },
      "CommandId": {
        "Ref": "CommandId"
      },
      "InstanceIds": {
        "Fn::Split": [
          ",",
          {
            "Ref": "InstanceIds"
          },
          {
            "Ref": "InstanceIds"
          }
        ]
      }
    }
  }
}
}
},
}
}
},
}

```

```

"Outputs": {
  "InvokeId": {
    "Description": "The id of command execution.",
    "Value": {
      "Fn::GetAtt": [
        "Invocation",
        "InvokeId"
      ]
    }
  }
}
}
}
}

```

5.5.1.14. ALIYUN::ECS::JoinSecurityGroup

ALIYUN::ECS::JoinSecurityGroup is used to add one or more ECS instances to a specified security group.

Syntax

```

{
  "Type": "ALIYUN::ECS::JoinSecurityGroup",
  "Properties": {
    "InstanceId": String,
    "InstanceIdList": List,
    "SecurityGroupId": String,
    "NetworkInterfaceList": List
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	No	The ID of the security group.	None
InstanceId	String	No	No	The ID of the ECS instance to be added to the security group.	None
InstanceIdList	List	No	Yes	The IDs of the ECS instances to be added to the security group.	None
NetworkInterfaceList	List	No	Yes	The IDs of the elastic network interfaces (ENIs).	None

Response parameters

Fn::GetAtt

None

Examples

JSON format

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::JoinSecurityGroup",
      "Properties": {
        "SecurityGroupId": "sg-m5eagh7rzys2z8sa****",
        "InstanceIdList": [
          "i-m5e505h9bgsio0wy****",
          "i-m5e505hio0wyjc6r****"
        ]
      }
    }
  }
}
```

YAML format

```
ROSTemplateFormatVersion: '2015-09-01'
Resources:
  SG:
    Type: ALIYUN::ECS::JoinSecurityGroup
    Properties:
      SecurityGroupId: sg-m5eagh7rzys2z8sa****
      InstanceIdList:
        - i-m5e505h9bgsio0wy****
        - i-m5e505hio0wyjc6r****
```

5.5.1.15. ALIYUN::ECS::LaunchTemplate

ALIYUN::ECS::LaunchTemplate is used to create an ECS instance launch template.

Syntax

```

{
  "Type": "ALIYUN::ECS::LaunchTemplate",
  "Properties": {
    "LaunchTemplateName": String,
    "VersionDescription": String,
    "ImageId": String,
    "InstanceType": String,
    "SecurityGroupId": String,
    "NetworkType": String,
    "VSwitchId": String,
    "InstanceName": String,
    "Description": String,
    "InternetMaxBandwidthIn": Integer,
    "InternetMaxBandwidthOut": Integer,
    "HostName": String,
    "ZoneId": String,
    "SystemDiskCategory": String,
    "SystemDiskSize": Number,
    "SystemDiskDiskName": String,
    "SystemDiskDescription": String,
    "IoOptimized": String,
    "InternetChargeType": String,
    "UserData": String,
    "KeyPairName": String,
    "RamRoleName": String,
    "AutoReleaseTime": String,
    "SpotStrategy": String,
    "SpotPriceLimit": String,
    "SecurityEnhancementStrategy": String,
    "DiskMappings": List,
    "NetworkInterfaces": List,
    "Tags": List,
    "TemplateTags": List
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

Property	Type	Required	Editable	Description	Constraint
LaunchTemplateName	String	Yes	No	The name of the instance launch template.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>. It can contain letters, digits, colons (:), underscores (_), and hyphens (-).
VersionDescription	String	No	No	The description of the version of the instance launch template.	<ul style="list-style-type: none"> The description must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.
ImageId	String	No	No	The ID of the image.	None
InstanceType	String	No	No	The type of the instance.	None
SecurityGroupId	String	No	No	The ID of the security group.	None
NetworkType	String	No	No	The network type of the instance.	Valid values: <ul style="list-style-type: none"> classic vpc
VSwitchId	String	No	No	The ID of the VSwitch. You must specify this parameter when you create an instance in a VPC.	None
InstanceName	String	No	No	The name of the instance.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.
Description	String	No	No	The description of the instance.	<ul style="list-style-type: none"> The description must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.

Property	Type	Required	Editable	Description	Constraint
InternetMaxBandwidthIn	Integer	No	No	Maximum inbound bandwidth from the Internet.	Valid values: 1 to 200. Unit: Mbit/s.
InternetMaxBandwidthOut	Integer	No	No	Maximum outbound bandwidth to the Internet.	Valid values: 0 to 100. Unit: Mbit/s.
HostName	String	No	No	The hostname of the instance.	<p>The name cannot start or end with a period (.) or a hyphen (-). It cannot contain consecutive periods (.) or hyphens (-).</p> <ul style="list-style-type: none"> For Windows instances: <ul style="list-style-type: none"> The name must be 2 to 15 characters in length and can contain letters, digits, and hyphens (-). It cannot only contain digits. For other instances such as Linux instances: <ul style="list-style-type: none"> The name must be 2 to 64 characters in length and can contain letters, digits, and hyphens (-).
ZoneId	String	No	No	The ID of the zone where the instance resides.	None
SystemDiskCategory	String	No	No	The category of the system disk.	Valid values: <ul style="list-style-type: none"> cloud: the basic disk cloud_efficiency: the ultra disk cloud_ssd: the standard SSD ephemeral_ssd: the local SSD
SystemDiskSize	Number	No	No	The size of the system disk.	Valid values: 20 to 500. Unit: GiB.

Property	Type	Required	Editable	Description	Constraint
SystemDiskDiskName	String	No	No	The name of the system disk.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.
SystemDiskDescription	String	No	No	The description of the system disk.	The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
IoOptimized	String	No	No	Specifies whether the instance is I/O optimized.	Valid values: <ul style="list-style-type: none"> none optimized
InternetChargeType	String	No	No	The billing method for network usage.	Valid values: <ul style="list-style-type: none"> PayByBandwidth PayByTraffic
UserData	String	No	No	The user data of the instance.	The data must be encoded in Base64. The maximum size of the raw data is 16 KB.
KeyPairName	String	No	No	The AccessKey pair name.	<ul style="list-style-type: none"> For Windows instances, this parameter is ignored and is empty by default. The Password parameter takes effect even if the KeyPairName parameter is specified. For Linux instances, the username and password authentication method is disabled by default.
RamRoleName	String	No	No	The RAM role name of the instance.	None

Property	Type	Required	Editable	Description	Constraint
AutoReleaseTime	String	No	No	The time scheduled for the instance to be automatically released.	Specify the time in the ISO 8601 standard in the yyyy-MM-ddTHH:mm:ssZ format. The time must be in UTC.
SpotStrategy	String	No	No	The preemption policy for pay-as-you-go instances.	This parameter takes effect only when the InstanceChargeType parameter is set to PostPaid. Valid values: <ul style="list-style-type: none"> NoSpot: The instance is created as a regular pay-as-you-go instance. SpotWithPriceLimit: The instance to be created is a preemptible instance with a user-defined maximum hourly price. SpotAsPriceGo: The instance to be created is a preemptible instance whose price is based on the market price at the time of purchase.
SpotPriceLimit	String	No	No	The maximum hourly price of the instance.	A maximum of three decimal places can be specified.
SecurityEnhancementStrategy	String	No	No	Specifies whether to enable security hardening.	Valid values: <ul style="list-style-type: none"> Active: enables security hardening. Deactive: disables security hardening.
DiskMappings	List	No	No	The list of data disks.	A maximum of 16 data disks can be specified.
NetworkInterfaces	List	No	No	The list of elastic network interfaces (ENIs).	A maximum of eight ENIs can be specified.
Tags	List	No	No	The tags of the instance, security group, disks, and ENIs.	A maximum of 20 tags can be specified.
TemplateTags	List	No	No	The tags of the launch template.	A maximum of 20 tags can be specified.

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "DiskName": String,
    "Description": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "DeleteWithInstance": String
  }
]
```

DiskMappings properties

Property	Type	Required	Editable	Description	Constraint
Category	String	No	No	The category of the data disk.	Valid values: <ul style="list-style-type: none"> cloud: the basic disk cloud_efficiency: the ultra disk cloud_ssd: the standard SSD ephemeral_ssd: the local SSD
DiskName	String	No	No	The name of the data disk.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length. It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>. It can contain letters, digits, colons (:), underscores (_), and hyphens (-).
Description	String	No	No	The description of the data disk.	The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
SnapshotId	String	No	No	The ID of the snapshot used to create the data disk.	None

Property	Type	Required	Editable	Description	Constraint
Size	String	No	No	The size of the system disk.	<ul style="list-style-type: none"> Valid values when the Category parameter is set to cloud: 5 to 2000. Valid values when the Category parameter is set to cloud_efficiency: 20 to 32768. Valid values when the Category parameter is set to cloud_ssd: 20 to 32768. Valid values when the Category parameter is set to ephemeral_ssd: 5 to 800. Unit: GiB.
Encrypted	Boolean	No	No	Specifies whether to encrypt the data disk.	None
DeleteWithInstance	Boolean	No	No	Specifies whether to release the data disk when the instance is released.	None

NetworkInterfaces syntax

```

"NetworkInterfaces": [
  {
    "PrimaryIpAddress": String,
    "VSwitchId": String,
    "SecurityGroupId": String,
    "NetworkInterfaceName": String,
    "Description": String
  }
]
    
```

NetworkInterfaces properties

Property	Type	Required	Editable	Description	Constraint
PrimaryIpAddress	String	No	No	The primary private IP address of the ENI.	None
VSwitchId	String	No	No	The ID of the VSwitch to which the ENI belongs.	None
SecurityGroupId	String	No	No	The ID of the security group to which the ENI belongs.	None

Property	Type	Required	Editable	Description	Constraint
NetworkInterfaceName	String	No	No	The name of the ENI.	None
Description	String	No	No	The description of the ENI.	The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Property	Type	Required	Editable	Description	Constraint
key	String	Yes	No	None	None
value	String	No	No	None	None

TemplateTags syntax

```
"TemplateTags": [
  {
    "Value": String,
    "Key": String
  }
]
```

TemplateTags properties

Property	Type	Required	Editable	Description	Constraint
key	String	Yes	No	None	None
value	String	No	No	None	None

Response parameters

Fn::GetAtt

- **LaunchTemplateId**: the ID of the instance launch template.
- **LaunchTemplateName**: the name of the instance launch template.
- **DefaultVersionNumber**: the default version number of the instance launch template.
- **LatestVersionNumber**: the latest version number of the instance launch template.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Template1": {
      "Type": "ALIYUN::ECS::LaunchTemplate",
      "Properties": {
        "LaunchTemplateName": "MyTemplate",
        "VersionDescription": "Launch template with all properties set",
        "ImageId": "m-2ze9uqi7wo61hwep****",
        "InstanceType": "ecs.n4.small",
        "SecurityGroupId": "sg-2ze8yxgempcdsq3i****",
        "NetworkType": "vpc",
        "VSwitchId": "vsw-2zei67xd9nhcqzxec****",
        "InstanceName": "InstanceName",
        "Description": "Description of template",
        "InternetMaxBandwidthIn": 100,
        "InternetMaxBandwidthOut": 200,
        "HostName": "ttinfo",
        "ZoneId": "cn-beijing-a",
        "SystemDiskCategory": "cloud_ssd",
        "SystemDiskSize": "40",
        "SystemDiskDiskName": "TheSystemDiskName",
        "SystemDiskDescription": "The system disk description",
        "IoOptimized": "optimized",
        "InternetChargeType": "PayByBandwidth",
        "UserData": "dGhpcyBpcyBhIHVzZXIgzGF0YSBleG1h****",
        "KeyPairName": "ThisIsKeyPair",
        "RamRoleName": "ThisIsRamRole",
        "AutoReleaseTime": "2050-10-01T00:00:00Z",
        "SpotStrategy": "SpotWithPriceLimit",
        "SpotPriceLimit": "100.001",
        "SecurityEnhancementStrategy": "Active",
        "DiskMappings": [
          {
            "Category": "cloud_ssd",
            "Size": 40,
            "SnapshotId": "s-2ze1fr2bipove27b****",
            "Encrypted": true,
            "DiskName": "dataDisk1",
            "Description": "I am data disk 1",
            "DeleteWithInstance": true
          },
          {
            "Category": "cloud_efficiency",
            "Size": 20,
```

```

    "SnapshotId": "s-2ze4k0w8b33mlsqu****",
    "Encrypted": false,
    "DiskName": "dataDisk2",
    "Description": "I am data disk 2",
    "DeleteWithInstance": true
  }
],
"NetworkInterfaces": [
  {
    "PrimaryIpAddress": "10.10.1.1",
    "VSwitchId": "vsw-2zetgeiqlemyok9z5****",
    "SecurityGroupId": "sg-2ze8yxgempcdsq3i****",
    "NetworkInterfaceName": "my-eni1",
    "Description": "My eni 1"
  },
],
"Tags": [
  {
    "Key": "key1",
    "Value": "value1"
  },
  {
    "Key": "key2",
    "Value": "value2"
  }
],
"TemplateTags": [
  {
    "Key": "templateKey1",
    "Value": "templateValue1"
  },
  {
    "Key": "templateKey2",
    "Value": "templateValue2"
  }
]
}
},
"Outputs": {
  "LaunchTemplateId": {
    "Value": {"Fn::GetAtt": ["Template1", "LaunchTemplateId"]}
  },
  "LaunchTemplateName": {
    "Value": {"Fn::GetAtt": ["Template1", "LaunchTemplateName"]}
  },
  "DefaultVersionNumber": {

```

```

DefaultVersionNumber: {
  "Value": {"Fn::GetAtt": ["Template1", "DefaultVersionNumber"]}
},
"LatestVersionNumber": {
  "Value": {"Fn::GetAtt": ["Template1", "LatestVersionNumber"]}
}
}
}
}

```

5.5.1.16. ALIYUN::ECS::NatGateway

ALIYUN::ECS::NatGateway is used to create a NAT Gateway for a VPC.

Statement

```

{
  "Type": "ALIYUN::ECS::NatGateway",
  "Properties": {
    "DeletionProtection": Boolean,
    "VpcId": String,
    "Description": String,
    "NatGatewayName": String,
    "VSwitchId": String,
    "DeletionForce": Boolean,
    "Spec": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	Yse	No	The ID of the VPC that you want to create NAT Gateway.	None
VSwitchId	String	Yse	No	The ID of the vSwitch in the specified VPC.	None
Description	String	Erased	Released	The description of the NAT Gateway.	The description must be 2 to 256 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
NatGatewayName	String	Erased	Released	The name of the NAT Gateway.	The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), periods (.), and hyphens (-). It must start with a letter.
Spec	String	Erased	Released	The type of the NAT Gateway.	Valid values: Small, Middle, and Large.
DeletionProtection	Boolean	Erased	Released	Indicates whether deletion protection is enabled. Default value: false.	None
DeletionForce	Boolean	Erased	Released	Specifies whether to forcibly delete SNAT and DNAT entries in the Gateway and unbind EIP from the NAT gateway. Default value: false.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- ForwardTableId: the ID of the port forwarding table.
- The ID of the SNatTableId: SNat source network address translation table.
- NatGatewayId: the unique ID of the Nat gateway.

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "NatGateway": {
      "Type": "ALIYUN::ECS::NatGateway",
      "Properties": {
        "NatGatewayName": "nat_gateway_1",
        "Description": "my nat gateway",
        "VpcId": "vpc-25o8s****",
        "VSwitchId": "vsw-25rc1****",
        "Spec": "Small"
      }
    }
  },
  "Outputs": {
    "NatGatewayId": {
      "Value": {"Fn::GetAttr": ["NatGateway", "NatGatewayId"]}
    },
    "ForwardTableId": {
      "Value": {"Fn::GetAttr": ["NatGateway", "ForwardTableId"]}
    },
    "SNatTableId": {
      "Value": {"Fn::GetAttr": ["NatGateway", "SNatTableId"]}
    }
  }
}

```

5.5.1.17. ALIYUN::ECS::NetworkInterface

ALIYUN::ECS::NetworkInterface is used to create an elastic network interface (ENI).

Statement

```

{
  "Type": "ALIYUN::ECS::NetworkInterface",
  "Properties": {
    "Description": String,
    "SecurityGroupId": String,
    "PrimaryIpAddress": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "NetworkInterfaceName": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
SecurityGroupId	String	No	Yes	The ID of the security group to which the instance belongs. The security group and the instance must be in the same VPC.	None
VSwitchId	String	No	No	The ID of the VSwitch in the VPC.	None
Description	String	Yes	True	The description of the ENI. It can contain 2 to 256 English letters or Chinese character. It cannot start with <code>http://</code> and <code>https://</code> the beginning. This parameter is empty by default.	None
NetworkInterfaceName	String	Yes	True	The name of the ENI. The name must be 2 to 128 characters in length. Must start with an uppercase or lowercase letter, and cannot start with <code>http://</code> and <code>https://</code> the beginning. It can contain letters, digits, colons (:), underscores (_), and hyphens (-). Default value: null.	None

Parameter	Type	Required	Editable	Description	Constraint
PrimaryIpAddress	String	Yes	Released	The primary private IP address of the ENI. The specified IP address must be available within the CIDR block of the VSwitch. If this parameter is not specified, an available IP address in the VSwitch CIDR block is assigned at random.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- NetworkInterfaceId: the ID of the ENI.
- The MAC address of the MacAddress: Elastic Network Interface.
- The private IP address of the PrivateIpAddress: Elastic Network Interface.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Description": {
      "Type": "String",
      "Description": "Description of your ENI. It is a string of [2, 256] English or Chinese characters."
    },
    "SecurityGroupId": {
      "Type": "String",
      "Description": "The ID of the security group that the ENI joins. The security group and the ENI must be in a same VPC."
    },
    "VSwitchId": {
      "Type": "String",
      "Description": "VSwitch ID of the specified VPC. Specifies the switch ID for the VPC."
    },
    "NetworkInterfaceName": {
      "Type": "String",
      "Description": "Name of your ENI. It is a string of [2, 128] Chinese or English characters. It must begin with a letter and can contain numbers, underscores (_), colons (:), or hyphens (-)."
    },
    "PrimaryIpAddress": {
      "Type": "String",
      "Description": "The primary private IP address of the ENI. The specified IP address must have the same Host ID as t
```

```

    "Description": "The primary private IP address of the ENI. The specified IP address must have the same Host ID as the VSwitch. If no IP addresses are specified, a random network ID is assigned for the ENI."
  },
  "Resources": {
    "EniInstance": {
      "Type": "ALIYUN::ECS::NetworkInterface",
      "Properties": {
        "Description": {
          "Ref": "Description"
        },
        "SecurityGroupId": {
          "Ref": "SecurityGroupId"
        },
        "VSwitchId": {
          "Ref": "VSwitchId"
        },
        "NetworkInterfaceName": {
          "Ref": "NetworkInterfaceName"
        },
        "PrimaryIpAddress": {
          "Ref": "PrimaryIpAddress"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfaceId": {
      "Description": "ID of your Network Interface.",
      "Value": {
        "Fn::GetAtt": [
          "EniInstance",
          "NetworkInterfaceId"
        ]
      }
    }
  }
}

```

5.5.1.18. ALIYUN::ECS::NetworkInterfaceAttachment

ALIYUN::ECS::NetworkInterfaceAttachment is used to attach an elastic network interface (ENI) to an instance in a VPC.

Statement

```
{
  "Type": "ALIYUN::ECS::NetworkInterfaceAttachment",
  "Properties": {
    "InstanceId": String,
    "NetworkInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	No	The ID of the RDS instance.	None
NetworkInterfaceId	String	No	No	The IDs of the elastic network interfaces (ENIs).	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

NetworkInterfaceId: the ID of the ENI.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "InstanceId": {
      "Type": "String",
      "Description": "ECS instance id"
    },
    "NetworkInterfaceId": {
      "Type": "String",
      "Description": "Network interface id"
    }
  },
  "Resources": {
    "EniAttachment": {
      "Type": "ALIYUN::ECS::NetworkInterfaceAttachment",
      "Properties": {
        "InstanceId": {
          "Ref": "InstanceId"
        },
        "NetworkInterfaceId": {
          "Ref": "NetworkInterfaceId"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfaceId": {
      "Description": "ID of your Network Interface.",
      "Value": {
        "Fn::GetAtt": [
          "EniAttachment",
          "NetworkInterfaceId"
        ]
      }
    }
  }
}
```

5.5.1.19. ALIYUN::ECS::NetworkInterfacePermission

ALIYUN::ECS::NetworkInterfacePermission is used to grant an account the permission to attach an elastic network interface (ENI) to an instance.

Syntax

```
{
  "Type": "ALIYUN::ECS::NetworkInterfacePermission",
  "Properties": {
    "NetworkInterfaceId": String,
    "AccountId": String,
    "Permission": String
  }
}
```

Properties

Name	Type	Required	Editable	Description	Validity
NetworkInterfaceId	String	Yes	No	The ID of the ENI.	None
AccountId	String	Yes	No	The ID of the account.	None
Permission	String	Yes	No	The permission granted to the account.	None

Response parameters

Fn::GetAtt

- NetworkInterfacePermissionId: the ID of the ENI permission.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "AccountId": {
      "Type": "String",
      "Description": "the account id"
    },
    "Permission": {
      "Type": "String",
      "Description": "the permission",
      "Default": "InstanceAttach"
    },
    "NetworkInterfaceId": {
      "Type": "String",
      "Description": "Network interface id"
    }
  },
  "Resources": {
    "EniPermission": {
      "Type": "ALIYUN::ECS::NetworkInterfacePermission",
      "Properties": {
        "AccountId": {
          "Ref": "AccountId"
        },
        "Permission": {
          "Ref": "Permission"
        },
        "NetworkInterfaceId": {
          "Ref": "NetworkInterfaceId"
        }
      }
    }
  },
  "Outputs": {
    "NetworkInterfacePermissionId": {
      "Description": "the network interface permission id",
      "Value": {
        "Fn::GetAtt": [
          "EniPermission",
          "NetworkInterfacePermissionId"
        ]
      }
    }
  }
}
```

5.5.1.20. ALIYUN::ECS::Route

ALIYUN::ECS::Route is used to create a custom route.

Syntax

```
{
  "Type": "ALIYUN::ECS::Route",
  "Properties": {
    "DestinationCidrBlock": String,
    "RouteTableId": String,
    "NextHopId": String,
    "NextHopType": String,
    "RouteId": String,
    "NextHopList": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
DestinationCidrBlock	String	Yes	No	The destination Classless Inter-Domain Routing (CIDR) block of the route entry.	None
RouteTableId	String	Yes	No	The ID of the route table.	None
NextHopId	String	No	No	The ID of the next-hop instance of the route entry.	The route is a non-ECMP route.
RouteId	String	Yes	No	The ID of the route.	None
NextHopType	String	No	No	The type of the next hop.	Default value: Instance. Valid values: <ul style="list-style-type: none"> • Instance • Tunnel • HaVip • RouterInterface

Property	Type	Required	Editable	Description	Constraint
NextHopList	List	No	No	The list of next hops of the route entry.	<p>You must specify the NextHopType and NextHopId parameters to specify the next hops.</p> <ul style="list-style-type: none"> If you specify the NextHopList parameter, the route is an ECMP route. The list contains two to four next hops of the ECMP route entry. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Note The NextHopList parameter can be specified only when the route entry belongs to a VRouter. In addition, the next hops must be the router interfaces pointing to the connected VBRs.</p> </div> <ul style="list-style-type: none"> If you do not specify the NextHopList parameter, the route is a non-ECMP route.

NextHopList syntax

```
"NextHopList": [
  {
    "NextHopId": String,
    "NextHopType": String
  }
]
```

NextHopList properties

Property	Type	Required	Editable	Description	Constraint
NextHopId	String	Yes	No	The ID of the next-hop instance of the route entry.	None
NextHopType	String	No	No	The type of the next hop.	<p>Default value: RouterInterface. Valid values:</p> <ul style="list-style-type: none"> Instance Tunnel HaVip RouterInterface

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ECSRoute": {
      "Type": "ALIYUN::ECS::Route",
      "Properties": {
        "RouteId": "vrt-25mz0****",
        "RouteTableId": "vtb-25oud****",
        "DestinationCidrBlock": "172.16.XX.XX/24",
        "NextHopId": "i-25xzy****"
      }
    }
  }
}
```

5.5.1.21. ALIYUN::ECS::SNatEntry

ALIYUN::ECS::SNatEntry is used to configure a NAT Gateway table in a source network address translation.

Statement

```
{
  "Type": "ALIYUN::ECS::SNatEntry",
  "Properties": {
    "SNatTableId": String,
    "SNatIp": String,
    "SourceVSwitchId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
SNatTableId	String	Retained	Yes	The ID source network address translation table.	None
SNatIp	String	Retained	Yes	The public IP address used to source network address translation.	The public IP address must be NAT Gateway in the bandwidth plan. It cannot exist in both the forwarding table and the SNAT table.

Parameter	Type	Required	Editable	Description	Constraint
SourceVSwitchId	String	Retained	Yes	The ID of the VSwitch that accesses the Internet through the SNAT function of NAT Gateway.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

SNatEntryId: the table entry ID in the source network address translation table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SNatEntry": {
      "Type": "ALIYUN::ECS::SNatEntry",
      "Properties": {
        "SNatTableId": "stb-3er41****",
        "SourceVSwitchId": "vsw-25rc1****",
        "SNatIp": "101.201.XX.XX"
      }
    }
  },
  "Outputs": {
    "SNatEntryId": {
      "Value": {"Fn::GetAttr": ["SNatEntry", "SNatEntryId"]}
    }
  }
}
```

5.5.1.22. ALIYUN::ECS::SecurityGroup

ALIYUN::ECS::SecurityGroup is used to create a security group.

Statement

```

{
  "Type": "ALIYUN::ECS::SecurityGroup",
  "Properties": {
    "VpcId": String,
    "Description": String,
    "SecurityGroupName": String,
    "Tags": List,
    "SecurityGroupEgress": List,
    "SecurityGroupIngress": List,
    "ResourceGroupId": String,
    "SecurityGroupType": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
VpcId	String	Yes	Released	The ID of the VPC.	None
Description	String	Yes	Released	The description of the new security group.	The description must be 2 to 256 characters in length.
Tags	List	Erased	Released	The tags of the security group.	A maximum of 20 tags can be specified.

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupName	String	Yes	Released	The name of the new security group.	Default value: empty. <ul style="list-style-type: none"> The name must be 2 to 128 characters in length Must start with an uppercase or lowercase letter, and cannot start with <code>http://</code> and <code>https://</code> the beginning. It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.
SecurityGroupEgress	List	Erased	Released	The outbound access rules of the security group.	None
SecurityGroupIngress	List	Erased	Released	The inbound access rules of the security group.	None
SecurityGroupType	String	Yes	Released	The type of the new security group.	Valid values: <ul style="list-style-type: none"> normal (basic security group) enterprise (Advanced Security Group)

Tags syntax

```
"Tags": [
  {
    "Value": String,
    "Key": String
  }
]
```

Tags properties

Parameter	Type	Required	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	None

SecurityGroupEgress syntax

```
"SecurityGroupEgress": [
  {
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Priority": Integer,
    "DestGroupId": String,
    "DestCidrIp": String,
    "Policy": String,
    "IpProtocol": String,
    "DestGroupOwnerAccount": String,
    "DestGroupOwnerId": String,
    "Ipv6DestCidrIp": String
  }
]
```

SecurityGroupEgress properties

Parameter	Type	Required	Editable	Description	Constraint
Description	String	Yes	Released	The description of the security group rule.	The description must be 1 to 512 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
DestGroupOwnerId	String	Yes	Released	The ID of the Alibaba Cloud account that owns the destination security group. This parameter is used to grant the current security group access to security groups in another Alibaba Cloud account.	If neither the DestGroupOwnerId parameter nor the DestGroupOwnerAccount parameter is specified, the current security group is granted access to other security groups in the same Alibaba Cloud account. If the DestCidrIp parameter is specified, the DestGroupOwnerId parameter is ignored.
IpProtocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: <ul style="list-style-type: none"> • TCP • udp • icmp • GRE • All A value of all specifies that all the four protocols are supported.

Parameter	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	Released	The range of port numbers corresponding to the Internet protocol.	<p>The range of destination ports corresponding to the transport layer protocol. Valid values:</p> <ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to all, the port number range is -1/-1.
SecurityGroupId	String	Yes	Released	The ID of the security group for which to create the outbound access rules.	None

Parameter	Type	Required	Editable	Description	Constraint
NicType	String	Yes	Released	The network type of the instance. Valid values:	Valid values: <ul style="list-style-type: none"> internet intranet Default value: internet.
Priority	String	Optional	Released	The priority of the authorization policy.	Valid values: 1 to 100. Default value: 1
DestGroupId	String	Yes	Released	The ID of the destination security group within the same region.	You must specify either the DestGroupId parameter or the DestCidrIp parameter. If both parameters are specified, the system authorizes the destination CIDR block specified by the DestCidrIp parameter. If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, you must set the NicType parameter to intranet.
DestCidrIp	String	Yes	Released	The source IPv4 CIDR block.	The value must be in CIDR format. The default value is 0.0.0.0/0, indicating that access from any IP addresses is allowed. Examples of other supported formats include 10.159.XX.XX/12. Only IPv4 addresses are supported.

Parameter	Type	Required	Editable	Description	Constraint
Policy	String	Yes	Released	The authorization policy.	Valid values: <ul style="list-style-type: none"> accept: grants access drop: denies access Default value: accept.
DestGroupOwnerAccount	String	Yes	Released	The Alibaba Cloud account of the destination security group when you grant security group permissions across accounts.	None
Ipv6DestCidrIp	String	Yes	Released	The destination IPv6 CIDR block.	IPv6 addresses in CIDR format are supported. You can only specify the IP addresses for ECS instances in VPCs.

SecurityGroupIngress syntax

```
"SecurityGroupIngress": [
  {
    "SourceGroupOwnerId": String,
    "SourceGroupOwnerAccount": String,
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Ipv6SourceCidrIp": String,
    "Priority": Integer,
    "SourceGroupId": String,
    "Policy": String,
    "IpProtocol": String,
    "SourcePortRange": String,
    "SourceCidrIp": String
  }
]
```

SecurityGroupIngress properties

Parameter	Type	Required	Editable	Description	Constraint
SourceGroupOwnerid	String	Yes	Released	The ID of the Alibaba Cloud account that owns the source security group.	None
Description	String	Yes	Released	The description of the security group rule.	The description must be 1 to 512 characters in length.
IpProtocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: tcp, udp, icmp, gre, and all. A value of all specifies that all the four protocols are supported.

Parameter	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	Released	The range of port numbers corresponding to the Internet protocol.	<p>The range of destination ports corresponding to the transport layer protocol. Valid values:</p> <ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to all, the port number range is -1/-1, indicating that all ports are available.

Parameter	Type	Required	Editable	Description	Constraint
SourceGroupId	String	Yes	Released	The ID of the source security group within the same region.	You must specify either the SourceGroupId parameter or the SourceCidrIp parameter. If both parameters are specified, the system authorizes the source CIDR block specified by the SourceCidrIp parameter. If the SourceGroupId parameter is specified, but the SourceCidrIp parameter is not, you must set the NicType parameter to intranet.
SecurityGroupId	String	Yes	Released	The ID of the security group for which you want to create the inbound access rule.	None
NicType	String	Yes	Released	The network type of the instance. Valid values:	Valid values: <ul style="list-style-type: none"> internet intranet Default value: internet.
SourceGroupOwnerAccount	String	Yes	Released	The Alibaba Cloud account of the destination security group when you grant security group permissions across accounts.	None
Priority	String	Optional	Released	The priority of the authorization policy.	Valid values: 1 to 100. Default value: 1

Parameter	Type	Required	Editable	Description	Constraint
SourceCidrIp	String	Yes	Released	The source IPv4 CIDR block.	The value must be in CIDR format. The default value is 0.0.0.0/0, indicating that access from any IP addresses is allowed. Examples of other supported formats include 10.159.XX.XX/12. Only IPv4 CIDR blocks are supported.
Policy	String	Yes	Released	The authorization policy.	Valid values: <ul style="list-style-type: none"> • accept: accepts the request. • drop: access is denied. Default value: accept.

Parameter	Type	Required	Editable	Description	Constraint
SourcePortRange	String	Yes	Released	The range of source ports relevant to transport layer protocols.	Valid values: <ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all values are valid. When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. The IpProtocol value is all: -1/-1.
Ipv6SourceCidrIp	String	Yes	Released	The source IPv6 CIDR block. IPv6 addresses in CIDR format are supported.	You can only specify the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

SecurityGroupId: the ID of the new security group.

Sample request

```


```

```

{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroup",
      "Properties": {
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "SecurityGroupIngress": [
          {
            "SourceCidrIp": "0.0.0.0/0",
            "IpProtocol": "all",
            "NicType": "internet",
            "PortRange": "-1/-1",
            "Priority": 1
          },
          {
            "SourceCidrIp": "0.0.0.0/0",
            "IpProtocol": "all",
            "NicType": "intranet",
            "PortRange": "-1/-1",
            "Priority": 1
          }
        ],
        "SecurityGroupEgress": [
          {
            "IpProtocol": "all",
            "DestCidrIp": "0.0.0.0/0",
            "NicType": "internet",
            "PortRange": "-1/-1",
            "Priority": 1
          },
          {
            "IpProtocol": "all",
            "DestCidrIp": "0.0.0.0/0",
            "NicType": "intranet",
            "PortRange": "-1/-1",
            "Priority": 1
          }
        ],
        "VpcId": {
          "Ref": "Vpc"
        }
      }
    }
  },
}

```

```
"Outputs": {
  "SecurityGroupId": {
    "Value" : {"Fn::GetAtt": ["SG","SecurityGroupId"]}
  }
}
```

5.5.1.23. ALIYUN::ECS::SecurityGroupClone

ALIYUN::ECS::SecurityGroupClone is used to clone a security group.

Syntax

```
{
  "Type": "ALIYUN::ECS::SecurityGroupClone",
  "Properties": {
    "DestinationRegionId": String,
    "VpcId": String,
    "Description": String,
    "SecurityGroupName": String,
    "SourceSecurityGroupId": String,
    "ResourceGroupId": String,
    "NetworkType": String,
    "SecurityGroupType": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group to which the instance belongs.	None
SourceSecurityGroupId	String	Yes	No	The ID of the source security group.	Only applicable security group rules are copied to the new security group. The security group rules are selected based on the network type of the new security group.
NetworkType	String	No	No	The network type of the new security group.	Set the value to Classic.

Property	Type	Required	Editable	Description	Constraint
VpcId	String	No	No	The ID of the VPC to which the new security group belongs.	The NetworkType parameter is ignored if both the VpcId and NetworkType parameters are specified.
Description	String	No	No	The description of the new security group.	The description must be 2 to 256 characters in length. It cannot start with http:// or https://.
SecurityGroupName	String	No	No	The name of the new security group.	This parameter is empty by default. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter and cannot start with http:// or https://.
DestinationRegionId	String	No	No	The ID of the destination region where the new security group resides.	Default value: CURRENT.
SecurityGroupType	String	No	No	The type of the new security group.	Valid values: normal and enterprise. A value of normal specifies a basic security group. A value of enterprise specifies an advanced security group.

Response parameters

Fn::GetAtt

SecurityGroupId: the ID of the new security group.

Examples

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SecurityGroupClone": {
      "Type": "ALIYUN::ECS::SecurityGroupClone",
      "Properties": {
        "SourceSecurityGroupId": {
          "Ref": "SourceSecurityGroupId"
        },
        "VpcId": {
          "Ref": "VpcId"
        },
        "Description": {
          "Ref": "Description"
        },
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "DestinationRegionId": {
          "Ref": "DestinationRegionId"
        },
        "NetworkType": {
          "Ref": "NetworkType"
        }
      }
    }
  },
  "Parameters": {
    "SourceSecurityGroupId": {
      "Type": "String",
      "Description": "Source security group ID is used to copy properties to clone new security group. If the NetworkType and VpcId is not specified, the same security group will be cloned. If NetworkType or VpcId is specified, only proper security group rules will be cloned."
    },
    "VpcId": {
      "Type": "String",
      "Description": "Physical ID of the VPC."
    },
    "Description": {
      "Type": "String",
      "Description": "Description of the security group, [2, 256] characters. Do not fill or empty, the default is empty."
    },
    "SecurityGroupName": {
      "Type": "String",
      "Description": "Display name of the security group, [2, 128] English or Chinese characters, must start with a letter or Chinese in size, can contain numbers, '_' or '-', '-'
    }
  }
}

```

```

"DestinationRegionId": {
  "Default": "CURRENT",
  "Type": "String",
  "Description": "Clone security group to the specified region. Default to current region."
},
"NetworkType": {
  "Type": "String",
  "Description": "Clone new security group as classic network type. If the VpcId is specified, the value will be ignored."
},
"AllowedValues": [
  "Classic"
]
},
"Outputs": {
  "SecurityGroupId": {
    "Description": "Generated security group id of new security group.",
    "Value": {
      "Fn::GetAtt": [
        "SecurityGroupClone",
        "SecurityGroupId"
      ]
    }
  }
}
}
}

```

5.5.1.24. ALIYUN::ECS::SecurityGroupEgress

ALIYUN::ECS::SecurityGroupEgress is used to create an outbound access rule for a security group.

Statement

```

{
  "Type": "ALIYUN::ECS::SecurityGroupEgress",
  "Properties": {
    "SecurityGroupId": String,
    "IpProtocol": String,
    "PortRange": String,
    "DestGroupId": String,
    "DestGroupOwnerAccount": String,
    "DestCidrIp": String,
    "Policy": String,
    "Priority": String,
    "NicType": String,
    "Ipv6DestCidrIp": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
IpProtocol	String	No	No	The transport layer protocol.	Valid values: tcp, udp, icmp, gre, and all. A value of all specifies that all the four protocols are supported.

Parameter	Type	Required	Editable	Description	Constraint
PortRange	String	No	No	The range of destination ports relevant to transport layer protocols.	<p>Valid values:</p> <ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to gre, the port number range is -1/-1, indicating that all ports are available. When the IpProtocol parameter is set to all, the port number range is -1/-1, indicating that all ports are available. <p>For more information about the application scenarios of the ports, see Typical applications of commonly used ports.</p>

Parameter	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	Yes	Released	The ID of the source security group.	None
NicType	String	Yes	Released	The type of the ENI.	<p>Valid values:</p> <ul style="list-style-type: none"> Network interface controller intranet <p>Default value: internet.</p> <p>If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, this parameter must be set to intranet.</p>
Priority	String	Optional	Released	The priority of the security group rule.	Valid values: 1 to 100. Default value: 1
DestGroupId	String	Yes	Released	The ID of the source security group for which you want to set access permissions.	<p>You must specify at least one of the DestGroupId and DestCidrIp parameters. If the DestGroupId parameter is specified, but the DestCidrIp parameter is not, the NicType parameter must be set to intranet. If both the DestGroupId and DestCidrIp parameters are specified, the DestCidrIp parameter prevails by default.</p>
DestCidrIp	String	Yes	Released	The destination CIDR block.	Only IPv4 CIDR blocks are supported.

Parameter	Type	Required	Editable	Description	Constraint
Policy	String	Yes	Released	The authorization policy.	Valid values: <ul style="list-style-type: none"> accept: grants access drop: denies access Default value: accept.
DestGroupOwnerAccount	String	Yes	Released	The Alibaba Cloud account that manages the destination security group when you set a security group rule across accounts.	If neither the DestGroupOwnerAccount parameter nor the DestGroupOwnerId parameter is specified, the access permission is configured on another security group managed by your account. If the DestCidrIp parameter is specified, this parameter is ignored.
Description	String	Yes	True	The description of the security group rule.	The description must be 1 to 512 characters in length.
DestGroupOwnerId	String	Yes	Released	The ID of the Alibaba Cloud account that manages the destination security group when you set a security group rule across accounts.	If neither the DestGroupOwnerId parameter nor the DestGroupOwnerAccount parameter is specified, the access permission is configured on another security group managed by your account. If the DestCidrIp parameter is specified, the DestGroupOwnerId parameter is ignored.

Parameter	Type	Required	Editable	Description	Constraint
Ipv6DestCidrIp	String	Yes	Released	The destination IPv6 CIDR block.	IPv6 CIDR blocks are supported. You can only specify the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroupEgress",
      "Properties": {
        "SecurityGroupId": "sg-25bow****",
        "IpProtocol": "tcp",
        "PortRange": "65535/65535",
        "DestCidrIp": "0.0.0.0/0"
      }
    }
  }
}
```

5.5.1.25. ALIYUN::ECS::SecurityGroupIngress

ALIYUN::ECS::SecurityGroupIngress is used to create an inbound access rule for a security group.

Syntax

```

{
  "Type": "ALIYUN::ECS::SecurityGroupIngress",
  "Properties": {
    "SourceGroupOwnerId": String,
    "Description": String,
    "PortRange": String,
    "SecurityGroupId": String,
    "NicType": String,
    "Ipv6SourceCidrIp": String,
    "Priority": Integer,
    "SourceGroupId": String,
    "Policy": String,
    "IpProtocol": String,
    "SourcePortRange": String,
    "SourceCidrIp": String
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
IpProtocol	String	Yes	No	The Internet protocol.	Valid values: tcp, udp, icmp, gre, and all. A value of all specifies that all the four protocols are supported.

Property	Type	Required	Editable	Description	Constraint
PortRange	String	Yes	No	The range of destination ports relevant to transport layer protocols.	<ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1. When the IpProtocol parameter is set to gre, the port number range is -1/-1. When the IpProtocol parameter is set to all, the port number range is -1/-1.

Property	Type	Required	Editable	Description	Constraint
SourceGroupId	String	No	No	The ID of the source security group for which you want to set access permissions.	You must specify at least one of the SourceGroupId and SourceCidrIp parameters. If the SourceGroupId parameter is specified, but the SourceCidrIp parameter is not, the NicType parameter must be set to intranet. If both the SourceGroupId and SourceCidrIp parameters are specified, the SourceCidrIp value is used by default.
SecurityGroupId	String	No	No	The ID of the security group for which you want to create the inbound access rule.	None
NicType	String	No	No	The network type of the instance.	Valid values: <ul style="list-style-type: none"> internet intranet Default value: internet.
SourceGroupOwnerAccount	String	No	No	The Alibaba Cloud account that manages the source security group when you set a security group rule across accounts.	If neither the SourceGroupOwnerAccount parameter nor the SourceGroupOwnerIid parameter is specified, the access permission is configured for another security group managed by your account. If the SourceCidrIp parameter is specified, this parameter is ignored.

Property	Type	Required	Editable	Description	Constraint
Priority	Integer	No	No	The priority of the security group rule.	Valid values: 1 to 100. Default value: 1.
SourceCidrIp	String	No	No	The source IPv4 CIDR block.	Only IPv4 CIDR blocks are supported.
Policy	String	No	No	The access control policy.	Valid values: <ul style="list-style-type: none"> accept: grants access. drop: denies access. Default value: accept.
SourceGroupOwnerId	String	No	No	The ID of the Alibaba Cloud account that manages the source security group when you set a security group rule across accounts.	If neither the SourceGroupOwnerId parameter nor the SourceGroupOwnerIdAccount parameter is specified, the access permission is configured for another security group managed by your account. If the SourceCidrIp parameter is specified, this parameter is ignored.
Description	String	No	Yes	The description of the security group rule.	The description must be 1 to 512 characters in length.

Property	Type	Required	Editable	Description	Constraint
SourcePortRange	String	No	No	The range of source ports relevant to transport layer protocols.	<ul style="list-style-type: none"> When the IpProtocol parameter is set to tcp or udp, the port number range is 1 to 65535. Separate the starting port and the ending port with a forward slash (/). Correct example: 1/200. Incorrect example: 200/1. When the IpProtocol parameter is set to icmp, the port number range is -1/-1. When the IpProtocol parameter is set to gre, the port number range is -1/-1. When the IpProtocol parameter is set to all, the port number range is -1/-1.
Ipv6SourceCidrIp	String	No	No	The range of source IPv6 addresses.	CIDR blocks and IPv6 addresses are supported. You can only specify the IP addresses of ECS instances in VPCs.

Response parameters

Fn::GetAtt

None

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SG": {
      "Type": "ALIYUN::ECS::SecurityGroupIngress",
      "Properties": {
        "SecurityGroupId": "sg-25bow****",
        "IpProtocol": "tcp",
        "PortRange": "65535/65535",
        "SourceCidrIp": "0.0.0.0/0"
      }
    }
  }
}
```

5.5.1.26. ALIYUN::ECS::Snapshot

ALIYUN::ECS::Snapshot is used to create a disk Snapshot.

Statement

```
{
  "Type": "ALIYUN::ECS::Snapshot",
  "Properties": {
    "SnapshotName": String,
    "Timeout": Integer,
    "Description": String,
    "DiskId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DiskId	String	No	No	The ID of the disk for which you want to create the snapshot.	None

Parameter	Type	Required	Editable	Description	Constraint
SnapshotName	String	Yes	Released	The name of the snapshot.	It must be 2 to 128 characters in length. And can contain letters, digits, underscores (_), and hyphens (-). It cannot start with auto. Snapshot names starting with auto are reserved for automatic snapshots. It cannot start with <code>http://</code> or <code>https://</code> .
Timeout	String	Optional	Released	The timeout period that is specified for the snapshot creation request.	If this parameter is set, the timeout period to create a resource stack is prolonged. If the snapshot is not created within the specified time period, the entire resource stack fails to be created. You must set the timeout period according to the disk size and data amount. Valid values: 200 to 1440. Unit: minute. The default value is 200 minutes.
Description	String	Yes	Released	The description of the snapshot.	The length must be 2 to 256 characters in length. This parameter is empty by default. It cannot start with <code>http://</code> or <code>https://</code> .

Response parameters

Fn::GetAtt

SnapshotId: the ID of the snapshot.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Snapshot": {
      "Type": "ALIYUN::ECS::Snapshot",
      "Properties": {
        "DiskId": "d-2zedgvuvu8cylvr*****"
      }
    }
  },
  "Outputs": {
    "SnapshotId": {
      "Value": {
        "Fn::GetAtt": [
          "Snapshot",
          "SnapshotId"
        ]
      }
    }
  }
}
```

5.5.1.27. ALIYUN::ECS::SSHKeyPair

ALIYUN::ECS::SSHKeyPair is used to create or import an SSH key pair to an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::SSHKeyPair",
  "Properties": {
    "ResourceGroupId": String,
    "KeyPairName": String,
    "PublicKeyBody": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None

Parameter	Type	Required	Editable	Description	Constraint
KeyPairName	String	No	No	The globally unique name of the SSH key pair.	The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It cannot start with <code>http://</code> or <code>https://</code> .
PublicKeyBody	String	Yes	Released	Specifies the SSH public key to import.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- **KeyPairFingerPrint**: the fingerprint of the key pair. The message-digest algorithm 5 (MD5) is used based on the public key fingerprint format defined in RFC 4716.
- **PrivateKeyBody**: the private key of the key pair. An unencrypted RSA private key must be encoded using PEM and must be in the PKCS#8 format. The private key of a key pair can only be obtained at the time of its creation. If you import an existing public key, no private key information will be available.
- **KeyPairName**: the globally unique name of the SSH key pair.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SSHKeyPair": {
      "Type": "ALIYUN::ECS::SSHKeyPair",
      "Properties": {
        "KeyPairName": "ssh_key_pair_v1"
      }
    }
  },
  "Outputs": {
    "KeyPairName": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "KeyPairName"
        ]
      }
    },
    "PrivateKeyBody": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "PrivateKeyBody"
        ]
      }
    },
    "KeyPairFingerPrint": {
      "Value": {
        "Fn::GetAtt": [
          "SSHKeyPair",
          "KeyPairFingerPrint"
        ]
      }
    }
  }
}
```

5.5.1.28. ALIYUN::ECS::SSHKeyPairAttachment

ALIYUN::ECS::SSHKeyPairAttachment is used to bind an SSH key pair to an ECS instance.

Statement

```
{
  "Type": "ALIYUN::ECS::SSHKeyPairAttachment",
  "Properties": {
    "InstanceIds": List,
    "KeyPairName": String
  }
}
```

Properties

Parameter	Type	Required or Not	Editable	Description	Constraint
InstanceIds	List	Retained	Yes	The IDs of the ECS instances with which you want to associate the EIP.	Separate the IDs with a comma (.). Only Linux instances are supported.
KeyPairName	String	No	No	The name of the SSH key pair.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "SSHKeyPairAttachment": {
      "Type": "ALIYUN::ECS::SSHKeyPairAttachment",
      "Properties": {
        "KeyPairName": "ssh_key_pair_v1",
        "InstanceIds": [
          'l-2zeiofnh20j**** has been added *',
          'l-2zebt3kfvxm2**** has two records *'
        ]
      }
    }
  }
}
```

5.5.1.29. ALIYUN::ECS::VPC

ALIYUN::ECS::VPC is used to create a VPC.

Statement

```
{
  "Type": "ALIYUN::ECS::VPC",
  "Properties": {
    "Description": String,
    "Ipv6CidrBlock": String,
    "EnableIpv6": Boolean,
    "ResourceGroupId": String,
    "VpcName": String,
    "CidrBlock": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the instance belongs.	None
VpcName	String	Yes	True	The name of the VPC.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length Must start with english letters or starts with a Chinese character. It cannot start with <code>ht</code> <code>tp://</code> or <code>htt</code> <code>ps://</code> . It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.

Parameter	Type	Required	Editable	Description	Constraint
CidrBlock	String	Yes	True	The CIDR block of the VPC.	Valid values: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.
Description	String	Yes	True	The description of the VPC.	The description must be 2 to 256 characters in length. It cannot start with <code>http://</code> or <code>https://</code> .
Ipv6CidrBlock	String	Yes	Released	The IPv6 CIDR block of the VPC.	None
EnableIpv6	Boolean	No.	True	Specifies whether to enable an IPv6 CIDR block.	Valid values: <ul style="list-style-type: none"> • true • false Default value: false.

Response parameters

Fn::GetAtt

- VpcId: The VPC ID allocated by the system.
- VRouterId: the ID of the vRouter.
- RouteTableId: the ID of the routing table.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "EcsVpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": "172.16.0.0/12",
        "VpcName": "vpc-test-del"
      }
    }
  },
  "Outputs": {
    "VpcId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "VpcId"
        ]
      }
    },
    "VRouterId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "VRouterId"
        ]
      }
    },
    "RouteTableId": {
      "Value": {
        "Fn::GetAtt": [
          "EcsVpc",
          "RouteTableId"
        ]
      }
    }
  }
}
```

5.5.1.30. ALIYUN::ECS::VSwitch

ALIYUN::ECS::VSwitch is used to create a VSwitch.

Statement

```
{
  "Type": "ALIYUN::ECS::VSwitch",
  "Properties": {
    "VSwitchName": String,
    "VpcId": String,
    "Description": String,
    "Ipv6CidrBlock": Integer,
    "ZoneId": String,
    "CidrBlock": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VpcId	String	No	No	The ID of the VPC where a vSwitch is to be created	None
ZoneId	String	No	No	The ID of the zone where the instance resides.	None
VSwitchName	String	Yes	True	The name of the VSwitch.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length It must start with a letter. Cannot http:// or https:// the beginning. It must start with a letter and cannot start with http:// or https://.
CidrBlock	String	No	No	The CIDR block of the VSwitch.	The VSwitch CIDR block must be a subset of the CIDR block assigned to the VPC where the VSwitch resides and not be used by other VSwitches.

Parameter	Type	Required	Editable	Description	Constraint
Description	String	Yes	True	The description of the vSwitch.	The description must be 2 to 256 characters in length. It cannot start with <code>http://</code> or <code>https://</code> .
Ipv6CidrBlock	String	Optional	Released	The IPv6 CIDR block of the VSwitch. You can customize the last eight bits of the IPv6 CIDR block.	Valid values: 0 to 255. The value is a decimal integer. By default, the prefix of the IPv6 CIDR block of the VSwitch is set to /64.

Response parameters

Fn::GetAtt

- **VSwitchId**: indicates the vSwitch ID allocated by the system.
- **CidrBlock**: the IPv4 CIDR block of the vSwitch.
- **Ipv6CidrBlock**: the IPv6 CIDR block of the vSwitch.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String"
    },
    "VSwitch1CidrBlock": {
      "Type": "String",
      "Default": "172.16.100.0/24"
    },
    "VSwitch2CidrBlock": {
      "Type": "String",
      "Default": "172.16.80.0/24"
    }
  },
  "Resources": {
    "EcsVpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": "172.16.0.0/12",
        "VpcName": {"Ref": "VpcName"},
      },
    },
    "VSwitch1": {
```

```

VSwitch1: {
  "Type": "ALIYUN::ECS::VSwitch",
  "Properties": {
    "ZoneId": "cn-beijing-a",
    "CidrBlock": {"Ref": "VSwitch1CidrBlock"},
    "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] },
    "VSwitchName": "create_vpc_vswitch_sg1"
  }
},
VSwitch2: {
  "Type": "ALIYUN::ECS::VSwitch",
  "Properties": {
    "ZoneId": "cn-beijing-a",
    "CidrBlock": {"Ref": "VSwitch2CidrBlock"},
    "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] },
    "VSwitchName": "create_vpc_vswitch_sg2"
  }
},
SG_VSwitch1: {
  "Type": "ALIYUN::ECS::SecurityGroup",
  "Properties": {
    "SecurityGroupName": "app_mall",
    "Description": "this is created by heat",
    "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] }
  },
  "Outputs": {
    "SecurityGroupId": {
      "Value": {"get_attr": ["SG_VSwitch1","SecurityGroupId"]}
    }
  }
},
SG_VSwitch1_InRule: {
  "Type": "ALIYUN::ECS::SecurityGroupIngress",
  "Properties": {
    "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch1", "SecurityGroupId" ] },
    "IpProtocol": "tcp",
    "PortRange": "1/65535",
    "SourceCidrIp": {"Ref": "VSwitch2CidrBlock"}
  }
},
SG_VSwitch1_OutRule: {
  "Type": "ALIYUN::ECS::SecurityGroupEgress",
  "Properties": {
    "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch1", "SecurityGroupId" ] },
    "IpProtocol": "tcp",
    "PortRange": "1/65535",
    "DestCidrIp": {"Ref": "VSwitch2CidrBlock"}
  }
}

```

```

    }
  },
  "SG_VSwitch2": {
    "Type": "ALIYUN::ECS::SecurityGroup",
    "Properties": {
      "SecurityGroupName": "app_mall",
      "Description": "this is created by heat",
      "VpcId": { "Fn::GetAtt": [ "EcsVpc", "VpcId" ] }
    },
  },
  "SG_VSwitch2_InRule": {
    "Type": "ALIYUN::ECS::SecurityGroupIngress",
    "Properties": {
      "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch2", "SecurityGroupId" ] },
      "IpProtocol": "tcp",
      "PortRange": "1/65535",
      "SourceCidrIp": {"Ref": "VSwitch1CidrBlock"}
    }
  },
  "SG_VSwitch2_OutRule": {
    "Type": "ALIYUN::ECS::SecurityGroupEgress",
    "Properties": {
      "SecurityGroupId": { "Fn::GetAtt": [ "SG_VSwitch2", "SecurityGroupId" ] },
      "IpProtocol": "tcp",
      "PortRange": "1/65535",
      "DestCidrIp": {"Ref": "VSwitch1CidrBlock"}
    }
  }
}
}
}
}

```

5.5.2. ESS

5.5.2.1. ALIYUN::ESS::AlarmTask

ALIYUN::ESS::AlarmTask is used to create a metric-based alarm task.

Syntax

```

{
  "Type": "ALIYUN::ESS::AlarmTask",
  "Properties": {
    "Statistics": String,
    "Name": String,
    "EvaluationCount": Integer,
    "Period": Integer,
    "MetricType": String,
    "ComparisonOperator": String,
    "Dimensions": List,
    "ScalingGroupId": String,
    "AlarmAction": List,
    "Threshold": Number,
    "MetricName": String,
    "GroupId": Integer,
    "Description": String
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
Statistics	String	No	No	The method used to calculate monitoring data. The statistics must be appropriate for the metric chosen.	Valid values: Average, Minimum, and Maximum. Default value: Average.
Name	String	No	Yes	The name of the alarm rule.	None
EvaluationCount	Integer	No	No	The number of consecutive times that the threshold must be exceeded before an alarm is triggered.	Default value: 3. Minimum value: 1.
Period	Integer	No	No	The metric query period, which must be appropriate for the metric chosen. Unit: seconds.	Valid values: 60, 120, 300, and 900. Default value: 300.
MetricType	String	No	No	The metric type.	Valid values: system and custom.
ComparisonOperator	String	No	No	The alarm comparison operator used to define a condition in the alarm rule.	Valid values: <=, <, >, and >=.
Dimensions	List	No	No	The list of instances associated with the alarm rule.	You must include at least one instance in the list.

Property	Type	Required	Editable	Description	Constraint
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None
AlarmAction	List	Yes	Yes	The list of alarm actions.	You must include one to five alarm actions in the list.
Threshold	Number	Yes	No	The alarm threshold, which must be a numeric value.	None
MetricName	String	Yes	No	The metric name of a service. For more information, see the metrics defined for each service.	None
GroupId	Integer	No	No	The group ID.	None
Description	String	No	Yes	The description of the alarm task.	None

Dimensions syntax

```
"Dimensions": [
  {
    "DimensionKey": String,
    "DimensionValue": String
  }
]
```

Dimensions properties

Property	Type	Required	Editable	Description	Constraint
DimensionValue	String	Yes	No	None	None
DimensionKey	String	Yes	No	None	None

Response parameters

Fn::GetAtt

AlarmTaskId: the ID of the alarm task.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "ComparisonOperator": {
      "Type": "String",
      "Description": "Comparison Operator",
      "AllowedValues": [
        ">="
      ]
    }
  }
}
```

```
"<=",
">",
"<"
]
},
"Description": {
  "Type": "String",
  "Description": "Description"
},
"ScalingGroupId": {
  "Type": "String",
  "Description": "The ID of the scaling group."
},
"MetricType": {
  "Type": "String",
  "Description": "Metric Type",
  "AllowedValues": [
    "system",
    "custom"
  ]
},
"EvaluationCount": {
  "Type": "Number",
  "Description": "Evaluation Count",
  "MinValue": 1
},
"Period": {
  "Type": "Number",
  "Description": "Period",
  "AllowedValues": [
    60,
    120,
    300,
    900
  ]
},
"Dimensions": {
  "Type": "CommaDelimitedList",
  "Description": "Dimensions",
  "MinLength": 1
},
"Statistics": {
  "Type": "String",
  "Description": "Statistics",
  "AllowedValues": [
    "Average",
    "Minimum"
```

```
    "Minimum": {
      "Type": "Number",
      "Description": "Minimum"
    },
    "Maximum": {
      "Type": "Number",
      "Description": "Maximum"
    }
  ],
  "Name": {
    "Type": "String",
    "Description": "Name"
  },
  "GroupId": {
    "Type": "Number",
    "Description": "Group Id"
  },
  "MetricName": {
    "Type": "String",
    "Description": "Metric Name"
  },
  "AlarmAction": {
    "Type": "CommaDelimitedList",
    "Description": "Alarm Actions",
    "MinLength": 1,
    "MaxLength": 5
  },
  "Threshold": {
    "Type": "Number",
    "Description": "Threshold"
  }
},
"Resources": {
  "AlarmTask": {
    "Type": "ALIYUN::ESS::AlarmTask",
    "Properties": {
      "ComparisonOperator": {
        "Ref": "ComparisonOperator"
      },
      "Description": {
        "Ref": "Description"
      },
      "ScalingGroupId": {
        "Ref": "ScalingGroupId"
      },
      "MetricType": {
        "Ref": "MetricType"
      },
      "EvaluationCount": {
        "Ref": "EvaluationCount"
      },
      "Period": {
```

```

    "Ref": "Period"
  },
  "Dimensions": {
    "Fn::Split": [
      ",",
      {
        "Ref": "Dimensions"
      }
    ],
    {
      "Ref": "Dimensions"
    }
  ]
},
"Statistics": {
  "Ref": "Statistics"
},
"Name": {
  "Ref": "Name"
},
"GroupID": {
  "Ref": "GroupID"
},
"MetricName": {
  "Ref": "MetricName"
},
"AlarmAction": {
  "Fn::Split": [
    ",",
    {
      "Ref": "AlarmAction"
    }
  ],
  {
    "Ref": "AlarmAction"
  }
]
},
"Threshold": {
  "Ref": "Threshold"
}
}
},
"Outputs": {
  "AlarmTaskId": {
    "Description": "The alarm task ID",
    "Value": {
      "Fn::GetAtt": [

```

```

    "AlarmTask",
    "AlarmTaskId"
  ]
}
}
}
}
}

```

5.5.2.2. ALIYUN::ESS::AlarmTaskEnable

ALIYUN::ESS::AlarmTaskEnable is used to start an alarm task. You can call this operation to enable alarm tasks when the task is stopped.

Statement

```

{
  "Type": "ALIYUN::ESS::AlarmTaskEnable",
  "Properties": {
    "AlarmTaskId": String,
    "Enable": Boolean
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AlarmTaskId	String	No	No	The ID of the monitoring task.	None
Enable	String	Retained	Yes	Specifies whether to enable the alarm task.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Enable": {
      "Type": "Boolean",
      "Description": "Enable alarm task or not",
      "AllowedValues": [
        "True",
        "true",
        "False",
        "false"
      ]
    },
    "AlarmTaskId": {
      "Type": "String",
      "Description": "The id of alarm task."
    }
  },
  "Resources": {
    "AlarmTaskEnable": {
      "Type": "ALIYUN::ESS::AlarmTaskEnable",
      "Properties": {
        "Enable": {
          "Ref": "Enable"
        },
        "AlarmTaskId": {
          "Ref": "AlarmTaskId"
        }
      }
    }
  },
  "Outputs": {}
}
```

5.5.2.3. ALIYUN::ESS::LifecycleHook

ALIYUN::ESS::LifecycleHook is used to create a lifecycle hook for a scaling group.

Syntax

```
{
  "Type": "ALIYUN::ESS::LifecycleHook",
  "Properties": {
    "LifecycleHookName": String,
    "NotificationArn": String,
    "HeartbeatTimeout": Integer,
    "NotificationMetadata": String,
    "ScalingGroupId": String,
    "DefaultResult": String,
    "LifecycleTransition": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
LifecycleHookName	String	No	Yes	The name of the lifecycle hook. Each lifecycle hook name must be unique within a scaling group.	<p>The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.</p> <p>The default name is the ID of the lifecycle hook.</p>
NotificationArn	String	No	Yes	The Alibaba Cloud Resource Name (ARN) of the notification target that Auto Scaling uses to notify you when an instance is in the transition state for the lifecycle hook.	<p>This target can be either an MNS queue or an MNS topic. The format of the parameter value is <code>acs:ess:{region}:{account-id}:{resource-relative-id}</code>.</p> <ul style="list-style-type: none"> <code>region</code> : the region where the scaling group resides. <code>account-id</code> : the ID of the Apsara Stack tenant account. <p>Examples:</p> <ul style="list-style-type: none"> MNS queue: <code>acs:ess:{region}:{account-id}:queue/{queueName}</code> MNS topic: <code>acs:ess:{region}:{account-id}:topic/{topicName}</code>

Property	Type	Required	Editable	Description	Constraint
HeartbeatTimeout	Integer	No	Yes	The waiting period before the lifecycle hook times out. When the lifecycle hook times out, the scaling group performs the action specified by the DefaultResult parameter. Unit: seconds.	Valid values: 30 to 21600. Default value: 600.
NotificationMetadata	String	No	Yes	The fixed string to include when Auto Scaling sends a notification about the wait state of a scaling activity. Auto Scaling sends the specified NotificationMetadata parameter value along with the notification message so that you can easily categorize notifications. The NotificationMetadata parameter is valid only after you set the NotificationArn parameter.	The parameter value cannot exceed 128 characters in length.
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None
DefaultResult	String	No	Yes	The action that the scaling group takes when the lifecycle hook times out. If the scaling group has multiple lifecycle hooks and one of them is terminated when the DefaultResult parameter is set to ABANDON during a scale-in event, the remaining lifecycle hooks in the same scaling group will also be terminated. Otherwise, the scaling activity will proceed normally after the waiting period expires and continue with the action specified by the DefaultResult parameter.	Valid values: <ul style="list-style-type: none"> CONTINUE: The scaling group continues the scale-in or scale-out event. ABANDON: The scaling group releases the created ECS instances if the scaling activity type is scale-out or removes the ECS instances to be scaled in if the scaling activity type is scale-in. Default value: CONTINUE.

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

LifecycleTransition	String	Yes	Yes	The type of scaling activity to which the lifecycle hook applies.	Valid values: <ul style="list-style-type: none"> SCALE_OUT: scale-out events of the scaling group. SCALE_IN: scale-in events of the scaling group.
---------------------	--------	-----	-----	---	--

Response parameters

Fn::GetAtt

LifecycleHookId: the ID of the lifecycle hook.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "LifecycleHookName": {
      "Type": "String",
      "Description": "The name of the lifecycle hook. Each name must be unique within a scaling group. The name must be 2 to 40 characters in length and can contain letters, numbers, Chinese characters, and special characters including underscores (_), hyphens (-) and periods (.).\nDefault value: Lifecycle Hook ID",
      "AllowedPattern": "^[a-zA-Z0-9\\u4e00-\\u9fa5][-_a-zA-Z0-9\\u4e00-\\u9fa5]{1,63}$"
    },
    "NotificationArn": {
      "Type": "String",
      "Description": "The Alibaba Cloud Resource Name (ARN) of the notification target that Auto Scaling will use to notify you when an instance is in the transition state for the lifecycle hook. This target can be either an SNS queue or an M"
    }
  }
}
```

by you when an instance is in the transition state for the lifecycle hook. This target can be either an MNS queue or an MNS topic. The format of the parameter value is `acs:ess:{region}:{account-id}:{resource-relative-id}.\nregion: the region to which the scaling group locates\naccount-id: Alibaba Cloud ID\nFor example:\nMNS queue: acs:ess:{region}:{account-id}:queue/{queuename}\nMNS topic: acs:ess:{region}:{account-id}:topic/{topicname}`",

```
"AllowedPattern": "^acs:ess:([a-zA-Z0-9-]+):(\d+):(queue|topic)/([a-zA-Z0-9][a-zA-Z0-9-]{0,255})$",
```

```
"MaxLength": 300
```

```
},
```

```
"ScalingGroupId": {
```

```
"Type": "String",
```

```
"Description": "The ID of the scaling group."
```

```
},
```

```
"LifecycleTransition": {
```

```
"Type": "String",
```

"Description": "The scaling activities to which lifecycle hooks apply Value range:\n SCALE_OUT: scale-out event\n SCALE_IN: scale-in event",

```
"AllowedValues": [
```

```
"SCALE_OUT",
```

```
"SCALE_IN"
```

```
]
```

```
},
```

```
"HeartbeatTimeout": {
```

```
"Type": "Number",
```

"Description": "The time, in seconds, that can elapse before the lifecycle hook times out. If the lifecycle hook times out, the scaling group performs the default action (DefaultResult). The range is from 30 to 21,600 seconds. The default value is 600 seconds.\nYou can prevent the lifecycle hook from timing out by calling the RecordLifecycleActionHeartbeat operation. You can also terminate the lifecycle action by calling the CompleteLifecycleAction operation.",

```
"MinValue": 30,
```

```
"MaxValue": 21600
```

```
},
```

```
"NotificationMetadata": {
```

```
"Type": "String",
```

"Description": "The fixed string that you want to include when Auto Scaling sends a message about the wait state of the scaling activity to the notification target. The length of the parameter can be up to 128 characters. Auto Scaling will send the specified NotificationMetadata parameter along with the notification message so that you can easily categorize your notifications. The NotificationMetadata parameter will only take effect after you specify the NotificationArn parameter.",

```
"MaxLength": 128
```

```
},
```

```
"DefaultResult": {
```

```
"Type": "String",
```

"Description": "The action that the scaling group takes when the lifecycle hook times out. Value range:\n CONTINUE: the scaling group continues with the scale-in or scale-out process.\n ABANDON: the scaling group stops any remaining action of the scale-in or scale-out event.\nDefault value: CONTINUE\nIf the scaling group has multiple lifecycle hooks and one of them is terminated by the DefaultResult=ABANDON parameter during a scale-in event (SCALE_IN), the remaining lifecycle hooks under the same scaling group will also be terminated. Otherwise, the action following the wait state is the next action, as specified in the parameter DefaultResult, after the last lifecycle event under the same scaling group.",

```

    "AllowedValues": [
      "CONTINUE",
      "ABANDON"
    ]
  },
  "Resources": {
    "LifecycleHook": {
      "Type": "ALIYUN::ESS::LifecycleHook",
      "Properties": {
        "LifecycleHookName": {
          "Ref": "LifecycleHookName"
        },
        "NotificationArn": {
          "Ref": "NotificationArn"
        },
        "ScalingGroupId": {
          "Ref": "ScalingGroupId"
        },
        "LifecycleTransition": {
          "Ref": "LifecycleTransition"
        },
        "HeartbeatTimeout": {
          "Ref": "HeartbeatTimeout"
        },
        "NotificationMetadata": {
          "Ref": "NotificationMetadata"
        },
        "DefaultResult": {
          "Ref": "DefaultResult"
        }
      }
    }
  },
  "Outputs": {
    "LifecycleHookId": {
      "Description": "The lifecycle hook ID",
      "Value": {
        "Fn::GetAtt": [
          "LifecycleHook",
          "LifecycleHookId"
        ]
      }
    }
  }
}

```

5.5.2.4. ALIYUN::ESS::ScalingConfiguration

ALIYUN::ESS::ScalingConfiguration is used to create a scaling configuration.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingConfiguration",
  "Properties": {
    "PasswordInherit": Boolean,
    "DiskMappings": List,
    "RamRoleName": String,
    "IoOptimized": String,
    "InternetChargeType": String,
    "KeyPairName": String,
    "InstanceId": String,
    "InstanceTypes": List,
    "ImageId": String,
    "ResourceGroupId": String,
    "SpotStrategy": String,
    "InstanceType": String,
    "SystemDiskCategory": String,
    "SystemDiskSize": Integer,
    "InternetMaxBandwidthOut": Integer,
    "InstanceName": String,
    "InternetMaxBandwidthIn": Integer,
    "ScalingConfigurationName": String,
    "UserData": String,
    "DeploymentSetId": String,
    "SecurityGroupId": String,
    "SpotPriceLimit": Number,
    "HpcClusterId": String,
    "ScalingGroupId": String,
    "SpotPriceLimitForInstanceType": Map,
    "TagList": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	Yes	The ID of the resource group to which the instance belongs.	None
DeploymentSetId	String	No	No	The ID of the deployment set.	None

Property	Type	Required	Editable	Description	Constraint
HpcClusterId	String	No	No	The ID of the E-HPC cluster to which the instance belongs.	None
ScalingGroupId	String	Yes	No	The ID of the scaling group to which the scaling configuration belongs.	None
DiskMappings	List	No	No	The data disks to be attached to the instance.	A maximum of 16 data disks can be attached.
InternetChargeType	String	No	No	The billing method for Internet usage.	Valid values: PayByBandwidth and PayByTraffic. Default value: PayByTraffic
InternetMaxBandwidthIn	Integer	No	No	The maximum inbound bandwidth from the Internet. Unit: Mbit/s.	Valid values: 1 to 100. Default value: 100.
InternetMaxBandwidthOut	Integer	No	No	The maximum outbound bandwidth to the Internet. Unit: Mbit/s.	Valid values for the PayByBandwidth mode: 0 to 200. Default value: 0. Valid values for the PayByTraffic mode: 1 to 200. If you choose to use the PayByTraffic mode, you must specify this parameter.
InstanceId	String	No	No	The ID of the ECS instance whose properties are used to create the scaling configuration.	None
SystemDiskCategory	String	No	No	The type of the system disk.	Valid values: cloud, cloud_efficiency, cloud_ssd, and ephemeral_ssd.
ImageId	String	No	No	The ID of the image used to start the ECS instance. You can use a public image, custom image, or Alibaba Cloud Marketplace image.	N/A
InstanceType	String	No	No	The type of the ECS instance.	N/A

Property	Type	Required	Editable	Description	Constraint
SecurityGroupId	String	No	No	The ID of the security group to which the created instance belongs.	None
IoOptimized	String	No	No	Specifies whether the instance is I/O optimized.	Valid values: none (non-I/O optimized) and optimized (I/O optimized). Default value: none.
ScalingConfigurationName	String	No	No	The display name of the scaling configuration.	The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. This parameter must be unique in a scaling group. The default name is the ID of the scaling configuration.
KeyPairName	String	No	No	The name of the key pair that is used to connect to the ECS instance.	For Windows ECS instances, this parameter is ignored and is empty by default. The Password parameter takes effect even if the KeyPairName parameter is specified. However, Linux instances do not allow logon by password, and the KeyPairName value is used instead.
RamRoleName	String	No	No	The RAM role name of the instance.	You can call the ListRoles operation to query the role name.
SystemDiskSize	Integer	No	Yes	The size of the system disk. Unit: GB.	Valid values: 40 to 500. If a custom image is used to create a system disk, ensure that the size of the system disk is greater than that of the custom image.
UserData	String	No	No	The user data provided to create the ECS instance.	The user data can be up to 16 KB in size. You must convert the data into Base64-encoded strings. If the data contains special characters, add a backslash (\) immediately before each special character.

Property	Type	Required	Editable	Description	Constraint
InstanceTypes	List	No	No	The ECS instance types that can be used in a scaling group.	A maximum of 10 ECS instance types can be specified. If this parameter is specified, the InstanceType parameter is ignored.
PasswordInherit	Boolean	No	Yes	Specifies whether to use the preconfigured password of the specified image. To use this parameter, ensure that a password is configured for the specified image.	None
TagList	List	No	Yes	<p>The tags of the instance. Tags must be specified as key-value pairs. A maximum of 20 tags can be specified. The following rules apply to keys and values:</p> <ul style="list-style-type: none"> • A key can contain a maximum of 64 characters and cannot start with aliyun, http://, or https://. You cannot specify an empty string as a key. • A value can contain a maximum of 128 characters and cannot start with aliyun, http://, or https://. You can specify an empty string as a value. 	None
SpotStrategy	String	No	Yes	<p>The preemption policy for pay-as-you-go instances. Valid values:</p> <ul style="list-style-type: none"> • NoSpot: applies to regular pay-as-you-go instances. • SpotWithPriceLimit: applies to preemptible instances with a maximum hourly price. • SpotAsPriceGo: applies to pay-as-you-go instances priced at the market price at the time of purchase. <p>Default value: NoSpot.</p>	Valid values: NoSpot, SpotWithPriceLimit, and SpotAsPriceGo.

Property	Type	Required	Editable	Description	Constraint
InstanceName	String	No	Yes	The name of the instance created based on the current scaling configuration.	None
SpotPriceLimit	Number	No	Yes	The maximum hourly price for preemptible instance N. Valid values of N: 1 to 10. This parameter is valid only when the SpotStrategy parameter is set to SpotWithPriceLimit. A maximum of three decimal places can be specified. The default value of InstanceTypes can be overridden by the value of SpotPriceLimitForInstanceType.	None
SpotPriceLimitForInstanceType	Map	No	Yes	The instance type of preemptible instance N. Valid values of N: 1 to 10. This parameter is valid only when the SpotStrategy parameter is set to SpotWithPriceLimit. Example: { "key1": "value1", "key2": "value2", ... "key5": "value5" }. A key is an ECS instance type. A value can have a maximum of three decimal places.	None

DiskMappings syntax

```
"DiskMappings": [
  {
    "Category": String,
    "Device": String,
    "SnapshotId": String,
    "Size": String,
    "Encrypted": String,
    "KMSKeyId": String,
    "Description": String,
    "DiskName": String
  }
]
```

DiskMappings properties

Property	Type	Required	Editable	Description	Constraint
Size	String	Yes	No	The size of the data disk. Unit: GB.	None
Category	String	No	No	The type of the data disk.	Valid values: cloud, cloud_efficiency, cloud_ssd, and ephemeral_ssd.
DiskName	String	No	No	The name of the data disk.	The name must be 2 to 128 characters in length and can contain letters, digits, colons (:), underscores (_), and hyphens (-). It must start with a letter and cannot start with http:// or https://. This parameter is empty by default.
Description	String	No	No	The description of the data disk.	The description must be 2 to 256 characters in length and cannot start with http:// or https://.
Device	String	No	No	The device name of the data disk.	The system allocates a device name in alphabetical order from /dev/xvda to /dev/xvdz.
SnapshotId	String	No	No	The ID of the snapshot used to create the data disk.	None
Encrypted	String	No	No	Specifies whether to encrypt the data disk.	Default value: false.
KMSKeyId	String	No	No	The KMS key ID for the data disk.	None

TagList syntax

```
"TagList": [
  {
    "Value": String,
    "Key": String
  }
]
```

TagList properties

Property	Type	Required	Editable	Description	Constraint
Key	String	Yes	No	None	None

Property	Type	Required	Editable	Description	Constraint
Value	String	Yes	No	None	None

Response parameters

Fn::GetAtt

ScalingConfigurationId: the ID of the scaling configuration. This ID is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingConfiguration": {
      "Type": "ALIYUN::ESS::ScalingConfiguration",
      "Properties": {
        "ImageId": "ubuntu1404_64_20G_aliaegis_20150325.vhd",
        "InstanceType": "ecs.t1.small",
        "InstanceId": "i-25xhh****",
        "InternetChargeType": "PayByTraffic",
        "InternetMaxBandwidthIn": 1,
        "InternetMaxBandwidthOut": 20,
        "SystemDisk_Category": "cloud",
        "ScalingGroupId": "bwhtvpcBcKYac9fe3vd0****",
        "SecurityGroupId": "sg-25zwc****",
        "DiskMappings": [
          {
            "Size": 10
          },
          {
            "Category": "cloud",
            "Size": 10
          }
        ]
      }
    }
  },
  "Outputs": {
    "ScalingConfiguration": {
      "Value": {"get_attr": ["ScalingConfigurationId"]}
    }
  }
}
```

5.5.2.5. ALIYUN::ESS::ScalingGroup

ALIYUN::ESS::ScalingGroup is used to create a scaling group. A scaling group is a group of ECS instances that are dynamically scaled based on the configured scenario. A scaling group does not take effect immediately after being created. You must use ALIYUN::ESS::ScalingGroupEnable to enable the scaling group to trigger scaling rules and execute scaling activities.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingGroup",
  "Properties": {
    "MultiAZPolicy": String,
    "DesiredCapacity": Integer,
    "NotificationConfigurations": List,
    "ProtectedInstances": List,
    "LaunchTemplateId": String,
    "LaunchTemplateVersion": String,
    "ScalingGroupName": String,
    "VSwitchIds": List,
    "DefaultCooldown": Integer,
    "MinSize": Integer,
    "GroupDeletionProtection": Boolean,
    "MaxSize": Integer,
    "Instanceld": String,
    "VSwitchId": String,
    "LoadBalancerIds": List,
    "StandbyInstances": List,
    "RemovalPolicys": List,
    "HealthCheckType": String,
    "DBInstancelds": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

Property	Type	Required	Editable	Description	Constraint
MinSize	Integer	Yes	Yes	The minimum number of ECS instances in the scaling group.	Valid values: 0 to 1000. When the number of ECS instances in the scaling group is less than the value of MinSize, Auto Scaling automatically creates ECS instances until the number of instances is equal to the value of MinSize.
MaxSize	Integer	Yes	Yes	The maximum number of ECS instances in the scaling group.	Valid values: 0 to 1000. When the number of ECS instances in the scaling group is greater than the value of MaxSize, Auto Scaling removes the ECS instances from the scaling group until the number of instances is equal to the value of MaxSize.

Property	Type	Required	Editable	Description	Constraint
ScalingGroupName	String	No	Yes	The display name of the scaling group.	<ul style="list-style-type: none"> The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. The name must be unique to an Alibaba Cloud account in a region. The default value is the ID of the scaling group.
LaunchTemplateId	String	No	Yes	The ID of the instance launch template from which the scaling group obtains launch configurations.	None
LaunchTemplateVersion	String	No	Yes	The version of the instance launch template.	Valid values: <ul style="list-style-type: none"> The fixed template version number. Default: The default template version is always used. Latest: The latest template version is always used.

Property	Type	Required	Editable	Description	Constraint
RemovalPolicys	List	No	Yes	The policies to remove ECS instances from the scaling group.	Default value: OldestScalingConfiguration or OldestInstance. Valid values: <ul style="list-style-type: none"> • OldestInstance: removes the ECS instance that is added to the scaling group at the earliest point in time. • NewestInstance: removes the ECS instance that is added to the scaling group at the latest point in time. • OldestScalingConfiguration: removes the ECS instance that is created based on the earliest scaling configuration.
VSwitchId	String	No	No	The ID of the VSwitch.	None
LoadBalancerIds	List	No	Yes	The IDs of the SLB instances.	This value can be a JSON array that contains up to five SLB instance IDs. Separate multiple IDs with commas (,).

Property	Type	Required	Editable	Description	Constraint
DefaultCooldown	Integer	No	Yes	The cooldown period after a scaling activity (adding or removing ECS instances) is executed.	<ul style="list-style-type: none"> Valid values: 0 to 86400. Unit: seconds. Default value: 300. During the cooldown period, the scaling group does not execute any other scaling activities triggered by CloudMonitor event-triggered tasks.
DBInstanceIds	List	No	Yes	The IDs of ApsaraDB for RDS instances.	This value can be a JSON array that contains up to eight RDS instance IDs. Separate multiple IDs with commas (,).
VSwitchIds	List	No	No	The IDs of VSwitches.	You can specify a maximum of five VSwitch IDs. If this parameter is specified, the VSwitchId parameter is ignored. VSwitches are sorted in descending order of priority. When an ECS instance cannot be created in the zone where the VSwitch with the highest priority resides, the system automatically uses the VSwitch with the next highest priority to create the ECS instance.
					Valid values: <ul style="list-style-type: none"> PRIORITY: ECS instances are

Property	Type	Required	Editable	Description	scaled based on the constraint on the
MultiAZPolicy	String	No	No	The ECS instance scaling policy for the multi-zone scaling group.	<p>specified VSwitch. When an ECS instance cannot be created in the zone where the VSwitch with the highest priority resides, the system automatically uses the VSwitch with the next highest priority to create the ECS instance.</p> <ul style="list-style-type: none"> • BALANCE: ECS instances are distributed evenly in multiple zones specified in the scaling group. • COST_OPTIMIZED: ECS instances are created based on the unit price of vCPUs, from low to high. Preemptible instances are created first when preemptible instance types are specified for the scaling configuration. Pay-as-you-go instances are created automatically when all types of preemptible instances are unavailable due to issues such as insufficient ECS resources.

Property	Type	Required	Editable	Description	Constraint
NotificationConfigurations	List	No	Yes	The notification configurations for event and resource changes.	None
ProtectedInstances	List	No	Yes	The number of protected ECS instances in the scaling group.	Maximum value: 1000.
StandbyInstances	List	No	Yes	The number of ECS instances that are in the standby state in the scaling group.	Maximum value: 1000.
HealthCheckType	String	No	Yes	The health check type.	Valid values: <ul style="list-style-type: none"> ECS NONE
GroupDeletionProtection	Boolean	No	Yes	Specifies whether to enable deletion protection for the scaling group.	Valid values: <ul style="list-style-type: none"> true: enables deletion protection for the scaling group. In this case, you cannot delete the scaling group. false: disables deletion protection for the scaling group. This is the default value.
DesiredCapacity	Integer	No	Yes	The expected number of ECS instances in the scaling group. The scaling group automatically keeps the number of ECS instances at the expected value.	The number of ECS instances must be greater than the value of MinSize and be less than the value of MaxSize.

Property	Type	Required	Editable	Description	Constraint
InstanceId	String	No	No	The ID of the ECS instance from which the scaling group obtains configuration information of the specified instance.	None

Response parameters

`Fn::GetAtt`

`ScalingGroupId`: the ID of the scaling group. This ID is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingGroup": {
      "Type": "ALIYUN::ESS::ScalingGroup",
      "Properties": {
        "MaxSize": 1,
        "MinSize": 1,
        # "ScalingGroupName": "HeatCreatedR****",
        # "DefaultCooldown": 500,
        # "RemovalPolicy_1": "",
        # "RemovalPolicy_2": "",
      }
    }
  },
  "Outputs": {
    "ScalingGroup": {
      "Value": {"Fn::GetAtt": ["ScalingGroup", "ScalingGroupId"]}
    }
  }
}
```

5.5.2.6. ALIYUN::ESS::ScalingGroupEnable

`ALIYUN::ESS::ScalingGroupEnable` is used to enable a scaling group.

Syntax

```

{
  "Type": "ALIYUN::ESS::ScalingGroupEnable",
  "Properties": {
    "ScalingConfigurationId": String,
    "ScalingRuleArisExecuteVersion": Integer,
    "ScalingRuleAris": List,
    "ScalingGroupId": String,
    "RemoveInstanceIds": List,
    "InstanceIds": List
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
ScalingGroupId	String	Yes	No	The ID of the scaling group.	None
ScalingConfigurationId	String	No	No	The ID of the scaling configuration to be activated in the scaling group.	None
InstanceIds	List	No	Yes	The IDs of ECS instances to be added to the enabled scaling group.	A maximum of 20 instance IDs can be specified.
ScalingRuleArisExecuteVersion	Integer	No	Yes	The version of the identifier for the scaling rule to be executed. If you change this property, all scaling rules specified by ScalingRuleAris will be executed once.	Minimum value: 0.
ScalingRuleAris	List	No	Yes	The unique identifiers of scaling rules in the scaling group. Invalid unique identifiers are not displayed in the query results and no errors are reported.	A maximum of 10 scaling rule identifiers can be specified.

Property	Type	Required	Editable	Description	Constraint
RemoveInstanceIds	List	No	Yes	The IDs of ECS instances to be deleted.	A maximum of 1,000 instance IDs can be specified.

Response parameters

Fn::GetAtt

- **LifecycleState**: the status of the scaling group.
- **ScalingInstances**: the instances that are automatically created in the scaling group.
- **ScalingGroupId**: the ID of the scaling group.
- **ScalingRuleArisExecuteResultInstancesRemoved**: the instances that are removed from the scaling group by executing the scaling rules specified by **ScalingRuleAris**.
- **ScalingRuleArisExecuteResultNumberOfAddedInstances**: the number of instances that are added to the scaling group by executing the scaling rules specified by **ScalingRuleAris**.
- **ScalingInstanceDetails**: the instance scaling details.
- **ScalingRuleArisExecuteErrorInfo**: the error information about the execution of the scaling rules specified by **ScalingRuleAris**.
- **ScalingRuleArisExecuteResultInstancesAdded**: the instances that are added to the scaling group by executing the scaling rules specified by **ScalingRuleAris**.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingGroupEnable": {
      "Type": "ALIYUN::ESS::ScalingGroupEnable",
      "Properties": {
        "ScalingGroupId": "r0HUqbj411cc2eQw8bU****",
        "ScalingConfigurationId": "bjLLfdexm77Ldsyptme1****",
        "InstanceIds": ""
      }
    }
  },
  "Outputs": {
    "ScalingGroupEnable": {
      "Value": {"Fn::GetAtt": ["ScalingGroupEnable", "LifecycleState"]}
    }
  }
}
```

5.5.2.7. ALIYUN::ESS::ScalingRule

ALIYUN::ESS::ScalingRule is used to create a scaling rule.

Syntax

```
{
  "Type": "ALIYUN::ESS::ScalingRule",
  "Properties": {
    "AdjustmentValue": Integer,
    "Cooldown": Integer,
    "ScalingGroupId": String,
    "AdjustmentType": String,
    "ScalingRuleName": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
AdjustmentValue	Integer	No	Yes	<p>The number of ECS instances to add or release when scaling occurs.</p> <p>The number of ECS instances to be adjusted in a single scaling activity cannot exceed 500.</p>	<p>Valid values in different adjustment modes:</p> <ul style="list-style-type: none"> QuantityChangeInCapacity: -500 to 500. PercentChangeInCapacity: -100 to 10000. TotalCapacity: 0 to 1000.
Cooldown	Integer	No	Yes	The cooldown period of the scaling rule. Unit: seconds.	<p>Valid values: 0 to 86400.</p> <p>This parameter is empty by default.</p>
ScalingGroupId	String	Yes	No	The ID of the scaling group to which the scaling rule belongs.	None
AdjustmentType	String	Yes	Yes	The adjustment mode of the scaling rule.	<p>Valid values:</p> <ul style="list-style-type: none"> QuantityChangeInCapacity: adds or removes a specified number of ECS instances. PercentChangeInCapacity: adds or removes a specified proportion of ECS instances. TotalCapacity: adds or removes ECS instances to ensure that the current scaling group has a specified number of ECS instances.

Property	Type	Required	Editable	Description	Constraint
ScalingRuleName	String	No	Yes	The display name of the scaling rule.	<p>The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit. The name of a scaling rule must be unique within the scaling group that it belongs to.</p> <p>The default value is the ID of the scaling rule.</p>

Response parameters

Fn::GetAtt

- ScalingRuleAri: the unique identifier of the scaling rule.
- ScalingRuleId: the ID of the scaling rule. It is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScalingRule": {
      "Type": "ALIYUN::ESS::ScalingRule",
      "Properties": {
        "ScalingRuleName": {
          "Ref": "ScalingRuleName"
        },
        "Cooldown": {
          "Ref": "Cooldown"
        },
        "ScalingGroupId": {
          "Ref": "ScalingGroupId"
        },
        "AdjustmentType": {
          "Ref": "AdjustmentType"
        },
        "AdjustmentValue": {
          "Ref": "AdjustmentValue"
        }
      }
    }
  },
  "Parameters": {
    "ScalingRuleName": {
      "AllowedPattern": "^[a-zA-Z0-9\\u4e00-\\u9fa5][-_\\.a-zA-Z0-9\\u4e00-\\u9fa5]{1,63}$",

```

```

    "Type": "String",
    "Description": "Name shown for the scaling group, which is a string containing 2 to 40 English or Chinese characters . It must begin with a number, a letter (case-insensitive) or a Chinese character and can contain numbers, \"_\", \"-\" or r\\.\\. The account name in the same scaling group is unique in the same region. If this parameter value is not specified, the default value is ScalingRuleId."
  },
  "Cooldown": {
    "Type": "Number",
    "Description": "Cool-down time of a scaling rule. Value range: [0, 86,400], in seconds. The default value is empty.",
    "MaxValue": 86400,
    "MinValue": 0
  },
  "ScalingGroupId": {
    "Type": "String",
    "Description": "ID of the scaling group of a scaling rule."
  },
  "AdjustmentType": {
    "Type": "String",
    "Description": "Adjustment mode of a scaling rule. Optional values:\n- QuantityChangeInCapacity: It is used to increase or decrease a specified number of ECS instances.\n- PercentChangeInCapacity: It is used to increase or decrease a specified proportion of ECS instances.\n- TotalCapacity: It is used to adjust the quantity of ECS instances in the current scaling group to a specified value.",
    "AllowedValues": [
      "QuantityChangeInCapacity",
      "PercentChangeInCapacity",
      "TotalCapacity"
    ]
  },
  "AdjustmentValue": {
    "Type": "Number",
    "Description": "Adjusted value of a scaling rule. Value range:\n- QuantityChangeInCapacity: [-500, 500]\n- PercentChangeInCapacity: [-100, 10000]\n- TotalCapacity: [0, 1000]",
    "MaxValue": 10000,
    "MinValue": -500
  }
},
"Outputs": {
  "ScalingRuleAri": {
    "Description": "Unique identifier of a scaling rule.",
    "Value": {
      "Fn::GetAtt": [
        "ScalingRule",
        "ScalingRuleAri"
      ]
    }
  }
},
"ScalingRuleId": {

```

```

ScalingRuleId : {
  "Description": "ID of a scaling rule, generated by the system and globally unique.",
  "Value": {
    "Fn::GetAtt": [
      "ScalingRule",
      "ScalingRuleId"
    ]
  }
}
}
}
}
}

```

5.5.2.8. ALIYUN::ESS::ScheduledTask

ALIYUN::ESS::ScheduledTask is used to create a scheduled task based on input parameters.

Syntax

```

{
  "Type": "ALIYUN::ESS::ScheduledTask",
  "Properties": {
    "TaskEnabled": Boolean,
    "Description": String,
    "ScheduledTaskName": String,
    "LaunchExpirationTime": Integer,
    "LaunchTime": String,
    "RecurrenceEndTime": String,
    "RecurrenceType": String,
    "RecurrenceValue": String,
    "ScheduledAction": String
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
TaskEnabled	Boolean	No	Yes	Specifies whether to start the scheduled task. <ul style="list-style-type: none"> true: starts the scheduled task. false: stops the scheduled task. Default value: true.	None
Description	String	No	Yes	The description of the scheduled task.	The description must be 2 to 200 characters in length.

Property	Type	Required	Editable	Description	Constraint
ScheduledTaskName	String	No	Yes	The display name of the scheduled task.	<p>The name must be 2 to 40 characters in length and can contain letters, digits, underscores (_), hyphens (-), and periods (.). It must start with a letter or digit.</p> <p>This parameter must be unique in a region and under an Apsara Stack tenant account.</p> <p>The default value is the ID of the scheduled scaling task.</p>
LaunchExpirationTime	Integer	No	Yes	<p>The time period during which a failed scheduled task is retried.</p> <p>Unit: seconds. Default value: 600.</p>	Valid values: 0 to 21600.
LaunchTime	String	Yes	Yes	<p>The time at which the scheduled task is triggered.</p> <p>Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. The time must be in UTC.</p> <p>If the RecurrenceType parameter is specified, the task is executed each day at the time specified by LaunchTime.</p> <p>If the RecurrenceType parameter is not specified, the task is only executed once at the date and time specified by LaunchTime.</p> <p>You cannot enter a point in time later than 90 days from the date of scheduled task creation or modification.</p>	None

Property	Type	Required	Editable	Description	Constraint
RecurrenceEndTime	String	No	Yes	<p>The end time after which the scheduled task will not be repeated.</p> <p>Specify the time in the ISO 8601 standard in the YYYY-MM-DDThh:mmZ format. The time must be in UTC.</p> <p>You cannot enter a point in time later than 90 days from the date of scheduled task creation or modification.</p> <p>If you set RecurrenceEndTime, you must also set both RecurrenceType and RecurrenceValue.</p>	None
RecurrenceType	String	No	Yes	<p>The interval that the scheduled task is repeated at.</p>	<p>Valid values:</p> <ul style="list-style-type: none"> • Daily: The scheduled task is executed once every specified number of days. • Weekly: The scheduled task is executed on each specified day of a week. • Monthly: The scheduled task is executed on each specified day of a month. • Cron: The scheduled task is executed based on the specified Cron expression. <p>If you set RecurrenceType, you must also set both RecurrenceEndTime and RecurrenceValue.</p>

Property	Type	Required	Editable	Description	Constraint
RecurrenceValue	String	No	Yes	Specifies how often the scheduled task recurs.	<ul style="list-style-type: none"> • Daily: indicates the interval of days that the scheduled task is repeated on. You can enter a single value ranging from 1 to 31. • Weekly: indicates which days of the week that the scheduled task is repeated on. You can enter multiple values separated by commas (.). The values 0 to 6 correspond to the days of the week in sequence from Sunday to Saturday. • Monthly: indicates which days of the month that the scheduled task is repeated on. You can enter two values ranging from 1 to 31. The format is A-B. B must be greater than or equal to A. • Cron: indicates a user-defined Cron expression that the scheduled task is repeated on. A Cron expression is written in UTC time and consists of five fields: minute, hour, day of month (date), month, and day of week. The expression can contain wildcard characters including commas (,), question marks (?), hyphens (-), asterisks (*), number signs (#), forward slashes (/), and the L and W characters. <p>If you set RecurrenceValue, you must also set both RecurrenceEndTime and RecurrenceType.</p>
ScheduledAction	String	Yes	Yes	<p>The operations to be performed when the scheduled task is triggered.</p> <p>When you set this parameter, you must also enter the unique identifier of the scaling rule.</p>	<p>The parameter value can be up to 200 characters in length.</p>

Response parameters

Fn::GetAtt

ScheduledTaskId: the ID of the scheduled task. This ID is a globally unique identifier (GUID) generated by the system.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "ScheduledTask": {
      "Type": "ALIYUN::ESS::ScheduledTask",
      "Properties": {
        "TaskEnabled": "true",
        "Description": "scheduledtask",
        "ScheduledTaskName": "task1",
        "LaunchTime": "2014-08-17T16:52Z",
        "RecurrenceEndTime": "2014-08-17T16:55Z",
        "RecurrenceType": "Daily",
        "RecurrenceValue": "1",
        "ScheduledAction": "ari:acs:ess:cn-qingdao:1344371:scalingRule/cCBpdYdQuBe2cUxOdu6piOk"
      }
    }
  },
  "Outputs": {
    "ScheduledTaskId": {
      "Value": {
        "FN::GetAtt": [
          "ScheduledTask",
          "ScheduledTaskId"
        ]
      }
    }
  }
}
```

5.5.3. OSS

5.5.3.1. ALIYUN::OSS::Bucket

ALIYUN::OSS::Bucket is used to create an OSS bucket.

Syntax

```
{
  "Type": "ALIYUN::OSS::Bucket",
  "Properties": {
    "AccessControl": String,
    "RefererConfiguration": Map,
    "ServerSideEncryptionConfiguration": Map,
    "CORSConfiguration": Map,
    "Tags": Map,
    "LoggingConfiguration": Map,
    "LifecycleConfiguration": Map,
    "StorageClass": String,
    "DeletionForce": Boolean,
    "WebsiteConfiguration": Map,
    "Policy": Map,
    "BucketName": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
BucketName	String	Yes	No	The name of the bucket.	<ul style="list-style-type: none"> The name must be 3 to 63 characters in length and can contain lowercase letters, digits, and hyphens (-). It must start and end with a lowercase letter or digit.
AccessControl	String	No	No	The access control policy.	Valid values: private, public-read, and public-read-write.
CORSConfiguration	Map	No	No	The configuration of cross-origin resource sharing for objects in the bucket.	None
LifecycleConfiguration	Map	No	No	The lifecycle configuration for objects in the bucket.	None
LoggingConfiguration	Map	No	No	The logging configuration.	None
RefererConfiguration	Map	No	No	The hotlinking protection configuration.	None

Property	Type	Required	Editable	Description	Constraint
DeletionForce	Boolean	No	No	Specifies whether to forcibly delete objects from an OSS bucket	Valid values: <ul style="list-style-type: none"> true false Default value: false.
WebsiteConfiguration	Map	No	No	The information used to configure the bucket as a static website.	None
ServerSideEncryptionConfiguration	Map	No	No	The server-side encryption rules.	None
Tags	Map	No	No	The tags of the bucket. Tags exist as key-value pairs.	<ul style="list-style-type: none"> A maximum of 20 tags can be specified. A tag key must be 1 to 64 bytes in length and cannot start with <code>http://</code>, <code>https://</code>, or <code>Aliyun</code>. A tag value can be up to 128 bytes in length and must be encoded in UTF-8.
StorageClass	String	No	No	The type of the bucket.	Valid values: Standard, IA, and Archive.
Policy	Map	No	No	The bucket policy configuration.	None

CORSConfiguration syntax

```
"CORSConfiguration": {
  "CORSRule": [
    {
      "AllowedHeader": String,
      "AllowedMethod": List,
      "AllowedOrigin": List,
      "ExposeHeader": List,
      "MaxAgeSeconds": Integer
    }
  ]
}
```

CORSConfiguration properties

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

Property	Type	Required	Editable	Description	Constraint
CORSRule	List	No	No	The rules that define cross-origin resource sharing of objects in the bucket.	None
AllowedHeader	String	No	No	The allowed cross-origin request headers.	Valid values: *, Cache-Control, Content-Language, Content-Type, Expires, Last-Modified, and Pragma.
AllowedMethod	List	No	No	The allowed cross-origin request methods.	Valid values: *, GET, PUT, POST, DELETE, and HEAD.
AllowedOrigin	List	No	No	The origins from which cross-origin requests are allowed.	None
ExposeHeader	List	No	No	The response headers for allowed access requests from applications.	Asterisks (*) cannot be used as wildcard characters.
MaxAgeSeconds	Integer	No	No	The period of time that the browser can cache the response of a preflight (OPTIONS) request to a specific resource.	None

LifecycleConfiguration syntax

```

"LifecycleConfiguration": {
  "Rule": [
    {
      "ID": String,
      "Prefix": String,
      "Status": String,
      "Expiration": Map,
      "AbortMultipartUpload": Map
    }
  ]
}

```

LifecycleConfiguration properties

Property	Type	Required	Editable	Description	Constraint
Rule	List	No	No	The lifecycle rule.	None
ID	String	No	No	The unique ID of the rule.	The ID can be up to 255 characters in length. When this parameter is empty or not specified, OSS generates a unique rule ID.
Prefix	String	No	No	The prefix to which the rule applies.	The rule takes effect only on objects that have a matching prefix.
Status	String	No	No	Specifies whether to enable or disable the rule.	Valid values: Enable and Disable.
Expiration	Map	No	No	The expiration attributes of the rule for the specified object.	None

Property	Type	Required	Editable	Description	Constraint
AbortMultipartUpload	Map	No	No	The expiration attributes of the multipart upload tasks that are not complete.	None

Expiration syntax

```
"Expiration":{
  "Days": Number,
  "CreatedBeforeDate": String
}
```

Expiration properties

Property	Type	Required	Editable	Description	Constraint
Days	Number	No	No	The number of days since the object was last modified after which the rule will take effect.	When the number of days since the object was last modified exceeds the specified number of days, the object is deleted. If you set the Days parameter to 30, objects that were last modified on January 1, 2016 are deleted by the backend application on January 31, 2016.
CreatedBeforeDate	String	No	No	The date before which the rule takes effect.	Specify the time in the ISO 8601 standard. The time must be UTC 00:00. Example: 2002-10-11T00:00:00.000Z.

AbortMultipartUpload syntax

```
"AbortMultipartUpload": {
  "CreatedBeforeDate": String,
  "Days": Number
}
```

AbortMultipartUpload properties

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

Property	Type	Required	Editable	Description	Constraint
Days	Number	No	No	The number of days since the object was last modified after which the rule will take effect.	When the number of days since the object was last modified exceeds the specified number of days, the object is deleted. If you set the Days parameter to 30, objects that were last modified on January 1, 2016 are deleted by the backend application on January 31, 2016.
CreatedBeforeDate	String	No	No	The date before which the rule takes effect.	Specify the time in the ISO 8601 standard. The time must be UTC 00:00. Example: 2002-10-11T00:00:00.000Z.

LoggingConfiguration syntax

```
"LoggingConfiguration": {
  "TargetBucket": String,
  "TargetPrefix": String
}
```

LoggingConfiguration properties

Property	Type	Required	Editable	Description	Constraint
TargetBucket	String	No	No	The storage space for storing access logs.	None
TargetPrefix	String	No	No	The prefix of the names of saved access log files.	None

WebsiteConfiguration syntax

```
"WebsiteConfiguration":{
  "IndexDocument": String,
  "ErrorDocument": String
}
```

WebsiteConfiguration properties

Property	Type	Required	Editable	Description	Constraint
IndexDocument	String	No	No	The default homepage for a static website.	None

Property	Type	Required	Editable	Description	Constraint
ErrorDocument	String	No	No	The default error page for a static website.	None

RefererConfiguration syntax

```
"RefererConfiguration":{
  "AllowEmptyReferer": String,
  "RefererList": List
}
```

RefererConfiguration properties

Property	Type	Required	Editable	Description	Constraint
AllowEmptyReferer	String	No	No	Specifies whether the Referer field can be left empty in an access request.	None
RefererList	List	No	No	The referer whitelist. OSS allows requests whose Referer field values are in the referer whitelist.	None

ServerSideEncryptionConfiguration syntax

```
"ServerSideEncryptionConfiguration":{
  "KMSMasterKeyID": String,
  "SSEAlgorithm": String
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
KMSMasterKeyID	String	No	No	The ID of the customer master key.	The key ID is required only when the SSEAlgorithm value is KMS and the specified key is used for encryption.
SSEAlgorithm	String	Yes	No	The default server-side encryption method.	Valid values: KMS and AES256.

Response parameters

Fn::GetAtt

- **Name:** the bucket name, which must be globally unique.
- **DomainName:** the public domain name of the specified bucket.

- `InternalDomainName`: the internal domain name of the specified bucket.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Bucket": {
      "Type": "ALIYUN::OSS::Bucket",
      "Properties": {
        "AccessControl": "private",
        "BucketName": "rostest",
        "WebsiteConfiguration": {
          "IndexDocument": "index1.html",
          "ErrorDocument": "error404.html"
        },
        "LoggingConfiguration": {
          "TargetBucket": "cos-mirror",
          "TargetPrefix": "test404"
        },
        "CORSConfiguration": {
          "CORSRule": [ {
            "AllowedHeader": ["*"],
            "AllowedMethod": ["GET", "PUT"],
            "AllowedOrigin": ["*"],
            "ExposeHeader": ["Date"],
            "MaxAgeSeconds": 3600
          } ]
        },
        "LifecycleConfiguration": {
          "Rule": [ {
            "ID": "deleteRule",
            "Prefix": "test/",
            "Status": "Enabled",
            "Expiration": {
              "Days": 2
            },
            "AbortMultipartUpload": {
              "CreatedBeforeDate": "2014-10-11T00:00:00.000Z"
            }
          } ]
        },
        "RefererConfiguration": {
          "AllowEmptyReferer": true,
          "RefererList": ["http://www.aliyun.com", "https://www?.aliyuncs.com"]
        }
      }
    }
  }
}
```

```

},
"Outputs": {
  "Name": {
    "Value": {"Fn::GetAtt": ["Bucket","Name"]}
  },
  "DomainName": {
    "Value": {"Fn::GetAtt": ["Bucket","DomainName"]}
  }
}
}
}
}

```

5.5.4. RDS

5.5.4.1. ALIYUN::RDS::Account

ALIYUN::RDS::Account is used to create a database management Account.

Statement

```

{
  "Type": "ALIYUN::RDS::Account",
  "Properties": {
    "AccountDescription": String,
    "DBInstanceId": String,
    "AccountPassword": String,
    "AccountType": String,
    "AccountName": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AccountDescription	String	Yes	True	The description of the account.	The name must be 2 to 256 characters in length. It can contain digits, letters, underscores (_), and hyphens (-); but must start with a letter.
DBInstanceId	String	No	No	The ID of the RDS instance.	None
AccountPassword	String	No	No	The password of the database account.	The password must be 8 to 32 characters in length.

Parameter	Type	Required	Editable	Description	Constraint
AccountType	String	Yes	Released	The type of the database account.	Valid values: <ul style="list-style-type: none"> Normal: indicates a standard account. Super: indicates a privileged account. Default value: Normal.
AccountName	String	No	No	The name of the database account.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

AccountName: the name of the database account.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Account": {
      "Type": "ALIYUN::RDS::Account",
      "Properties": {
        "AccountDescription": {
          "Ref": "AccountDescription"
        },
        "DBInstanceId": {
          "Ref": "DBInstanceId"
        },
        "AccountPassword": {
          "Ref": "AccountPassword"
        },
        "AccountType": {
          "Ref": "AccountType"
        },
        "AccountName": {
          "Ref": "AccountName"
        }
      }
    }
  }
}
```

```

}
},
"Parameters": {
  "AccountDescription": {
    "Type": "String",
    "Description": "Account remarks.\nIt cannot begin with http:// or https://.\nIt must start with a Chinese character
or English letter.\nIt can include Chinese and English characters/letters, underscores (_), hyphens (-), and digits.\nThe
length may be 2-256 characters."
  },
  "DBInstanceId": {
    "Type": "String",
    "Description": "RDS instance ID."
  },
  "AccountPassword": {
    "MinLength": 8,
    "Type": "String",
    "Description": "The account password for the database instance. It may consist of letters, digits, or underlines, wit
h a length of 8 to 32 characters.",
    "MaxLength": 32
  },
  "AccountType": {
    "Default": "Normal",
    "Type": "String",
    "Description": "Privilege type of account.\nNormal: Common privilege.\nSuper: High privilege. And the default valu
e is Normal.\nThis parameter is valid for MySQL 5.5/5.6 only.\nMySQL 5.7, SQL Server 2012/2016, PostgreSQL, and PPAS
each can have only one initial account. Other accounts are created by the initial account that has logged on to the dat
abase.",
    "AllowedValues": ["Normal", "Super"]
  },
  "AccountName": {
    "Type": "String",
    "Description": "Account name, which must be unique and meet the following requirements:\nStart with a letter;\nC
onsist of lower-case letters, digits, and underscores (_);\nContain no more than 16 characters.\nFor other invalid char
acters, see Forbidden keywords table."
  }
},
"Outputs": {
  "AccountName": {
    "Description": "Account name",
    "Value": {
      "Fn::GetAtt": ["Account", "AccountName"]
    }
  }
}
}
}

```

5.5.4.2. ALIYUN::RDS::AccountPrivilege

ALIYUN::RDS::AccountPrivilege is used to grant database access permissions to accounts.

Statement

```
{
  "Type": "ALIYUN::RDS::AccountPrivilege",
  "Properties": {
    "AccountPrivilege": String,
    "DBInstancedId": String,
    "DBName": String,
    "AccountName": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
-----------	------	----------	----------	-------------	------------

Parameter	Type	Required	Editable	Description	Constraint
AccountPrivilege	String	No	Yes	The permissions of the database account.	<p>Valid values:</p> <ul style="list-style-type: none"> • ReadWrite: has read and write permissions on the database. • ReadOnly: The account has read-only permission on the database. • DDLOnly: The account can run only data definition language (DDL) commands in the database. This is applicable to MySQL and MariaDB. • DMLOnly: The account can run only data manipulation language (DML) commands in the database. This is applicable to MySQL and MariaDB. • DBOwner: The account has full permissions on the database. This is applicable to SQL Server.
DBInstanceId	String	No	No	The ID of the RDS instance.	None
DBName	String	No	No	The name of the database.	None

Parameter	Type	Required	Editable	Description	Constraint
AccountName	String	No	No	The name of the account.	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

None

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AccountPrivilege": {
      "Type": "ALIYUN::RDS::AccountPrivilege",
      "Properties": {
        "AccountPrivilege": {
          "Ref": "AccountPrivilege"
        },
        "DBInstanceId": [
          "Ref": "DBInstanceId"
        ],
        "DBName": {
          "Ref": "DBName"
        },
        "AccountName": {
          "Ref": "AccountName"
        }
      }
    },
    "Parameters": {
      "AccountPrivilege": {
        "Type": "String",
        "Description": "RDS account privilege",
        "AllowedValues": ["ReadOnly", "ReadWrite", "DDLOnly", "DMLOnly", "DBOwner"]
      },
      "DBInstanceId": [
        "Type": "String",
        "Description": "RDS instance ID."
      ],
      "DBName": {
        "Type": "String",
        "Description": "RDS database name"
      },
      "AccountName": {
        "Type": "String",
        "Description": "RDS account name."
      }
    },
    "Outputs": {}
  }
}

```

5.5.4.3. ALIYUN::RDS::DBInstance

ALIYUN::RDS::DBInstance is used to create an ApsaraDB for RDS instance.

Syntax

```
{
  "Type": "ALIYUN::RDS::DBInstance",
  "Properties": {
    "Engine": String,
    "MultiAZ": Boolean,
    "VpcId": String,
    "DBMappings": List,
    "DBInstanceDescription": String,
    "ConnectionMode": String,
    "MasterUsername": String,
    "MasterUserPassword": String,
    "ZonId": String,
    "DBInstanceNetType": String,
    "DBInstanceStorage": Integer,
    "VSwitchId": String,
    "AllocatePublicConnection": Boolean,
    "EngineVersion": String,
    "PreferredBackupTime": String,
    "DBInstanceClass": String,
    "SecurityIPList": String,
    "BackupRetentionPeriod": Integer,
    "PrivateIpAddress": String,
    "PreferredBackupPeriod": List,
    "PeriodType": String,
    "PayType": String,
    "Period": Integer,
    "ResourceGroupId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None
Engine	String	Yes	No	The type of the database	Valid values: <ul style="list-style-type: none"> MySQL SQLServer PostgreSQL PPAS

Property	Type	Required	Editable	Description	Constraint
DBInstanceStorage	Integer	Yes	Yes	The storage capacity of the instance.	<ul style="list-style-type: none"> Valid values when Engine is set to MySQL: 5 to 1000. Valid values when Engine is set to SQLServer: 10 to 1000. Valid values when Engine is set to PostgreSQL: 5 to 2000. Valid values when Engine is set to PPAS: 5 to 2000. Unit: GB. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ⓘ Note This value must be in 5 GB increments. </div>
EngineVersion	String	Yes	No	The engine version of the database.	Valid values: <ul style="list-style-type: none"> When Engine is set to MySQL, the valid values are 5.5, 5.6, 5.7, and 8.0. When Engine is set to SQLServer, set the value to 2008r2. When Engine is set to PostgreSQL, set the value to 9.4. When Engine is set to PPAS, set the value to 9.3.
DBInstanceClasses	String	Yes	Yes	The type of the instance.	Valid values: <ul style="list-style-type: none"> rds.mys2.large rds.mss1.large rds.pg.s1.small
SecurityIPList	String	Yes	Yes	The whitelist of IP addresses that are allowed to access all databases in the instance.	<ul style="list-style-type: none"> Separate multiple IP addresses with commas (.). Each IP address in the whitelist must be unique. A maximum of 1,000 IP addresses can be specified. The 0.0.0.0/0 format is supported. You can specify IP addresses in the 10.23.XX.XX format and CIDR blocks in the 10.23.XX.XX/24 format. In 10.23.XX.XX/24, /24 indicates the length of the prefix in the CIDR block, and a prefix can be 1 to 32 characters in length. 0.0.0.0/0 indicates that no access restriction is applied.
MultiAZ	Boolean	No	No	Specifies whether the instance can be deployed in multiple zones.	None
VpcId	String	No	No	The ID of the VPC.	None

Property	Type	Required	Editable	Description	Constraint
DBMappings	List	No	No	The list of databases created in the instance.	None
DBInstanceDescription	String	No	No	The description or remarks of the instance.	<ul style="list-style-type: none"> The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.
ConnectionMode	String	No	No	The database connection mode.	Valid values: <ul style="list-style-type: none"> Performance: standard connection mode Safety: safe connection mode If this parameter is not specified, the connection mode assigned by the system is used by default.
MasterUsername	String	No	No	The name for the primary account of the instance.	The name must be unique. The name cannot exceed 16 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.
MasterUserPassword	String	No	No	The password for the primary account of the instance.	The password must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).
ZoneId	String	No	No	The ID of the zone.	None
DBInstanceNetType	String	No	No	The network type of the instance.	Default value: Intranet. Valid value: <ul style="list-style-type: none"> Internet Intranet
VSwitchId	String	No	No	The ID of the VSwitch that is connected to the specified VPC.	None
AllocatePublicConnection	Boolean	No	No	Specifies whether to apply for a public endpoint for the instance.	None

Property	Type	Required	Editable	Description	Constraint
PreferredBackupTime	String	No	No	The backup time.	<ul style="list-style-type: none"> Specify the time in the <code>HH:mmZ- HH:mmZ</code> format. Valid values: 00:00Z-01:00Z, 01:00Z-02:00Z, 02:00Z-03:00Z, 03:00Z-04:00Z, 04:00Z-05:00Z, 05:00Z-06:00Z, 06:00Z-07:00Z, 07:00Z-08:00Z, 08:00Z-09:00Z, 09:00Z-10:00Z, 10:00Z-11:00Z, 11:00Z-12:00Z, 12:00Z-13:00Z, 13:00Z-14:00Z, 14:00Z-15:00Z, 15:00Z-16:00Z, 16:00Z-17:00Z, 17:00Z-18:00Z, 18:00Z-19:00Z, 19:00Z-20:00Z, 20:00Z-21:00Z, 21:00Z-22:00Z, 22:00Z-23:00Z, and 23:00Z-24:00Z.
BackupRetentionPeriod	Number	No	No	The number of days for which backup files can be retained.	Valid values: 7 to 30. Unit: days. Default value: 7.
PrivateIpAddress	String	No	No	The private IP address of the instance on the specified VSwitch.	If this parameter is not specified, the system automatically allocates a private IP address.
PreferredBackupPeriod	List	No	No	The backup periods.	Valid values: <ul style="list-style-type: none"> Monday Tuesday Wednesday Thursday Friday Saturday Sunday
MasterUserType	String	No	No	The type of the primary account.	Default value: Normal. Valid values: <ul style="list-style-type: none"> Normal Super
Tags	Map	No	Yes	The list of tags. Each tag consists of a TagKey and a TagValue.	<ul style="list-style-type: none"> The TagKey is required and the TagValue is optional. Format example: <code>{"key1": "value1", "key2": ""}</code>.
PeriodType	String	No	No	The subscription period type.	Default value: Month. Valid values: <ul style="list-style-type: none"> Month Year

Property	Type	Required	Editable	Description	Constraint
PayType	String	No	No	The billing method of the instance.	Valid values: <ul style="list-style-type: none"> PostPaid: pay-as-you-go PrePaid: subscription
Period	Integer	No	No	The subscription period of the instance.	<ul style="list-style-type: none"> Valid values when the PeriodType parameter is set to Year: 1 to 3. Valid values when the PeriodType parameter is set to Month: 1 to 9.

DBMappings syntax

```
"DBMappings": [
{
  "DBDescription": String,
  "CharacterSetName": String,
  "DBName": String
}
]
```

DBMappings properties

Property	Type	Required	Editable	Description	Constraint
CharacterSetName	String	Yes	No	The character set.	<ul style="list-style-type: none"> Valid values when Engine is set to MySQL: <ul style="list-style-type: none"> utf8 gbk latin1 utf8mb4 (applicable to versions 5.5 and 5.6) Valid values when Engine is set to SQLServer: <ul style="list-style-type: none"> Chinese_PRC_CI_AS Chinese_PRC_CS_AS SQL_Latin1_General_CP1_CI_AS SQL_Latin1_General_CP1_CS_AS Chinese_PRC_BIN
DBName	String	Yes	No	The name of the database.	<p>The name must be unique.</p> <p>The name can be up to 64 characters in length and can contain letters, digits, and underscores (_). It must start with a letter.</p>

Property	Type	Required	Editable	Description	Constraint
DBDescription	String	No	No	The description of the database.	<ul style="list-style-type: none"> The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.

Response parameters

Fn::GetAtt

- **DBInstanceId:** the ID of the RDS instance.
- **InnerPort:** the internal port of the RDS instance.
- **InnerIPAddress:** the internal IP address of the RDS instance.
- **InnerConnectionString:** the internal endpoint of the RDS instance.
- **PublicPort:** the public port of the RDS instance.
- **PublicConnectionString:** the public endpoint of the RDS instance.
- **PublicIPAddress:** the public IP address of the RDS instance.

Examples

The following example demonstrates how to create an ApsaraDB for RDS instance in the classic network:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine": "MySQL",
        "EngineVersion": "5.6",
        "DBInstanceClass": "rds.mysql.t1.small",
        "DBInstanceStorage": 10,
        "DBInstanceNetType": "Internet",
        "SecurityIPList": "0.0.0.0/0",
        "MasterUsername": "A****",
        "DBMappings": [{
          "DBName": "hope",
          "CharacterSetName": "utf8"
        }]
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {"get_attr": ["DBInstanceId"]}
    },
    "PublicConnectionString": {
      "Value": {"get_attr": ["ConnectionString"]}
    },
    "PublicPort": {
      "Value": {"get_attr": ["Port"]}
    }
  }
}
```

The following example demonstrates how to create an ApsaraDB for RDS instance in a VPC:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine": "MySQL",
        "EngineVersion": "5.6",
        "DBInstanceClass": "rds.mys2.small",
        "DBInstanceStorage": "10",
        "DBInstanceNetType": "Intranet",
        "SecurityIPList": "0.0.0.0/0",
        "VSwitchId": "tvt",
        "VpcId": "myvp****"
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {"get_attr": ["DBInstanceId"]}
    },
    "InnerConnectionString": {
      "Value": {"get_attr": ["ConnectionString"]}
    },
    "InnerPort": {
      "Value": {"get_attr": ["Port"]}
    }
  }
}

```

5.5.4.4. ALIYUN::RDS::DBInstanceParameterGroup

ALIYUN::RDS::DBInstanceParameterGroup is used to modify the parameter list of an apsaradb for RDS instance.

Statement

```

{
  "Type": "ALIYUN::RDS::DBInstanceParameterGroup",
  "Properties": {
    "Forcerestart": String,
    "DBInstanceId": String,
    "Parameters": List
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DBInstanceid	String	No	No	The ID of the ApsaraDB RDS instance to query.	None
Parameters	List	Yes	No	parameter.	Parameters in JSON format and their values. Example: { "auto_increment_increment": "1" ,"character_set_client": "utf8" }
Forcerestart	String	Yes	Released	Specifies whether to force a restart of the ApsaraDB for RDS instance.	Valid values: <ul style="list-style-type: none"> • true: specifies to forcibly restart the database. • false: Do not force restart Default value: false.

Response parameters

Fn::GetAtt

None

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Database": {
      "Type": "ALIYUN::RDS::DBInstance",
      "Properties": {
        "Engine": "MySQL",
        "EngineVersion": "5.6",
        "DBInstanceClass": "rds.mys2.small",
        "DBInstanceStorage": "10",
        "DBInstanceNetType": "Intranet",
        "SecurityIPList": "0.0.0.0/0"
      }
    },
    "DatabaseConfig": {
      "Type": "ALIYUN::RDS::DBInstanceParameterGroup",
      "Properties": {
        "DBInstanceId": {
          "Ref": "Database"
        },
        "Parameters": [
          {
            "Key": "auto_increment_increment",
            "Value": "xxx"
          }
        ]
      }
    }
  },
  "Outputs": {
    "DBInstanceId": {
      "Value": {
        "Fn::GetAtt": [
          "Database",
          "DBInstanceId"
        ]
      }
    }
  }
}

```

5.5.4.5. ALIYUN::RDS::DBInstanceSecurityIps

ALIYUN::RDS::DBInstanceSecurityIps is used to modify the instance whitelist.

Statement

```
{
  "Type": "ALIYUN::RDS::DBInstanceSecurityIps",
  "Properties": {
    "DBInstanceID": String,
    "DBInstanceIPArrayName": String,
    "DBInstanceIPArrayAttribute": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
DBInstanceID	String	No	No	The ID of the RDS instance.	None
DBInstanceIPArrayAttribute	String	No	Yes	The attribute of the IP address whitelist.	The console does not display groups labeled with hidden.
DBInstanceIPArrayName	String	Yes	Released	The name of the IP address whitelist.	The name can contain only lowercase letters and underscores (_). Default value: Default.

Response parameters

Fn::GetAtt

SecurityIps: the IP address whitelist after the modification.

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DBInstanceSecurityIps": {
      "Type": "ALIYUN::RDS::DBInstanceSecurityIps",
      "Properties": {
        "DBInstanceIPArrayName": {
          "Ref": "DBInstanceIPArrayName"
        },
        "DBInstanceId": [
          "Ref": "DBInstanceId"
        ],
        "DBInstanceIPArrayAttribute": {
          "Ref": "DBInstanceIPArrayAttribute"
        }
      }
    }
  },
  "Parameters": {
    "DBInstanceIPArrayName": {
      "Type": "String",
      "Description": "Group name of the security ips, only support lower characters and '_'. Advice use a new group name avoid effect your database system. If the properties is not specified, it will set to default group, please be careful."
    },
    "DBInstanceId": [
      "Type": "String",
      "Description": "Database instance id to update security ips."
    ],
    "DBInstanceIPArrayAttribute": {
      "Type": "String",
      "Description": "Security ips to add or remove."
    }
  },
  "Outputs": {
    "SecurityIps": {
      "Description": "The security ips of selected database instance.",
      "Value": {
        "Fn::GetAtt": [
          "DBInstanceSecurityIps",
          "SecurityIps"
        ]
      }
    }
  }
}

```

5.5.4.6. ALIYUN::RDS::PrepayDBInstance

ALIYUN::RDS::PrepayDBInstance is used to create a subscription ApsaraDB for RDS instance.

Statement

```
{
  "Type": "ALIYUN::RDS::PrepayDBInstance",
  "Properties": {
    "DBMappings": List,
    "CouponCode": String,
    "MasterUsername": String,
    "PeriodType": String,
    "PayType": String,
    "DBInstanceNetType": String,
    "MasterUserType": String,
    "AutoRenew": Boolean,
    "PreferredBackupTime": String,
    "PrivateIpAddress": String,
    "Engine": String,
    "MultiAZ": Boolean,
    "VpcId": String,
    "ConnectionMode": String,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "BackupRetentionPeriod": Number,
    "Quantity": Number,
    "CommodityCode": String,
    "ZoneId": String,
    "AutoPay": Boolean,
    "EngineVersion": String,
    "DBInstanceClass": String,
    "PreferredBackupPeriod": List,
    "DBInstanceStorage": Integer,
    "DBInstanceDescription": String,
    "Tags": Map,
    "Period": Number,
    "MasterUserPassword": String,
    "AllocatePublicConnection": Boolean
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
-----------	------	----------	----------	-------------	------------

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the RDS instance belongs.	None
DBMappings	List	Erased	Released	The databases created in the instance.	None
CouponCode	String	Yes	Released	None	None
MasterUsername	String	Yes	Released	The name of the Apsara Stack tenant account for the instance.	The name must be unique. The name must be 1 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter.
PeriodType	String	No	No	The type of the billing cycle.	Valid values: <ul style="list-style-type: none"> Year Month Default value: Month.
DBInstanceNetType	String	Yes	Released	The network type of the instance.	Valid values: <ul style="list-style-type: none"> Internet: used for access from public networks. Intranet: used for private network access. Default value: Intranet.
MasterUserType	String	Yes	Released	The permission type of the account.	Valid values: <ul style="list-style-type: none"> Normal Master
PreferredBackupTime	String	Yes	Released	The preferred backup time.	Specify the time in the HH:mmZ- HH:mmZ format. Valid values: 00:00Z-01:00Z, 01:00Z-02:00Z, 02:00Z-03:00Z, and 03:00Z-04:00 to 23:00Z-24:00Z.
PrivateIpAddress	String	Yes	Released	The private IP address of the instance on the specified VSwitch.	If this parameter is not specified, the system automatically assigns a value.

Parameter	Type	Required	Editable	Description	Constraint
Engine	String	No	No	The database engine that the instance runs.	Valid values: <ul style="list-style-type: none"> MySQL SQLServer PostgreSQL PPAS
MultiAZ	Boolean	Erased	Released	Specifies whether the instance can be deployed across multiple zones.	None
VpcId	String	Yes	Released	The ID of the VPC.	None
ConnectionMode	String	Yes	Released	The access mode of the instance.	Valid values: <ul style="list-style-type: none"> Performance: a standard access mode. Safety: high security access mode. If this parameter is not specified, apsaradb for RDS immediately assigns a value to it. Default value: Safty
AutoRenew	Boolean	Erased	Released	Specifies whether to enable automatic renewal for the instance.	Valid values: <ul style="list-style-type: none"> True False
VSwitchId	String	Yes	Released	The ID of the VSwitch in the specified VPC.	None
BackupRetentionPeriod	Number	Erased	Released	The retention period of backup data. Unit: days.	None
Quantity	Number	Erased	Released	The number of instances to be created.	Valid values: 1 to 99. Default value: 1
CommodityCode	String	No	No	The commodity code.	Valid values: <ul style="list-style-type: none"> rds bards rords
ZoneId	String	Yes	Released	The ID of the zone where the instance resides.	None

Parameter	Type	Required	Editable	Description	Constraint
EngineVersion	String	No	No	The version of the database engine that the instance runs.	The range of valid engine version values varies by database type. <ul style="list-style-type: none"> For an ApsaraDB RDS for MySQL instance, the valid values are 5.5 and 5.6. For an ApsaraDB RDS for SQL Server instance, set the value to 2008r2. For an ApsaraDB RDS for PostgreSQL instance, set the value to 9.4. For an ApsaraDB RDS for PPAS instance, set the value to 9.3.
DBInstanceClass	String	No	Yes	The specification of the instance.	Examples of RDS instance types: rds.mys2.large, rds.ms1.large, and rds.pg.s1.small.
PreferredBackupPeriod	List	Erased	Released	The list of preferred backup periods.	Valid values: <ul style="list-style-type: none"> Monday Tuesday Wednesday Thursday Friday Saturday Sunday
DBInstanceStorage	Integer	Retained	Yes	The storage capacity of the instance.	The range of valid storage capacity values varies by database type. <ul style="list-style-type: none"> MySQL: 5 to 1000 SQL Server: 10 to 1000 PostgreSQL and PPAS: 5 to 2000 Unit: GB. Every 5GB.
DBInstanceDescription	String	Yes	Released	The description of the instance.	The value must be 2 to 256 bytes in length. It must start with a letter and cannot start with http:// or https://. It must start with a letter and cannot start with http:// or https://.
Tags	MAP	No.	True	The tags of the instance.	None

Parameter	Type	Required	Editable	Description	Constraint
Period	Number	Yes	No	The subscription period of the instance.	<ul style="list-style-type: none"> Select monthly payment. Valid values: 1 to 9. Select pay by year. Valid values: 1 to 3.
MasterUserPassword	String	Yes	Released	The password of the Apsara Stack tenant account for the instance.	The password must be 6 to 32 characters in length and can contain letters, digits, and underscores (_).
AllocatePublicConnection	Boolean	Erased	Released	Specifies whether to apply for a public connection string for the instance.	None
PayType	String	Yes	Released	The billing method of the router interface.	Valid values: <ul style="list-style-type: none"> Postpaid: pay-as-you-go Prepaid: Subscription
AutoPay	Boolean	Erased	Released	Specifies whether to enable automatic payment.	Valid values: <ul style="list-style-type: none"> True False Default value: False.

DBMappings syntax

```
"DBMappings": [
  {
    "DBDescription": String,
    "CharacterSetName": String,
    "DBName": String
  }
]
```

DBMappings properties

Parameter	Type	Required	Editable	Description	Constraint
DBDescription	String	Yes	Released	The description of the database.	The description must be 2 to 256 characters in length It cannot start with http:// or https://. It must start with a letter and cannot start with http:// or https://.

Parameter	Type	Required	Editable	Description	Constraint
CharacterSetName	String	No	No	The character set of the database.	Valid values: <ul style="list-style-type: none"> Valid values for MySQL: utf8 gbk latin1 utf8mb4 (applicable to version 5.5 and 5.6). Valid values for an SQL Server database: Chinese_PRC_CI_AS, Chinese_PRC_CS_AS, SQL_Latin1_General_CP1_CI_AS, SQL_Latin1_General_CP1_CS_AS, and Chinese_PRC_BIN.
DBName	String	No	No	The name of the database.	The name must be unique. It can be up to 64 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.

Response parameters

Fn::GetAtt

- **InnerPort**: the internal port of the RDS instance.
- **OrderId**: the ID of the order.
- **PublicConnectionString**: public network connection string.
- **InnerIPAddress**: the internal IP address of the instance.
- **DBInstanceId**: the ID of the RDS instance.
- **PublicIPAddress**: the public IP address of the instance.
- **PublicPort**: The public port of the RDS instance.
- **InnerConnectionString**: intranet connection string.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "PeriodType": {
      "Type": "String",
      "Description": "Charge period for created instances.",
      "AllowedValues": [
        "Month",
        "Year"
      ],
      "Default": "Month"
    },
    "PrivateIpAddress": {
      "Type": "String",
```

```

    "Description": "The private ip for created instance."
  },
  "DBInstanceNetType": {
    "Type": "String",
    "Description": "Database instance net type, default is Intranet.Internet for public access, Intranet for private access.",
    "AllowedValues": [
      "Internet",
      "Intranet"
    ],
    "Default": "Intranet"
  },
  "AutoRenew": {
    "Type": "Boolean",
    "Description": "Auto renew the prepay instance. If the period type is by year, it will renew by year, else it will renew by month.",
    "AllowedValues": [
      "True",
      "true",
      "False",
      "false"
    ],
    "Default": false
  },
  "PreferredBackupPeriod": {
    "Type": "CommaDelimitedList",
    "Description": "Automate backups cycle if automated backups are enabled.",
    "AllowedValues": [
      "Monday",
      "Tuesday",
      "Wednesday",
      "Thursday",
      "Friday",
      "Saturday",
      "Sunday"
    ]
  },
  "DBInstanceStorage": {
    "Type": "Number",
    "Description": "Database instance storage size. mysql is [5,1000]. sql server 2008r2 is [10,1000], sql server 2012/2012_web/2016-web is [20,1000]. PostgreSQL and PPAS is [5,2000]. Increased every 5 GB, Unit in GB"
  },
  "CommodityCode": {
    "Type": "String",
    "Description": "The CommodityCode of the order.",
    "AllowedValues": [
      "rds",

```

```

    "bards",
    "rords"
  ],
  "Default": "rds"
},
"DBMappings": {
  "Type": "CommaDelimitedList",
  "Description": "Database mappings to attach to db instance."
},
"MultiAZ": {
  "Type": "Boolean",
  "Description": "Specifies if the database instance is a multiple Availability Zone deployment. ",
  "AllowedValues": [
    "True",
    "true",
    "False",
    "false"
  ],
  "Default": false
},
"Engine": {
  "Type": "String",
  "Description": "Database instance engine type. Support MySQL/SQLServer/PostgreSQL/PPAS now.",
  "AllowedValues": [
    "MySQL",
    "SQLServer",
    "PostgreSQL",
    "PPAS"
  ]
},
"DBInstanceDescription": {
  "Type": "String",
  "Description": "Description of created database instance."
},
"Tags": {
  "Type": "Json",
  "Description": "The tags of an instance.\nYou should input the information of the tag with the format of the Key-Value, such as {\"key1\": \"value1\", \"key2\": \"value2\", ... \"key5\": \"value5\"}.\nAt most 5 tags can be specified.\nKey-Value can be up to 64 characters in length.\nCannot begin with aliyun.\nCannot begin with http:// or https://.\nCannot be a null string.\nValue-Value can be up to 128 characters in length.\nCannot begin with aliyun.\nCannot begin with http:// or https://.\nCan be a null string."
},
"EngineVersion": {
  "Type": "String",
  "Description": "Database instance version of the relative engine type.Support MySQL: 5.5/5.6/5.7; SQLServer: 2008r2, 20012, 2012_web, 2012_std_ha, 2012_ent_ha, 2016_web, 2016_std_ha, 2016_ent_ha; PostgreSQL:9.4; PPAS: 9.3.",
  "AllowedValues": [

```

```

AllowedValues: [
  "5.5",
  "5.6",
  "5.7",
  "2008r2",
  "2012",
  "2012_web",
  "2012_std_ha",
  "2012_ent_ha",
  "2016_web",
  "2016_std_ha",
  "2016_ent_ha",
  "9.4",
  "9.3"
]
},
"ZoneId": {
  "Type": "String",
  "Description": "selected zone to create database instance. You cannot set the ZoneId parameter if the MultiAZ parameter is set to true."
},
"DBInstanceClass": {
  "Type": "String",
  "Description": "Database instance type. Refer the RDS database instance type reference, such as 'rds.mys2.large', 'rds.mss1.large', 'rds.pg.s1.small' etc"
},
"AllocatePublicConnection": {
  "Type": "Boolean",
  "Description": "If true, allocate public connection automate.",
  "AllowedValues": [
    "True",
    "true",
    "False",
    "false"
  ]
},
"PreferredBackupTime": {
  "Type": "String",
  "Description": "The daily time range during which automated backups are created if automated backups are enabled.",
  "AllowedValues": [
    "00:00Z-01:00Z",
    "01:00Z-02:00Z",
    "02:00Z-03:00Z",
    "03:00Z-04:00Z",
    "04:00Z-05:00Z",
    "05:00Z-06:00Z",

```

```
"06:00Z-07:00Z",
"07:00Z-08:00Z",
"08:00Z-09:00Z",
"09:00Z-10:00Z",
"10:00Z-11:00Z",
"11:00Z-12:00Z",
"12:00Z-13:00Z",
"13:00Z-14:00Z",
"14:00Z-15:00Z",
"15:00Z-16:00Z",
"16:00Z-17:00Z",
"17:00Z-18:00Z",
"18:00Z-19:00Z",
"19:00Z-20:00Z",
"20:00Z-21:00Z",
"21:00Z-22:00Z",
"22:00Z-23:00Z",
"23:00Z-24:00Z"
]
},
"VSwitchId": {
  "Type": "String",
  "Description": "The vSwitch id of created instance. For VPC network, the property is required."
},
"Quantity": {
  "Type": "Number",
  "Description": "The number of instance to be created, default is 1, max number is 99",
  "MinValue": 1,
  "MaxValue": 99,
  "Default": 1
},
"Period": {
  "Type": "Number",
  "Description": "Prepaid time period. While choose by pay by month, it could be from 1 to 9. While choose pay by year
, it could be from 1 to 3.",
  "MinValue": 1,
  "MaxValue": 9,
  "Default": 1
},
"MasterUserPassword": {
  "Type": "String",
  "Description": "The master password for the database instance. ",
  "MinLength": 8,
  "MaxLength": 32
},
"CouponCode": {
  "Type": "String",
```

```

    "Description": "The coupon code of the order."
  },
  "MasterUserType": {
    "Type": "String",
    "Description": "Privilege type of account.\n Normal: Common privilege. \n Super: High privilege. And the default value is Normal.This parameter is valid for MySQL 5.5/5.6 only. MySQL 5.7, SQL Server 2012/2016, PostgreSQL, and PPAS each can have only one initial account. \nOther accounts are created by the initial account that has logged on to the database.",
    "AllowedValues": [
      "Normal",
      "Super"
    ],
    "Default": "Normal"
  },
  "VpcId": {
    "Type": "String",
    "Description": "The VPC id of created database instance. For VPC network, the property is required."
  },
  "MasterUsername": {
    "Type": "String",
    "Description": "The master user name for the database instance. "
  },
  "ConnectionMode": {
    "Type": "String",
    "Description": "Connection Mode for database instance,support 'Performance' and 'Safty' mode. Default is RDS system assigns. ",
    "AllowedValues": [
      "Performance",
      "Safty"
    ]
  },
  "BackupRetentionPeriod": {
    "Type": "Number",
    "Description": "The number of days for which automatic DB backups are retained.",
    "MinValue": 7,
    "MaxValue": 30,
    "Default": 7
  },
  "Resources": {
    "PrepayDBInstance": {
      "Type": "ALIYUN::RDS::PrepayDBInstance",
      "Properties": {
        "PeriodType": {
          "Ref": "PeriodType"
        }
      },
      "PrivateIpAddress": {

```

```

PrivateIpAddress : {
  "Ref": "PrivateIpAddress"
},
DBInstanceNetType: {
  "Ref": "DBInstanceNetType"
},
AutoRenew: {
  "Ref": "AutoRenew"
},
PreferredBackupPeriod: {
  "Fn::Split": [
    ",",
    {
      "Ref": "PreferredBackupPeriod"
    }
  ],
  {
    "Ref": "PreferredBackupPeriod"
  }
]
},
DBInstanceStorage: {
  "Ref": "DBInstanceStorage"
},
CommodityCode: {
  "Ref": "CommodityCode"
},
DBMappings: {
  "Fn::Split": [
    ",",
    {
      "Ref": "DBMappings"
    }
  ],
  {
    "Ref": "DBMappings"
  }
]
},
MultiAZ: {
  "Ref": "MultiAZ"
},
Engine: {
  "Ref": "Engine"
},
DBInstanceDescription: {
  "Ref": "DBInstanceDescription"
},
Tags: {

```

```
"Ref": "Tags"
},
"EngineVersion": {
  "Ref": "EngineVersion"
},
"ZoneId": {
  "Ref": "ZoneId"
},
"DBInstanceClass": {
  "Ref": "DBInstanceClass"
},
"AllocatePublicConnection": {
  "Ref": "AllocatePublicConnection"
},
"PreferredBackupTime": {
  "Ref": "PreferredBackupTime"
},
"VSwitchId": {
  "Ref": "VSwitchId"
},
"Quantity": {
  "Ref": "Quantity"
},
"Period": {
  "Ref": "Period"
},
"MasterUserPassword": {
  "Ref": "MasterUserPassword"
},
"CouponCode": {
  "Ref": "CouponCode"
},
"MasterUserType": {
  "Ref": "MasterUserType"
},
"VpcId": {
  "Ref": "VpcId"
},
"MasterUsername": {
  "Ref": "MasterUsername"
},
"ConnectionMode": {
  "Ref": "ConnectionMode"
},
"BackupRetentionPeriod": {
  "Ref": "BackupRetentionPeriod"
}
}
```

```
    }
  }
},
"Outputs": {
  "InnerConnectionString": {
    "Description": "DB instance connection url by Intranet.",
    "Value": {
      "Fn::GetAtt": [
        "PrepayDBInstance",
        "InnerConnectionString"
      ]
    }
  },
  "DBInstanceId": [
    "Description": "The instance id of created database instance.",
    "Value": {
      "Fn::GetAtt": [
        "PrepayDBInstance",
        "DBInstanceId"
      ]
    }
  },
  "InnerIPAddress": {
    "Description": "IP Address for created DB instance of Intranet.",
    "Value": {
      "Fn::GetAtt": [
        "PrepayDBInstance",
        "InnerIPAddress"
      ]
    }
  },
  "PublicConnectionString": {
    "Description": "DB instance connection url by Internet.",
    "Value": {
      "Fn::GetAtt": [
        "PrepayDBInstance",
        "PublicConnectionString"
      ]
    }
  },
  "PublicIPAddress": {
    "Description": "IP Address for created DB instance of Internet.",
    "Value": {
      "Fn::GetAtt": [
        "PrepayDBInstance",
        "PublicIPAddress"
      ]
    }
  }
}
```



```
{
  "Type": "ALIYUN::ROS::WaitCondition",
  "Properties": {
    "Count": Number,
    "Handle": String,
    "Timeout": Number
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Handle	String	No	No	Reference ALIYUN::ROS::WaitConditionHandle.	None
Timeout	Number	Yes	No	The length of time to wait for UserData messages.	Valid values: 1 to 43200. Unit: seconds.
Count	Number	No.	True	The total number of messages to be received.	None

Response parameters

Fn::GetAtt

- **Data:** A JSON-serialized dictionary that contains the signal Data after the most recent stack creation or update.
- **LastData:** a JSON-serialized dictionary that contains the signal data before the most recent stack update.
- **JoinedErrorData:** a string consisting of the ErrorData signal data.
- **JoinedLastErrorData:** a string consisting of the LastErrorData signal data.
- **ErrorData:** a JSON-serialized dictionary that contains the error signal data after the most recent stack creation or update.
- **Lasterprotodata:** a JSON-serialized dictionary that contains the error signal data before the most recent stack update.

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "WaitCondition": {
      "Type": "ALIYUN::ROS::WaitCondition",
      "Properties": {
        "Handle": {
          "Ref": "WaitConHandle"
        },
        "Timeout": 5,
        "Count": 2
      }
    },
    "WaitConHandle": {
      "Type": "ALIYUN::ROS::WaitConditionHandle"
    }
  },
  "Outputs": {
    "CurlCli": {
      "Value": {
        "Fn::GetAtt": [
          "WaitConHandle",
          "CurlCli"
        ]
      }
    },
    "Data": {
      "Value": {
        "Fn::GetAtt": [
          "WaitCondition",
          "Data"
        ]
      }
    }
  }
}

```

5.5.5.2. ALIYUN::ROS::WaitConditionHandle

ALIYUN::ROS::WaitConditionHandle is used to create an instance that sends and receives messages during UserData execution.

Statement

```
{
  "Type": "ALIYUN::ROS::WaitConditionHandle",
  "Properties": {
    "Count": Integer,
    "Mode": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Count	Integer	No.	True	The total number of messages to be received.	Default value: -1.
Mode	String	Yes	True	If you set this parameter to Increment, all previous signals will be updated before they are deleted. If you set this parameter to Full, no previous signals will be deleted unless the Count parameter is specified.	Valid values: <ul style="list-style-type: none"> Increment Full Default value: Full.

Response parameters

Fn::GetAtt

- **CurlCli:** A curl Command is generated by the resource. You can use the command to send the UserData execution result or status to Resource Orchestration Service.
- **WindowsCurlCli:** provides Windows with cURL CLI command prefixes and sends a message indicating that the execution is completed or failed. Windows does not support the curl command. Therefore, you must install curl.exe and add it to PATH. You can add `--data-binary "{\"status\": \" success \"}` to indicate success, or by adding `--data-binary "{\"status\": \" failure \"}` to indicate failure.
- **PowerShellCurlCli:** provides PowerShell with cURL CLI command prefixes and sends a message indicating that the execution is completed or failed. Because this cmdlet was introduced in PowerShell 3.0, make sure that the PowerShell version meets this constraint. By `$PSVersionTable.PSVersion` displays the version. You can add `-Body {"status": "success "}` to indicate success, or by adding `-Body {"status": "failure "}` to indicate failure.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "Mode": {
      "Type": "String",
```

"Description": "If set to Increment, all old signals will be deleted before update. In this mode, WaitCondition.Count should reference an incremental value instead of a full value, such as ScalingGroupEnable.ScalingRuleArisExecuteResultNumberOfAddedInstances.\n\nIf set to Full, no old signal will be deleted unless Count is set. In this mode, WaitCondition.Count should reference a full value, such as the same value with InstanceGroup.MaxAmount. It is recommended to use this mode with Count.\n\nDefault to Full.",

```
"AllowedValues": [
```

```
  "Increment",
```

```
  "Full"
```

```
],
```

```
"Default": "Full"
```

```
},
```

```
"Count": {
```

```
  "Type": "Number",
```

"Description": "There are 3 preconditions that make Count taking effect:\n1.Mode is set to Full.\n2.Count >= 0.\n3.The id of signal is not specified. If so, it will be a self-increasing integer started from 1. For example, the id of the first signal is 1, the id of the second signal is 2, and so on.\n\nIf Count takes effect, signals with id > Count will be deleted before update.\n\nThe default value is -1, which means no effect.\n\nIt is recommended to quote the same value with WaitCondition.Count.",

```
  "Default": -1
```

```
}
```

```
},
```

```
"Resources": {
```

```
  "WaitConditionHandle": {
```

```
    "Type": "ALIYUN::ROS::WaitConditionHandle",
```

```
    "Properties": {
```

```
      "Mode": {
```

```
        "Ref": "Mode"
```

```
      },
```

```
      "Count": {
```

```
        "Ref": "Count"
```

```
      }
```

```
    }
```

```
  }
```

```
},
```

```
"Outputs": {
```

```
  "CurlCli": {
```

"Description": "Convenience attribute, provides curl CLI command prefix, which can be used for signalling handle completion or failure. You can signal success by adding --data-binary '{"status": "SUCCESS"}', or signal failure by adding --data-binary '{"status": "FAILURE"}",

```
  "Value": {
```

```
    "Fn::GetAtt": [
```

```
      "WaitConditionHandle",
```

```
      "CurlCli"
```

```
    ]
```

```
  }
```

```
},
```

```
"WindowsCurlCli": {
```

```

    "Description": "Convenience attribute, provides curl CLI command prefix for Windows, which can be used for signalling handle completion or failure. As Windows does not support curl command, you need to install curl.exe and add it to PATH first. You can signal success by adding --data-binary \"{\\\"status\\\": \\\"SUCCESS\\\"}\" , or signal failure by adding --data-binary \"{\\\"status\\\": \\\"FAILURE\\\"}\" ",
    "Value": {
      "Fn::GetAtt": [
        "WaitConditionHandle",
        "WindowsCurlCli"
      ]
    }
  },
  "PowerShellCurlCli": {
    "Description": "Convenience attribute, provides curl CLI command prefix for PowerShell, which can be used for signalling handle completion or failure. As this cmdlet was introduced in PowerShell 3.0, ensure the version of PowerShell satisfies the constraint. (Show the version via $PSVersionTable.PSVersion.) You can signal success by adding -Body '{\"status\": \"SUCCESS\"}' , or signal failure by adding -Body '{\"status\": \"FAILURE\"}' ",
    "Value": {
      "Fn::GetAtt": [
        "WaitConditionHandle",
        "PowerShellCurlCli"
      ]
    }
  }
}
}
}
}
}

```

5.5.5.3. ALIYUN::ROS::Stack

ALIYUN::ROS::Stack is used to create a nested stack. You can have a maximum of five nested levels.

ALIYUN::ROS::Stack is used in a top-level template to nest stacks as resources.

You can add output values from a nested stack contained within the template. You can use Fn::GetAtt together with the logical name of the nested stack and the output name in the Outputs.NestedStackOutputName format.

 **Note** We recommend that you run an update to the Nested stack from the parent stack.

When you apply a template change to update a top-level stack, ROS updates the top-level stack and initiates an update to its nested stacks. Resource orchestration service (ROS) updates resources that have been modified in the nested stack, but does not update resources that have not been modified in the nested stack.

Statement

```
{
  "Type": "ALIYUN::ROS::Stack",
  "Properties": {
    "TemplateURL": String,
    "TimeoutMins": Number,
    "Parameters": Map
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
TemplateURL	String	No	Yes	<p>The URL of the file containing the template body. The template file can be up to 524,288 bytes in size.</p> <p>The URL must point to a template located on the http or https Web server or Alibaba Cloud OSS bucket.</p> <p>For example: <code>oss://ros/tem plate/demo</code> , <code>oss://ros/tem plate/demo? RegionId=cn- hangzhou</code> .</p> <p>If the region of the OSS bucket is not specified, the RegionId of the stack is used.</p>	The URL can be up to 1,024 bytes in length.
TimeoutMins	Number	No.	True	The length of time that ROS will wait for the nested stack to be created or updated.	Unit: minutes. Default value: 60.

Parameter	Type	Required	Editable	Description	Constraint
Parameters	Map	No.	True	A set of value pairs that represent the parameters passed to ROS when this Nested stack is created. Each parameter has a name corresponding to a parameter defined in the embedded template and the value to which you want to set the parameter. This parameter is required if the nested stack needs input parameters.	None

Response parameters

Fn::GetAtt

You can use the following code to obtain the output of the nested stack:

```
{
  "Fn::GetAtt": [
    "<nested_stack>",
    "Outputs.<nested_stack_output_name>"
  ]
}
```

When you use `Ref` to reference resources in a nested stack, the Alibaba Cloud Resource Name (ARN) of the nested stack is returned. Example: `arn:acs:ros::cn-hangzhou:12345****:stacks/test-nested-stack-Demo-jzkyq7mn2ykj/e71c1e04-1a57-46fc-b9a4-cf7ce0d3****`

Examples

- The following code provides an example of how to create a VPC, a VSwitch, and a security group in a nested stack and save the output results to the `oss://ros/template/vpc.txt` directory:

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Description": "One VPC, VSwitch, security group.",
  "Parameters": {
    "ZoneId": {
      "Type": "String",
      "Description": "The available zone"
    },
    "SecurityGroupName": {
```

```

SecurityGroupName : {
  "Type": "String",
  "Description": "The security group name",
  "Default": "my-sg-name"
},
VpcName: {
  "Type": "String",
  "Description": "The VPC name",
  "MinLength": 2,
  "MaxLength": 128,
  "ConstraintDescription": "[2, 128] English or Chinese letters",
  "Default": "my-vpc-name"
},
VpcCidrBlock: {
  "Type": "String",
  "AllowedValues": [
    "192.168.0.0/16",
    "172.16.0.0/12",
    "10.0.0.0/8"
  ],
  "Default": "10.0.0.0/8"
},
VSwitchCidrBlock: {
  "Type": "String",
  "Description": "The VSwitch subnet which must be within VPC",
  "Default": "10.0.10.0/24"
},
UpdateVersion: {
  "Type": "Number",
  "Default": 0
}
},
Resources: {
  Vpc: {
    "Type": "ALIYUN::ECS::VPC",
    "Properties": {
      CidrBlock: {
        "Ref": "VpcCidrBlock"
      },
      VpcName: {
        "Ref": "VpcName"
      }
    }
  },
  VSwitch: {
    "Type": "ALIYUN::ECS::VSwitch",
    "Properties": {

```

```
"CidrBlock": {
  "Ref": "VSwitchCidrBlock"
},
"ZoneId": {
  "Ref": "ZoneId"
},
"VpcId": {
  "Fn::GetAtt": [
    "Vpc",
    "VpcId"
  ]
}
},
"SecurityGroup": {
  "Type": "ALIYUN::ECS::SecurityGroup",
  "Properties": {
    "SecurityGroupName": {
      "Ref": "SecurityGroupName"
    },
    "VpcId": {
      "Ref": "Vpc"
    }
  }
},
"WaitConditionHandle": {
  "Type": "ALIYUN::ROS::WaitConditionHandle",
  "Properties": {
    "UpdateVersion": {
      "Ref": "UpdateVersion"
    }
  }
}
},
"Outputs": {
  "SecurityGroupId": {
    "Value": {
      "Fn::GetAtt": [
        "SecurityGroup",
        "SecurityGroupId"
      ]
    }
  },
  "VpcId": {
    "Value": {
      "Fn::GetAtt": [
        "Vpc",
```

```

    "VpcId"
  ]
}
},
"VSwitchId": {
  "Value": {
    "Fn::GetAtt": [
      "VSwitch",
      "VSwitchId"
    ]
  }
}
}
}
}

```

- The following code provides an example of a top-level stack:

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Description": "One ECS instance.",
  "Parameters": {
    "ImageId": {
      "Default": "centos_7",
      "Type": "String",
      "Description": "Image Id, represents the image resource to startup the ECS instance"
    },
    "InstanceType": {
      "Type": "String",
      "Description": "The ECS instance type,",
      "Default": "ecs.xn4.small"
    },
    "ZoneId": {
      "Type": "String",
      "Description": "The available zone "
    },
    "InstanceChargeType": {
      "Type": "String",
      "AllowedValues": [
        "PrePaid",
        "PostPaid"
      ],
      "Default": "PostPaid",
      "Description": "The instance charge type"
    },
    "SecurityGroupName": {
      "Type": "String",
      "Description": "The security group name",
      "Default": "sg-xxxxxxx"
    }
  }
}

```

```

    "Default": "my-sg-name"
  },
  "NetworkInterfaceName": {
    "Type": "String",
    "Description": "The Network interface name",
    "Default": "my-eni-name"
  },
  "VpcName": {
    "Type": "String",
    "Description": "The VPC name",
    "MinLength": 2,
    "MaxLength": 128,
    "ConstraintDescription": "[2, 128] English or Chinese letters",
    "Default": "my-vpc-name"
  },
  "IoOptimized": {
    "AllowedValues": [
      "none",
      "optimized"
    ],
    "Description": "IO optimized, optimized is for the IO optimized instance type",
    "Type": "String",
    "Default": "optimized"
  },
  "SystemDiskCategory": {
    "AllowedValues": [
      "cloud",
      "cloud_efficiency",
      "cloud_ssd"
    ],
    "Description": "System disk category: average cloud disk(cloud), efficient cloud disk(cloud_efficiency) or SSD cloud disk(cloud_ssd)",
    "Type": "String",
    "Default": "cloud_ssd"
  },
  "VpcCidrBlock": {
    "Type": "String",
    "AllowedValues": [
      "192.168.0.0/16",
      "172.16.0.0/12",
      "10.0.0.0/8"
    ],
    "Default": "10.0.0.0/8"
  },
  "VSwitchCidrBlock": {
    "Type": "String",
    "Description": "The VSwitch subnet which must be within VPC",

```

```
"Default": "10.0.10.0/24"
},
"UpdateVersion": {
  "Type": "Number",
  "Default": 0
}
},
"Resources": {
  "NetworkStack": {
    "Type": "ALIYUN::ROS::Stack",
    "Properties": {
      "TemplateURL": "oss://ros/template/vpc.txt",
      "TimeoutMins": 5,
      "Parameters": {
        "ZoneId": {
          "Ref": "ZoneId"
        },
        "SecurityGroupName": {
          "Ref": "SecurityGroupName"
        },
        "VpcName": {
          "Ref": "VpcName"
        },
        "VpcCidrBlock": {
          "Ref": "VpcCidrBlock"
        },
        "VSwitchCidrBlock": {
          "Ref": "VSwitchCidrBlock"
        },
        "UpdateVersion": {
          "Ref": "UpdateVersion"
        }
      }
    }
  },
  "WebServer": {
    "Type": "ALIYUN::ECS::Instance",
    "Properties": {
      "ImageId": {
        "Ref": "ImageId"
      },
      "InstanceType": {
        "Ref": "InstanceType"
      },
      "InstanceChargeType": {
        "Ref": "InstanceChargeType"
      }
    }
  }
}
```

```
"SecurityGroupid": {
  "Fn::GetAtt": [
    "NetworkStack",
    "Outputs.SecurityGroupid"
  ]
},
"Vpcid": {
  "Fn::GetAtt": [
    "NetworkStack",
    "Outputs.Vpcid"
  ]
},
"VSwitchid": {
  "Fn::GetAtt": [
    "NetworkStack",
    "Outputs.VSwitchid"
  ]
},
"ioOptimized": {
  "Ref": "IoOptimized"
},
"Zoneid": {
  "Ref": "Zoneid"
},
"SystemDisk_Category": {
  "Ref": "SystemDiskCategory"
},
"DiskMappings": [
  {
    "Category": "cloud_ssd",
    "Size": 20
  }
]
}
},
"Outputs": {
  "Instanceid": {
    "Value": {
      "Fn::GetAtt": [
        "WebServer",
        "Instanceid"
      ]
    }
  }
},
"Publicip": {
```

```
    "Value": {
      "Fn::GetAtt": [
        "WebServer",
        "PublicIp"
      ]
    },
    "SecurityGroupId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.SecurityGroupId"
        ]
      }
    },
    "VpcId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.VpcId"
        ]
      }
    },
    "VSwitchId": {
      "Value": {
        "Fn::GetAtt": [
          "NetworkStack",
          "Outputs.VSwitchId"
        ]
      }
    },
    "NetworkStackArn": {
      "Value": {
        "Ref": "NetworkStack"
      }
    }
  }
}
```

5.5.6. SLB

5.5.6.1. ALIYUN::SLB::AccessControl

ALIYUN::SLB::AccessControl is used to create an access control list (ACL).

Syntax

```
{
  "Type": "ALIYUN::SLB::AccessControl",
  "Properties": {
    "AddressIPVersion": String,
    "AclName": String,
    "AclEntrys": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
AddressIPVersion	String	No	No	The Internet protocol version.	Valid values: ipv4 and ipv6.
AclName	String	Yes	Yes	The name of the ACL.	None
AclEntrys	List	No	No	The list of ACL entries.	A list can contain up to 50 ACL entries.

AclEntrys syntax

```
"AclEntrys": [
  {
    "comment": String,
    "entry": String
  }
]
```

AclEntrys properties

Property	Type	Required	Editable	Description	Constraint
comment	String	No	No	The comments on ACL entries.	None
entry	String	Yes	No	The authorized IP addresses or CIDR blocks.	None

Response parameters

Fn::GetAtt

AclId: the ID of the ACL.

Examples

Resource usage example

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
```

```

"Resources": {
  "AccessControl": {
    "Type": "ALIYUN::SLB::AccessControl",
    "Properties": {
      "AddressIPVersion": {
        "Ref": "AddressIPVersion"
      },
      "AclName": {
        "Ref": "AclName"
      },
      "AclEntrys": {
        "Fn::Split": [",", {
          "Ref": "AclEntrys"
        }], {
          "Ref": "AclEntrys"
        }
      ]
    }
  }
},
"Parameters": {
  "AddressIPVersion": {
    "Type": "String",
    "Description": "IP version. Could be \"ipv4\" or \"ipv6\".",
    "AllowedValues": ["ipv4", "ipv6"]
  },
  "AclName": {
    "Type": "String",
    "Description": "The name of the access control list."
  },
  "AclEntrys": {
    "Type": "CommaDelimitedList",
    "Description": "A list of acl entrys. Each entry can be IP addresses or CIDR blocks. Max length: 50.",
    "MaxLength": 50
  }
},
"Outputs": {
  "AclId": {
    "Description": "The ID of the access control list.",
    "Value": {
      "Fn::GetAtt": ["AccessControl", "AclId"]
    }
  }
}
}

```

Example of combined use of SLB-related resources

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "slb-with-listener-and-acl",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth",
        "Bandwidth": 10,
        "VpcId": "vpc-xxxxxxxxxxxxxxxxxxxx",
        "VSwitchId": "vsw-xxxxxxxxxxxxxxxxxxxx"
      }
    },
    "ACL": {
      "Type": "ALIYUN::SLB::AccessControl",
      "Properties": {
        "AclName": "acl-for-listener",
        "AddressIPVersion": "ipv4",
        "AclEntries": [
          {
            "entry": "192.168.x.x"
          },
          {
            "entry": "10.0.x.x/24",
            "comment": "just comment"
          }
        ]
      }
    },
    "CreateListener": {
      "Type": "ALIYUN::SLB::Listener",
      "Properties": {
        "LoadBalancerId": {
          "Ref": "LoadBalancer"
        },
        "ListenerPort": "80",
        "BackendServerPort": 8080,
        "Bandwidth": 1,
        "Protocol": "http",
        "HealthCheck": {
          "HealthyThreshold": 3,
          "UnhealthyThreshold": 3,
          "Interval": 2,
          "Timeout": 5
        }
      }
    }
  }
}
```

```

    },
    "Scheduler": "wrr",
    "RequestTimeout": 179,
    "IdleTimeout": 59,
    "AclId": {
      "Ref": "ACL"
    },
    "AclStatus": "on",
    "AclType": "white"
  }
}
},
"Outputs": {
  "LoadBalanceDetails": {
    "Value": {
      "Fn::GetAtt": [
        "LoadBalancerId",
        "Listeners"
      ]
    }
  }
}
}
}
}
}

```

5.5.6.2. ALIYUN::SLB::BackendServerAttachment

ALIYUN::SLB::BackendServerAttachment is used to add backend servers.

Statement

```

{
  "Type": "ALIYUN::SLB::BackendServerAttachment",
  "Properties": {
    "LoadBalancerId": String,
    "BackendServers": List,
    "BackendServerList": List,
    "BackendServerWeightList": List
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
LoadBalancerId	String	No	No	The unique ID of the SLB instance.	None

Parameter	Type	Required	Editable	Description	Constraint
BackendServerList	List	No.	True	The list of backend servers to add.	You can call this operation with LoadBalancerId and BackendServerWeightList. Separate ECS instance IDs with commas (,). This parameter is ignored when the BackendServers parameter is specified.
BackendServerWeightList	List	No.	True	The weights of the ECS instances in the BackendServerList, which are specified in order.	If this parameter is not specified, the weight of all ECS instances included in the BackendServerList is 100. When the BackendServerWeightList length is less than BackendServerList, the last value in the BackendServerWeightList is used to weight the remaining ECS instances in the BackendServerList.
BackendServers	List	No.	True	The list of backend servers to add.	Only backend servers in the running state can be attached to the SLB instance.

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId" : String,
    "Weight" : Integer
  }
]
```

BackendServers properties

Parameter	Type	Required	Editable	Description	Constraint
ServerId	String	No	Yes	The ID of the ECS instance that acts as a backend server.	The ECS instance must be in the Running state.
Weight	Integer	Retained	Yes	The weight of the ECS instance in the SLB instance.	Valid values: 0 to 100. Default value: 100.

Response parameters

Fn::GetAtt

- **BackendServers:** the backend servers added to the SLB instance.
- **LoadBalancerId:** the ID of the SLB instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Attachment2": {
      "Type": "ALIYUN::SLB::BackendServerAttachment",
      "Properties": {
        "LoadBalancerId": "15187200816-cn-beijing-btc-****",
        "BackendServerList": [
          "i-25o0m****",
          "i-25zsk****"
        ],
        "BackendServerWeightList": [
          "20",
          "100"
        ]
      }
    }
  }
}
```

5.5.6.3. ALIYUN::SLB::BackendServerToVServerGroupAddition

ALIYUN::SLB::BackendServerToVServerGroupAddition is used to add backend servers to an existing VServer group.

Statement

```
{
  "Type": "ALIYUN::SLB::BackendServerToVServerGroupAddition",
  "Properties": {
    "BackendServers": List,
    "VServerGroupId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
VServerGroupId	String	No	No	The ID of the VServer group.	None
BackendServers	List	Retained	Yes	The list of ECS instances to be added.	None

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId": String,
    "Port": Integer,
    "Weight": Integer
  }
]
```

BackendServers properties

Parameter	Type	Required	Editable	Description	Constraint
ServerId	String	No	Yes	The ID of the ECS instance that acts as a backend server.	None
Port	Integer	Retained	Yes	The ECS port number that is listened to in the server load balancer instance.	Valid values: 1 to 65535.
Weight	Integer	Retained	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

VServerGroupId: the ID of the VServer group.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "AttachVServerGroup": {
      "Type": "ALIYUN::SLB::BackendServerToVServerGroupAddition",
      "Properties": {
        "VServerGroupId": "sg-2zenh4ndwrqg14yt0****",
        "BackendServers": [
          {
            "ServerId": "i-25zsk****",
            "Weight": 20,
            "Port": 8080
          },
          {
            "ServerId": "i-25zsk****",
            "Weight": 100,
            "Port": 8081
          }
        ]
      }
    }
  }
}
```

5.5.6.4. ALIYUN::SLB::Certificate

ALIYUN::SLB::Certificate is used to upload a certificate to an SLB instance. Server certificates and CA certificates are supported.

Notice

- You can upload only one CA certificate at a time ("CertificateType": "CA").
- You can upload only one server certificate and the corresponding private key at a time ("CertificateType": "Server").

Syntax

```
{
  "Type": "ALIYUN::SLB::Certificate",
  "Properties": {
    "CertificateName": String,
    "Certificate": String,
    "AliCloudCertificateName": String,
    "PrivateKey": String,
    "ResourceGroupId": String,
    "CertificateType": String,
    "AliCloudCertificateId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	No	No	The ID of the resource group.	None
CertificateName	String	No	Yes	The name of the certificate.	None
Certificate	String	Yes	No	The public key of the certificate.	None
AliCloudCertificateName	String	No	No	The name of the Alibaba Cloud certificate.	None
PrivateKey	String	No	No	The server private key that you want to upload.	None
AliCloudCertificateId	String	No	No	The ID of the Alibaba Cloud certificate.	This parameter is required if you use a certificate from Alibaba Cloud SSL Certificates Service.
CertificateType	String	No	No	The type of the certificate.	Valid values: Server and CA.

Response parameters

Fn::GetAtt

- CertificateId: the ID of the certificate.
- Fingerprint: the fingerprint of the certificate.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01"
```

```

ROSTemplateFormatVersion : 2015-09-01 ,
"Parameters": {
  "CertificateType": {
    "Type": "String",
    "Description": "The type of the certificate.",
    "AllowedValues": [
      "Server",
      "CA"
    ],
    "Default": "Server"
  },
  "AliCloudCertificateName": {
    "Type": "String",
    "Description": "The name of the Alibaba Cloud certificate."
  },
  "PrivateKey": {
    "Type": "String",
    "Description": "The private key."
  },
  "CertificateName": {
    "Type": "String",
    "Description": "The name of the certificate."
  },
  "Certificate": {
    "Type": "String",
    "Description": "The content of the certificate public key."
  },
  "AliCloudCertificateId": {
    "Type": "String",
    "Description": "The ID of the Alibaba Cloud certificate."
  }
},
"Resources": {
  "Certificate": {
    "Type": "ALIYUN::SLB::Certificate",
    "Properties": {
      "CertificateType": {
        "Ref": "CertificateType"
      },
      "AliCloudCertificateName": {
        "Ref": "AliCloudCertificateName"
      },
      "PrivateKey": {
        "Ref": "PrivateKey"
      },
      "CertificateName": {
        "Ref": "CertificateName"
      }
    }
  }
}

```

```

    },
    "Certificate": {
      "Ref": "Certificate"
    },
    "AliCloudCertificateId": {
      "Ref": "AliCloudCertificateId"
    }
  }
}
},
"Outputs": {
  "Fingerprint": {
    "Description": "The fingerprint of the certificate.",
    "Value": {
      "Fn::GetAtt": [
        "Certificate",
        "Fingerprint"
      ]
    }
  },
  "CertificateId": {
    "Description": "The ID of the certificate.",
    "Value": {
      "Fn::GetAtt": [
        "Certificate",
        "CertificateId"
      ]
    }
  }
}
}
}
}

```

5.5.6.5. ALIYUN::SLB::DomainExtension

ALIYUN::SLB::DomainExtension is used to create a domain extension for an SLB instance.

Statement

```

{
  "Type": "ALIYUN::SLB::DomainExtension",
  "Properties": {
    "Domain": String,
    "ListenerPort": Integer,
    "ServerCertificateId": String,
    "LoadBalancerId": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
Domain	String	No	No	The custom domain name.	None
ListenerPort	Integer	Yes	No	The frontend port used by the HTTPS listener of the SLB instance.	Valid values: 1 to 65535.
ServerCertificateId	String	No	Yes	The ID of the certificate corresponding to the domain name.	None
LoadBalancerId	String	No	No	The ID of the SLB instance.	None

Response parameters

Fn::GetAtt

- **DomainExtensionId**: the ID of the created domain extension.
- **ListenerPort**: The frontend port used by the SLB instance.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "DomainExtension": {
      "Type": "ALIYUN::SLB::DomainExtension",
      "Properties": {
        "Domain": "*.example1.com",
        "ListenerPort": "443",
        "ServerCertificateId": "123157908552****_166f8204689_1714763408_70998****",
        "LoadBalancerId": "lb-bp1o94dp5i6earr9g****"
      }
    }
  },
  "Outputs": {
    "DomainExtensionId": {
      "Value": {
        "Fn::GetAtt": [
          "DomainExtension",
          "DomainExtensionId"
        ]
      }
    },
    "ListenerPort": {
      "Value": {
        "Fn::GetAtt": [
          "DomainExtension",
          "ListenerPort"
        ]
      }
    }
  }
}
```

5.5.6.6. ALIYUN::SLB::Listener

ALIYUN::SLB::Listener is used to create a listener for an SLB instance.

Statement

```

{
  "Type": "ALIYUN::SLB::Listener",
  "Properties": {
    "MasterSlaveServerGroupId": String,
    "AclStatus": String,
    "Protocol": String,
    "AclId": String,
    "ServerCertificateId": String,
    "HealthCheck": Map,
    "RequestTimeout": Integer,
    "IdleTimeout": Integer,
    "ListenerPort": Integer,
    "HttpConfig": Map,
    "Bandwidth": Integer,
    "AclType": String,
    "BackendServerPort": Integer,
    "Scheduler": String,
    "LoadBalancerId": String,
    "CACertificateId": String,
    "Persistence": Map,
    "VServerGroupId": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
MasterSlaveServerGroupId	String	Yes	Released	The ID of the active/standby server group.	None
AclStatus	String	Yes	Released	Specifies whether to enable access control on the listener.	Valid values: <ul style="list-style-type: none"> on off Default value: off.
AclId	String	Yes	Released	The ID of the access control list (ACL) to which the listener is bound. This parameter is required when the AclStatus parameter is set to on.	None
				The type of the ACL. Valid values: white	

Parameter	Type	Required	Editable	Description	Constraint
AclType	String	Yes	Released	<p>and black:</p> <ul style="list-style-type: none"> white: specifies the ACL as a whitelist. Only requests from the IP addresses or CIDR blocks specified in the ACL are forwarded. Whitelists are applicable to scenarios where you want an application to only be accessed from specific IP addresses. Configuring a whitelist poses risks to your services. After a whitelist is configured, only the IP addresses specified in the whitelist are able to access the SLB listener. If a whitelist is enabled without any IP addresses specified, the SLB listener will not forward any requests. black: specifies the ACL as a blacklist. Requests from the IP addresses or CIDR blocks specified in the ACL are not forwarded. Blacklists are applicable to scenarios where you want an application to only be 	<p>Valid values:</p> <ul style="list-style-type: none"> White Black

Parameter	Type	Required	Editable	Description	Constraint
				denied access from specific IP addresses. If a blacklist is enabled without any IP addresses specified, the SLB listener will forward all requests. This parameter is required when the AclStatus parameter is set to on.	
Protocol	String	No	No	The Internet protocol over which the listener will forward requests.	Valid values: <ul style="list-style-type: none"> • http • https • tcp • udp
ListenerPort	Integer	Yes	No	The frontend port used by the SLB instance.	Valid values: 1 to 65535.

Parameter	Type	Required	Editable	Description	Constraint
Bandwidth	Integer	Yes	No	The peak bandwidth of the listener. Unit: Mbit/s.	<ul style="list-style-type: none"> Valid values:- 1 and 1 to 1000. For an SLB instance that is connected to the Internet and billed by fixed bandwidth, this parameter cannot be set to -1, and the sum of peak bandwidth values assigned to different listeners cannot exceed the Bandwidth value specified when the SLB instance is created. For an SLB instance that is connected to the Internet and billed by traffic, this parameter can be set to -1. Unit: Mbit/s.
BackendServerPort	Integer	Yes	No	The backend port used by the SLB instance.	Valid values: 1 to 65535.
LoadBalancerId	String	No	No	The ID of the SLB instance.	None
HealthCheck	Map	Erased	Released	The health check settings of the listener.	None
Persistence	Map	Erased	Released	The persistence properties.	None

Parameter	Type	Required	Editable	Description	Constraint
Scheduler	String	Yes	Released	The algorithm used to direct traffic to individual servers.	Valid values: <ul style="list-style-type: none"> wrr wlc Default value: wrr
CACertificateId	String	Yes	Released	The ID of the CA certificate.	Only valid for HTTPS
ServerCertificateId	String	Yes	Released	The ID of the server certificate.	This parameter is required and valid only for HTTPS listeners.
VServerGroupId	String	Yes	Released	The ID of the VServer group.	None
RequestTimeout	String	Optional	Released	The request timeout period. Unit: seconds.	Valid values: 1 to 180.
IdleTimeout	String	Optional	Released	The idle connection timeout period. Unit: seconds.	Valid values: 1 to 60.
HttpConfig	Map	Erased	Released	The HTTP configurations.	None

HealthCheck syntax

```
"HealthCheck": {
  "Domain": String,
  "Interval": Integer,
  "URI": String,
  "HttpCode": String,
  "HealthyThreshold": Integer,
  "Timeout": Integer,
  "UnhealthyThreshold": Integer,
  "Port": Integer
}
```

HealthCheck properties

Parameter	Type	Required	Editable	Description	Constraint
-----------	------	----------	----------	-------------	------------

Parameter	Type	Required	Editable	Description	Constraint
Domain	String	Yes	Released	The domain name used for health checks.	<ul style="list-style-type: none"> The value can be <code>\$_ip</code>, a custom string, or an empty string. A custom string must be 1 to 80 characters in length and can contain only letters, digits, hyphens (-), and periods (.). When this parameter is set to <code>\$_ip</code> or left empty, the SLB instance uses the private IP addresses of backend servers as the domain names for health checks.
Interval	String	Optional	Released	The time interval between consecutive health checks. Unit: seconds.	Valid values: 1 to 5. Unit: seconds.
URI	String	Yes	Released	The URI used for health checks.	<ul style="list-style-type: none"> The URI must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), percent signs (%), question marks (?), number signs (#), and ampersands (&). It must start with a forward slash (/).

Parameter	Type	Required	Editable	Description	Constraint
HttpCode	String	Yes	Released	The HTTP status code that indicates a positive health status of the backend servers.	<ul style="list-style-type: none"> Valid values: http_2xx, http_3xx, http_4xx, and http_5xx. Separate multiple HTTP status codes with commas (,). Default value: http_2xx
HealthyThreshold	String	Optional	Released	The threshold used to determine that the backend servers are healthy. This value indicates the number of consecutive successful health checks required before the health status of a backend server can be changed from fail to success.	Valid values: 1 to 10.
Timeout	String	Optional	Released	The length of time to wait for the response from a health check. Unit: seconds.	Valid values: 1 to 50. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;">  Notice This parameter is valid only when its value is greater than or equal to that of the Interval parameter. Otherwise, this parameter will be overridden by the Interval value. </div>

Parameter	Type	Required	Editable	Description	Constraint
UnhealthyThreshhold	String	Optional	Released	The threshold used to determine that the backend servers are unhealthy. This value indicates the number of consecutive failed health checks required before the health status of a backend server can be changed from success to fail.	Valid values: 1 to 10.
Port	String	Optional	Released	The port used for health checks.	Valid values: 0 to 65535.

Persistence syntax

```
"Persistence": {
  "PersistenceTimeout": Integer,
  "CookieTimeout": Integer,
  "XForwardedFor": String,
  "Cookie": String,
  "StickySession": String,
  "StickySessionType": String
}
```

Persistence properties

Parameter	Type	Required	Editable	Description	Constraint
StickySession	String	No	No	Specifies whether to enable session persistence.	Valid values: <ul style="list-style-type: none"> on off
PersistenceTimeout	String	Optional	Released	The maximum duration that the client can be connected to the server. Unit: seconds.	Valid values: 0 to 1000. The default value is 0, which indicates that connection persistence is disabled. Unit: seconds.

Parameter	Type	Required	Editable	Description	Constraint
CookieTimeout	String	Optional	Released	The maximum duration the cookie can be retained before it expires. Unit: seconds.	<p>This parameter is required when the StickySession parameter is set to on and the StickySessionType parameter is set to insert.</p> <p>Valid values: 1 to 86400.</p> <p>Unit: seconds.</p>
XForwardedFor	String	Yes	Released	Specifies whether to use the X-Forwarded-For header field to obtain the real IP address of the client.	<p>Valid values:</p> <ul style="list-style-type: none"> • on • off <p>Default value: on</p>
Cookie	String	Yes	Released	The cookie configured on the backend server.	<ul style="list-style-type: none"> • The parameter value must be 1 to 200 characters in length and follow the RFC 2965 standard. It can contain only ASCII characters. It cannot contain commas (,), semicolons (;), or spaces, and cannot start with a dollar sign (\$). • This parameter is required when the StickySession parameter is set to on and the StickySessionType parameter is set to server.

Parameter	Type	Required	Editable	Description	Constraint
StickySessionType	String	Yes	Released	The method for processing cookies.	<ul style="list-style-type: none"> Valid values: insert and server. When this parameter is set to insert, SLB adds a cookie to the first response from the backend server. Then, the next request contains the cookie and the listener distributes the request to the same backend server. When this parameter is set to server, SLB overwrites the original cookie when a new cookie is set. The next time the client carries the new cookie to access SLB, the listener distributes the request to the previously recorded backend server. This parameter is required when the StickySession parameter is set to on. This parameter is ignored when the StickySession parameter is set to off.

HttpConfig syntax

```
"HttpConfig": {
  "ForwardPort": Integer,
  "ListenerForward": String
}
```

HttpConfig properties

Parameter	Type	Required	Editable	Description	Constraint
ForwardPort	String	Optional	Released	The port used to redirect HTTP requests to HTTPS.	Valid values: 1 to 65535. Default value: 443.
ListenerForward	String	No	No	Specifies whether to enable HTTP-to-HTTPS redirection.	Valid values: • on • off Default value: off.

Response parameters

Fn::GetAtt

- LoadBalancerId: the unique ID of the SLB instance.
- ListenerPortsAndProtocol: an array consisting of the ports and protocols used by the SLB listener.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "LoadBalancer": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "createdByHeat",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth"
      }
    },
    "CreateListener": {
      "Type": "ALIYUN::SLB::Listener",
      "Properties": {
        "LoadBalancerId": {"Ref": "LoadBalancer"},
        "ListenerPort": "8094",
        "BackendServerPort": 8080,
        "Bandwidth": 1,
        "Protocol": "http",
        "HealthCheck": {
          "HealthyThreshold": 3,
          "UnhealthyThreshold": 3,
          "Interval": 2,
          "Timeout": 5,
          "HttpCode": "http_2xx,http_3xx,http_4xx,http_5xx"
        },
        "Scheduler": "wrr",
        "Persistence": {
          "PersistenceTimeout": 1,
          "XForwardedFor": "on",
          "StickySession": "on",
          "StickySessionType": "insert",
          "CookieTimeout": 10,
          "Cookie": "1"
        }
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
      "Value": {"Fn::GetAtt": ["LoadBalancer", "LoadBalancerId"]}
    }
  }
}
```

5.5.6.7. ALIYUN::SLB::LoadBalancer

ALIYUN::SLB::LoadBalancer is used to create an SLB instance.

Statement

```
{
  "Type": "ALIYUN::SLB::LoadBalancer",
  "Properties": {
    "DeletionProtection": Boolean,
    "AddressType": String,
    "Tags": List,
    "InternetChargeType": String,
    "Bandwidth": Integer,
    "SlaveZoneId": String,
    "ResourceGroupId": String,
    "AutoPay": Boolean,
    "VpcId": String,
    "PricingCycle": String,
    "LoadBalancerName": String,
    "Duration": Number,
    "VSwitchId": String,
    "LoadBalancerSpec": String,
    "MasterZoneId": String,
    "PayType": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the RDS instance belongs.	None
DeletionProtection	Boolean	Erased	Released	Specifies whether to enable deletion protection to prevent the SLB instance from being deleted by mistake.	Valid values: <ul style="list-style-type: none"> • true • false
VpcId	String	Yes	Released	The ID of the VPC.	None

Parameter	Type	Required	Editable	Description	Constraint
SlaveZoneId	String	Yes	Released	The ID of the secondary zone to which the SLB instance belongs.	None
Bandwidth	String	Optional	Released	The peak bandwidth of SLB instances that are connected to the Internet and billed by fixed bandwidth.	For SLB instances that are connected to the Internet and billed by fixed Bandwidth, this parameter is valid only when the Bandwidth parameter of the SLB Listener is specified. For Internet instances whose billing type is to pay by traffic, we recommend that you set the peak Bandwidth through the Listener parameter. In this case, this parameter is ignored. Valid values: 1 to 1000. Unit: Mbps. Default value: 1 VPC-type instances are billed by traffic.
AddressType	String	Yes	Released	The address type of the SLB instance.	Valid values: <ul style="list-style-type: none"> internet intranet Default value: internet.
VSwitchId	String	Yes	Released	The ID of the vSwitch in the VPC.	None

Parameter	Type	Required	Editable	Description	Constraint
LoadBalancerName	String	Yes	Released	The name of the SLB instance to be created.	A custom string. The name must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), and underscores (_). If this parameter is not specified, the system assigns a value.
InternetChargeType	String	Yes	Released	The billing method of SLB instances that are connected to the Internet.	Valid values: <ul style="list-style-type: none"> • paybybandwidth • paybytraffic Default value: paybytraffic.
MasterZoneId	String	Yes	Released	The ID of the primary zone to which the SLB instance belongs.	None
Tags	List	Erased	Released	The tags to be attached to the SLB instance. The tags are listed in JSON format. Each tag consists of a TagKey and a TagValue.	A maximum of five tags can be attached to an SLB instance.
LoadBalancerSpec	String	Yes	Released	The type of the SLB instance.	Valid values: <ul style="list-style-type: none"> • slb.s1.small • slb.s2.small • slb.s2.medium • slb.s3.small • slb.s3.medium • slb.s3.large The available types vary by region.

Parameter	Type	Required	Editable	Description	Constraint
AutoPay	Boolean	Erased	Released	Specifies whether to automatically pay for subscription SLB instances that are connected to the Internet.	<p>Valid values:</p> <ul style="list-style-type: none"> • true • false <p>Default value: false.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note This parameter is valid only on the China site (aliyun.com)</p> </div>
PayType	String	Yes	Released	The billing method of the SLB instance.	<p>Valid values:</p> <ul style="list-style-type: none"> • PayOnDemand • PrePay
PricingCycle	String	Yes	Released	The billing cycle of subscription SLB instances that are connected to the Internet.	<p>Valid values:</p> <ul style="list-style-type: none"> • month • year <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note This parameter is valid only on the China site (aliyun.com)</p> </div>

Parameter	Type	Required	Editable	Description	Constraint
Duration	Number	Erased	Released	The subscription period of subscription SLB instances that are connected to the Internet.	Valid values: <ul style="list-style-type: none"> Valid values when the PricingCycle parameter is set to month: 1 to 9. Valid values when the PricingCycle parameter is set to year: 1 to 3. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note This parameter is valid only on the China site (aliyun.com).</p> </div>

Tags syntax

```
"Tags": [
  {
    "Value": String ,
    "Key": String
  }
]
```

Tags properties

Property	Type	Required or Not	Editable	Description	Constraint
Key	String	No	No	None	None
Value	String	Yes	Released	None	Valid values: 1025 to 10000. You cannot use the following commonly used port numbers: 2222, 4500, 4510, 4560, 7505, 9000, 9001, and 9002.

Response parameters

Fn::GetAtt

- **LoadBalancerId**: the unique ID of the SLB instance.
- **NetworkType**: the network type of the SLB instance, which can be vpc or classic.
- **AddressType**: the address type of the SLB instance, which can be intranet or internet.

- `IpAddress`: the IP address of the SLB instance.
- `OrderId`: the ID of the order.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "CreateLoadBalance": {
      "Type": "ALIYUN::SLB::LoadBalancer",
      "Properties": {
        "LoadBalancerName": "createdByHeat",
        "AddressType": "internet",
        "InternetChargeType": "paybybandwidth",
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
      "Value": {
        "Fn::GetAtt": ["CreateLoadBalance", "LoadBalancerId"]
      }
    }
  }
}
```

5.5.6.8. ALIYUN::SLB::LoadBalancerClone

`ALIYUN::SLB::LoadBalancerClone` is used to clone an SLB instance.

Syntax

```
{
  "Type": "ALIYUN::SLB::LoadBalancerClone",
  "Properties": {
    "Tags": List,
    "ResourceGroupId": String,
    "VSwitchId": String,
    "LoadBalancerName": String,
    "SourceLoadBalancerId": String,
    "TagsPolicy": String,
    "BackendServersPolicy": String,
    "BackendServers": List
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
ResourceGroupI d	String	No	No	The ID of the resource group.	None
VSwitchId	String	No	No	The ID of the VSwitch.	The VSwitch must exist in the VPC to which the source SLB instance belongs. If the parameter is not specified, the VSwitch to which the source SLB instance belongs is used.
SourceLoadBala ncerId	String	Yes	No	The ID of the SLB instance to be cloned.	None

Property	Type	Required	Editable	Description	Constraint
BackendServers Policy	String	No	No	The clone policy. The ECS instances listened by the new SLB instance and the weight of each ECS instance are specified in the policy.	<p>Valid values:</p> <ul style="list-style-type: none"> clone: The ECS instances listened by the source SLB instance and the ECS instance weights are cloned to the new SLB instance. empty: No ECS instances are attached to the new SLB instance. append: The ECS instances listened by the source SLB instance and the ECS instance weights are cloned to the new SLB instance. New ECS instances with specified weights are also attached to the new SLB instance. replace: New ECS instances with specified weights are attached to the new SLB instance. However, the ECS instances listened by the source SLB instance and the ECS instance weights are not cloned to the new SLB instance. <p>Default value: clone.</p>
BackendServers	List	No	Yes	The new ECS instances to be listened.	None

Property	Type	Required	Editable	Description	Constraint
LoadBalancerName	String	No	No	The name of the new SLB instance.	You can set the name to any string. The name must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), and underscores (_).
Tags	List	No	Yes	The tags of the SLB instance.	Tags must be specified as key-value pairs. A maximum of five tags can be specified.
TagsPolicy	String	No	No	The policy of the tags.	<p>Valid values:</p> <ul style="list-style-type: none"> clone: The tags of the source SLB instance are used. empty: No tags are configured. append: The tags of the source SLB instance are reserved while new tags are added. replace: The tags of the source SLB instance are deleted while new tags are added. <p>Default value: empty.</p>

BackendServers syntax

```
"BackendServers": [
  {
    "ServerId": String,
    "Weight": Integer
  }
]
```

BackendServers properties

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	Yes	The ID of the ECS instance.	The ECS instances must be in the running state.
Weight	Integer	Yes	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100. Default value: 100.

Response parameters

`Fn::GetAtt`

`LoadBalancerId`: the ID of the new SLB instance.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "CloneLoadBalance": {
      "Type": "ALIYUN::SLB::LoadBalancerClone",
      "Properties": {
        "SourceLoadBalancerId": "150ebed5f06-cn-beijing-btc-***",
        "LoadBalancerName": "rosnew",
        "BackendServersPolicy": "replace",
        "BackendServers": [
          {
            "ServerId": "i-25zsk****",
            "Weight": 20
          }
        ]
      }
    }
  },
  "Outputs": {
    "LoadBalanceDetails": {
      "Value": {"Fn::GetAtt": ["CloneLoadBalance", "LoadBalancerId"]}
    }
  }
}
```

5.5.6.9. ALIYUN::SLB::MasterSlaveServerGroup

`ALIYUN::SLB::MasterSlaveServerGroup` is used to create a primary/secondary server group.

 **Notice** A primary/secondary server group contains only two ECS instances: a primary server and a secondary server.

Syntax

```
{
  "Type": "ALIYUN::SLB::MasterSlaveServerGroup",
  "Properties": {
    "MasterSlaveServerGroupName": String,
    "MasterSlaveBackendServers": List,
    "LoadBalancerId": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
MasterSlaveServerGroupName	String	No	No	The name of the primary/secondary server group.	None
MasterSlaveBackendServers	List	Yes	No	The list of backend servers in the primary/secondary server group.	A primary/secondary server group can contain a maximum of two backend servers. If you do not specify this parameter, an empty primary/secondary server group is created.
LoadBalancerId	String	Yes	No	The ID of the SLB instance.	None

MasterSlaveBackendServers syntax

```
"MasterSlaveBackendServers": [
  {
    "ServerId": String,
    "Port": Integer,
    "Weight": Integer,
    "ServerType": String
  }
]
```

MasterSlaveBackendServers properties

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	No	The ID of the ECS instance or Elastic Network Interface (ENI) to be added.	None
ServerType	String	No	No	The type of the server.	Default value: Master. Valid values: <ul style="list-style-type: none"> • Master • Slave
Port	Integer	Yes	No	The port number used by the backend server.	Valid values: 1 to 65535.
Weight	Integer	Yes	No	The weight of the backend server.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

MasterSlaveServerGroupId: the ID of the primary/secondary server group.

Examples

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "MasterSlaveServerGroup": {
      "Type": "ALIYUN::SLB::MasterSlaveServerGroup",
      "Properties": {
        "MasterSlaveServerGroupName": "Group1",
        "MasterSlaveBackendServers": [
          {
            "ServerId": "vm****",
            "Port": "80",
            "Weight": "100",
            "ServerType": "Master"
          },
          {
            "ServerId": "vm****",
            "Port": "90",
            "Weight": "100",
            "ServerType": "Slave"
          }
        ],
        "LoadBalancerId": "lb-bp1hv944r69al4j9j****"
      }
    }
  },
  "Outputs": {
    "MasterSlaveServerGroupId": {
      "Value": {
        "Fn::GetAtt": [
          "MasterSlaveServerGroup",
          "MasterSlaveServerGroupId"
        ]
      }
    }
  }
}

```

5.5.6.10. ALIYUN::SLB::Rule

ALIYUN::SLB::Rule is used to add forwarding rules for a specified HTTP or HTTPS listener.

Statement

```
{
  "Type": "ALIYUN::SLB::Rule",
  "Properties": {
    "ListenerPort": Integer,
    "RuleList": List,
    "LoadBalancerId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ListenerPort	Integer	Yes	No	The frontend listener port used by the SLB instance.	Valid values: 1 to 65,535.
RuleList	List	Yes	No	The list of forwarding rules to be added.	<p>A maximum of 10 forwarding rules can be added at a time.</p> <p>Each forwarding rule contains the following parameters: RuleName, Domain, Url, and VServerGroupId.</p> <p>You must specify at least one of the following parameters: Domain and URL.</p> <p>The combination of Domain and URL must be unique in a listener.</p>
LoadBalancerId	String	No	No	The IDs of SLB instances.	None

RuleList syntax

```
"RuleList": [
  {
    "Url": String,
    "Domain": String,
    "VServerGroupId": String,
    "RuleName": String
  }
]
```

RuleList properties

Parameter	Type	Required	Editable	Description	Constraint
Url	String	Yes	Released	The request URL.	The name must be 1 to 80 characters in length and can contain letters, numbers, and special characters. -/., percent signs (%), question marks (?), #& .
Domain	String	Yes	Released	The request domain name associated with the forwarding rule.	None
VServerGroupId	String	No	No	The ID of the destination VServer group specified in the forwarding rule.	None
RuleName	String	No	No	The name of the forwarding rule.	The name must be 1 to 40 characters in length and can contain letters, numbers, and special characters. -/., _ . Forwarding rule names must be unique within each listener.

Response parameters

Fn::GetAtt

Rules: the list of forwarding rules.

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Rule": {
      "Type": "ALIYUN::SLB::Rule",
      "Properties": {
        "ListenerPort": {
          "Ref": "ListenerPort"
        },
        "RuleList": {
          "Fn::Split": [",", {
            "Ref": "RuleList"
          }], {
            "Ref": "RuleList"
          }
        ]
      },
      "LoadBalancerId": {
        "Ref": "LoadBalancerId"
      }
    }
  },
  "Parameters": {
    "ListenerPort": {
      "Type": "Number",
      "Description": "The front-end HTTPS listener port of the Server Load Balancer instance. Valid value:\n1-65535",
      "MaxValue": 65535,
      "MinValue": 1
    },
    "RuleList": {
      "MinLength": 1,
      "Type": "CommaDelimitedList",
      "Description": "The forwarding rules to add.",
      "MaxLength": 10
    },
    "LoadBalancerId": {
      "Type": "String",
      "Description": "The ID of Server Load Balancer instance."
    }
  },
  "Outputs": {
    "Rules": {
      "Description": "A list of forwarding rules. Each element of rules contains \"RuleId\".",
      "Value": {
        "Fn::GetAtt": ["Rule", "Rules"]
      }
    }
  }
}

```

```

}
}
}
}

```

5.5.6.11. ALIYUN::SLB::VServerGroup

ALIYUN::SLB::VServerGroup is used to create a VServer group and adds backend servers to the SLB instance.

Syntax

```

{
  "Type" : "ALIYUN::SLB::VServerGroup",
  "Properties" : {
    "VServerGroupName" : String,
    "BackendServers" : List,
    "LoadBalancerId" : String
  }
}

```

Properties

Property	Type	Required	Editable	Description	Constraint
VServerGroupName	String	Yes	No	The name of the VServer group.	None
BackendServers	List	Yes	Yes	The list of ECS instances to be added.	A list can contain up to 20 instances.
LoadBalancerId	String	Yes	No	The ID of the SLB instance.	None

BackendServers syntax

```

"BackendServers" : [
  {
    "ServerId" : String,
    "Port" : Integer,
    "Weight" : Integer
  }
]

```

BackendServers properties

Property	Type	Required	Editable	Description	Constraint
----------	------	----------	----------	-------------	------------

Property	Type	Required	Editable	Description	Constraint
ServerId	String	Yes	Yes	The ID of the ECS instance.	None
Port	Integer	Yes	Yes	The backend port used by the SLB instance.	Valid values: 1 to 65535.
Weight	Integer	Yes	Yes	The weight of the ECS instance to be attached to the SLB instance.	Valid values: 0 to 100.

Response parameters

Fn::GetAtt

- **VServerGroupId**: the ID of the VServer group.
- **BackendServers**: the list of backend servers added to the SLB instance

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "CreateVServerGroup": {
      "Type": "ALIYUN::SLB::VServerGroup",
      "Properties": {
        "LoadBalancerId": "lb-2zenh4ndwrqg14yt0****",
        "VServerGroupName": "VServerGroup-****",
        "BackendServers": [
          {
            "ServerId": "i-25zsk****",
            "Weight": 20,
            "Port": 8080
          },
          {
            "ServerId": "i-25zsk****",
            "Weight": 100,
            "Port": 8081
          }
        ]
      }
    }
  },
  "Outputs": {
    "VServerGroupId": {
      "Value": {"Fn::GetAttr": ["CreateVServerGroup", "VServerGroupId"]}
    }
  }
}
```

5.5.7. VPC

5.5.7.1. ALIYUN::VPC::EIP

ALIYUN::VPC::EIP is used to apply for an Elastic IP address.

Statement

```
{
  "Type": "ALIYUN::VPC::EIP",
  "Properties": {
    "Isp": String,
    "Period": Number,
    "ResourceGroupId": String,
    "AutoPay": Boolean,
    "InstanceChargeType": String,
    "PricingCycle": String,
    "InternetChargeType": String,
    "Bandwidth": Number
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
ResourceGroupId	String	Yes	Released	The ID of the resource group to which the RDS instance belongs.	None
Bandwidth	Number	Erased	Released	The network bandwidth. Unit: Mbit/s.	If this parameter is not specified, the default value 5Mbps is used.
InternetChargeType	String	Yes	Released	The billing method for network usage. Default value: PayByBandwidth.	Valid values: <ul style="list-style-type: none"> PayByBandwidth: pay-by-bandwidth. PayByTraffic Default value: PayByBandwidth.
InstanceChargeType	String	Yes	Released	The billing method of the Elastic IP address. Default value: Postpaid.	Valid values: <ul style="list-style-type: none"> Prepaid Postpaid: pay-as-you-go Default value: PostPaid.
PricingCycle	String	Yes	Released	The billing cycle of the subscription. Default value: Month.	Valid values: <ul style="list-style-type: none"> Month: paid by month. Year Default value: Month. <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <p> Note This parameter is required when InstanceChargeType is set to Prepaid.</p> </div>

Parameter	Type	Required	Editable	Description	Constraint
Period	Number	Erased	Released	The subscription period.	<p>Valid values:</p> <ul style="list-style-type: none"> If pay by month is selected, the billing method can be a fee of 1 to 9. If pay-as-you-go is selected, the payment can be in the range of 1 to 3. <p>Default value: 1</p> <p> Note This parameter is required when InstanceChargeType is set to Prepaid.</p>
AutoPay	Boolean	Erased	Released	Specifies whether to enable automatic payment.	<p>Valid values:</p> <ul style="list-style-type: none"> false: Automatic payment is disabled. After an order is generated, you must go to the Order Center to make the payment. true: Automatic payment is enabled. Payments are automatically made. <p> Note This parameter is required when InstanceChargeType is set to Prepaid.</p>
Isp	String	Yes	Released	The ISP tag used for Finance Cloud. This parameter takes effect only when your region is set to China (Hangzhou).	This parameter is ignored if you are not a Finance Cloud user.

Response parameters

Fn::GetAtt

- **EipAddress**: the allocated Elastic IP address.
- **AllocationId**: the ID of the instance that the Elastic IP address is allocated to.
- **OrderId**: The order ID that is returned when you set the InstanceChargeType parameter to Prepaid.

Sample request

```

{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Eip": {
      "Type": "ALIYUN::VPC::EIP",
      "Properties": {
        "InternetChargeType": "PayByTraffic",
        "Bandwidth": 200
      }
    }
  },
  "Outputs": {
    "EipAddress": {
      "Value": {"Fn::GetAtt": ["Eip", "EipAddress"]}
    },
    "AllocationId": {
      "Value": {"Fn::GetAtt": ["Eip", "AllocationId"]}
    },
    "OrderId": {
      "Value": {"Fn::GetAtt": ["Eip", "OrderId"]}
    }
  }
}

```

5.5.7.2. ALIYUN::VPC::EIPAssociation

ALIYUN::VPC::EIPAssociation is used to associate an Elastic IP address with a cloud service instance.

Statement

```

{
  "Type": "ALIYUN::VPC::EIPAssociation",
  "Properties": {
    "AllocationId": String,
    "InstanceId": String,
    "PrivateIpAddress": String,
    "Mode": String
  }
}

```

Properties

Parameter	Type	Required	Editable	Description	Constraint
AllocationId	String	No	Yes	The ID of the Elastic IP address.	None

Parameter	Type	Required	Editable	Description	Constraint
InstanceId	String	No	Yes	The ID of the cloud service instance.	The following instance types are supported: <ul style="list-style-type: none"> VPC-connected ECS instances VPC-connected SLB instances NAT gateways HAVIP Elastic network interfaces
PrivateIpAddresses	String	Yes	True	The private IP address in the CIDR block of the VSwitch.	None
Mode	String	Yes	True	The association mode.	Valid values: <ul style="list-style-type: none"> NAT MULTI_BINDED

Response parameters

Fn::GetAtt

- EipAddress: The allocated Elastic IP address.
- AllocationId: The ID of the instance to which the Elastic IP address is allocated.

Examples

JSON format

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "Eip": {
      "Type": "ALIYUN::VPC::EIP",
      "Properties": {
        "InternetChargeType": "PayByTraffic",
        "Bandwidth": 200
      }
    },
    "EipAssociation": {
      "Type": "ALIYUN::VPC::EIPAssociation",
      "Properties": {
        "InstanceId": "<LoadBalancerId>",
        "InstanceType": "EcsInstance",
        "AllocationId": {
          "Fn::GetAtt": ["Eip", "AllocationId"]
        }
      }
    }
  },
  "Outputs": {
    "EipAddress": {
      "Value": {"Fn::GetAtt": ["EipAssociation", "EipAddress"]}
    },
    "AllocationId": {
      "Value": {"Fn::GetAtt": ["EipAssociation", "AllocationId"]}
    }
  }
}
```

YAML format

```
ROSTemplateFormatVersion: '2015-09-01'
```

```
Resources:
```

```
Eip:
```

```
  Type: ALIYUN::VPC::EIP
```

```
  Properties:
```

```
    InternetChargeType: PayByTraffic
```

```
    Bandwidth: 200
```

```
EipAssociation:
```

```
  Type: ALIYUN::VPC::EIPAssociation
```

```
  Properties:
```

```
    InstanceId: "<LoadBalancerId>"
```

```
    InstanceType: EcsInstance
```

```
  AllocationId:
```

```
    Fn::GetAtt:
```

```
      - Eip
```

```
      - AllocationId
```

```
Outputs:
```

```
EipAddress:
```

```
  Value:
```

```
    Fn::GetAtt:
```

```
      - EipAssociation
```

```
      - EipAddress
```

```
AllocationId:
```

```
  Value:
```

```
    Fn::GetAtt:
```

```
      - EipAssociation
```

```
      - AllocationId
```

5.5.7.3. ALIYUN::VPC::PeeringRouterInterfaceBinding

ALIYUN::VPC::PeeringRouterInterfaceBinding is used to associate two router interfaces to be interconnected.

Statement

```
{
  "Type": "ALIYUN::VPC::PeeringRouterInterfaceBinding",
  "Properties": {
    "OppositeRouterId": String,
    "OppositeInterfaceId": String,
    "OppositeInterfaceOwnerId": String,
    "RouterInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
RouterInterfaceId	String	No	No	The ID of the router interface.	None
OppositeInterfaceId	String	No	No	The ID of the peer router interface.	None
OppositeRouterId	String	Yes	Released	The ID of the router to which the peer router interface belongs.	None
OppositeInterfaceOwnerId	String	Yes	Released	The ID of the owner of the peer router interface.	None

Response parameters

Fn::GetAtt

RouterInterfaceId: the ID of the vRouter.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "InitiatorRouterInterfaceBinding": {
      "Type": "ALIYUN::VPC::PeeringRouterInterfaceBinding",
      "Properties": {
        "RouterInterfaceId": "ri-2zedgo0ih64g1me29****",
        "OppositeInterfaceId": "ri-2zex1tkyym98pjaor****",
        "OppositeRouterId": "vrt-2zexb35tzorIU0286****"
      }
    }
  }
}
```

5.5.7.4. ALIYUN::VPC::PeeringRouterInterfaceConnection

ALIYUN::VPC::PeeringRouterInterfaceConnection is used to initiate a router interface connection.

Statement

```
{
  "Type": "ALIYUN::VPC::PeeringRouterInterfaceConnection",
  "Properties": {
    "OppositeInterfaceId": String,
    "RouterInterfaceId": String
  }
}
```

Properties

Parameter	Type	Required	Editable	Description	Constraint
OppositeInterfaceId	String	No	No	The ID of the acceptor router interface.	None
RouterInterfaceId	String	No	No	The ID of the router interface to initiate the connection.	None

Response parameters

Fn::GetAtt

- **OppositeInterfaceId**: the ID of the acceptor router interface.
- **RouterInterfaceId**: the ID of the router interface that initiates the connection.

Sample request

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "InitiatorRouterInterfaceBinding": {
      "Type": "ALIYUN::VPC::PeeringRouterInterfaceConnection",
      "Properties": {
        "RouterInterfaceId": "ri-2zedgo0ih64g1me29****",
        "OppositeInterfaceId": "ri-2ze4k5n2aeardu8cy****"
      }
    }
  }
}
```

5.5.7.5. ALIYUN::VPC::RouterInterface

ALIYUN::VPC::RouterInterface is used to create a router interface.

Syntax

```
{
  "Type": "ALIYUN::VPC::RouterInterface",
  "Properties": {
    "OppositeRegionId": String,
    "Description": String,
    "HealthCheckSourceIp": String,
    "RouterType": String,
    "AccessPointId": String,
    "RouterId": String,
    "Role": String,
    "OppositeInterfaceOwnerId": String,
    "OppositeAccessPointId": String,
    "HealthCheckTargetIp": String,
    "OppositeRouterId": String,
    "Spec": String,
    "OppositeRouterType": String,
    "Name": String,
    "PricingCycle": String,
    "Period": Number,
    "AutoPay": Boolean,
    "InstanceChargeType": String
  }
}
```

Properties

Property	Type	Required	Editable	Description	Constraint
RouterId	String	Yes	No	The ID of the router	None
Role	String	Yes	No	The role of the router interface.	<ul style="list-style-type: none"> When RouterType is set to VBR, set the value to InitiatingSide. When OppositeRouterType is set to VBR, set the value to AcceptingSide.
RouterType	String	No	No	The type of the router to which the router interface belongs.	Valid values: <ul style="list-style-type: none"> VRouter VBR

Property	Type	Required	Editable	Description	Constraint
AccessPointId	String	No	No	The ID of the access point of the router interface.	<ul style="list-style-type: none"> This parameter is required when RouterType is set to VBR. The access point ID cannot be modified after the router interface is created. This parameter is not required when RouterType is set to VRouter.
Spec	String	No	No	The specifications of the router interface.	<p>The following list includes available specifications and the corresponding bandwidth values:</p> <ul style="list-style-type: none"> Mini.2: 2 Mbit/s Mini.5: 5 Mbit/s Small.1: 10 Mbit/s Small.2: 20 Mbit/s Small.5: 50 Mbit/s Middle.1: 100 Mbit/s Middle.2: 200 Mbit/s Middle.5: 500 Mbit/s Large.1: 1,000 Mbit/s Large.2: 2,000 Mbit/s Large.5: 5,000 Mbit/s Xlarge.1: 10,000 Mbit/s <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfcfcf;"> <p> Note</p> <ul style="list-style-type: none"> This parameter is required when Role is set to InitiatingSide. The value Negative is used by default when Role is set to AcceptingSide. </div>

Property	Type	Required	Editable	Description	Constraint
OppositeRegionId	String	No	No	The region ID of the peer router interface.	None
OppositeInterfaceOwnerId	String	No	No	The ID of the owner of the peer router interface.	The default value is the ID of the current user.
OppositeRouterId	String	No	No	The ID of the router to which the peer router interface belongs.	None
OppositeRouterType	String	No	No	The type of the router to which the peer router interface belongs.	Valid values: <ul style="list-style-type: none"> When RouterType is set to VBR, set the value to VRouter. VBR
OppositeAccessPointId	String	No	No	The ID of the access point of the peer router interface.	<ul style="list-style-type: none"> When OppositeRouterType is set to VBR, this parameter is required. The access point ID cannot be modified after the router interface is created. When OppositeRouterType is set to VRouter, this parameter is not required. When OppositeRouterType is set to VBR, the VBR specified by the OppositeRouterId parameter must be in the access point specified by the OppositeAccessPointId parameter.
Description	String	No	No	The description of the router interface.	<p>The description must be 2 to 256 characters in length. It cannot start with <code>http://</code> or <code>https://</code>.</p> <p>The parameter is empty by default.</p>

Property	Type	Required	Editable	Description	Constraint
Name	String	No	No	The display name of the router interface.	<ul style="list-style-type: none"> The name must be 2 to 128 characters in length and can contain letters, digits, periods(.), underscores (_), and hyphens (-). It must start with a letter but cannot start with <code>http://</code> or <code>https://</code>.
HealthCheckSourceIp	String	No	No	The source IP address of health check packets used in leased line disaster recovery and ECMP scenarios.	<p>This parameter is valid only for VRouter interfaces with a peer router interface on a VBR.</p> <p>It must be an unused IP address in the VPC where the local VRouter is located.</p> <p>The HealthCheckSourceIp and HealthCheckTargetIp parameters must either both be specified or both left unspecified.</p>

Property	Type	Required	Editable	Description	Constraint
HealthCheckTargetIp	String	No	No	The destination IP address of health check packets used in leased line disaster recovery and ECMP scenarios.	<p>This parameter is valid only for VRouter interfaces with a peer router interface on a VBR. Typically, you can use the IP address of a customer premises equipment (CPE) on the user side of the leased line, which is the IP address of the peer gateway on the VBR where the peer router interface is located. You can also specify another IP address on the user side of the leased line as the destination IP address.</p> <p>The HealthCheckSourceIp and HealthCheckTargetIp parameters must either both be specified or both left unspecified.</p>
PricingCycle	String	No	No	The billing cycle of the subscription.	<p>Valid values:</p> <ul style="list-style-type: none"> Month Year
Period	Number	No	No	The subscription duration.	<ul style="list-style-type: none"> Valid values when the PricingCycle parameter is set to Month: 1 to 9. Valid values when the PricingCycle parameter is set to Year: 1 to 3.
AutoPay	Boolean	No	No	Specifies whether to enable automatic payment.	<p>Default value: false.</p> <p>Valid values:</p> <ul style="list-style-type: none"> true false
InstanceChargeType	String	No	No	The billing method of the instance.	<p>Valid values:</p> <ul style="list-style-type: none"> Postpaid: pay-as-you-go Prepaid: subscription

Response parameters

Fn::GetAtt

RouterInterfaceId: the ID of the router interface.

Examples

```
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Resources": {
    "RouterInterface": {
      "Type": "ALIYUN::VPC::RouterInterface",
      "Properties": {
        "Name": "RouterInterface_1",
        "Description": "VPC initiator RouterInterface",
        "RouterId": "vrt-2ze2i147e5n0bicoe****",
        "Role": "AcceptingSide",
        "OppositeRegionId": "cn-beijing",
        "HealthCheckSourceIp": "10.0.XX.XX",
        "HealthCheckTargetIp": "192.168.XX.XX"
      }
    }
  },
  "Outputs": {
    "RouterInterfaceId": {
      "Value": {"Fn::GetAtt": ["RouterInterface", "RouterInterfaceId"]}
    }
  }
}
```

6.Object Storage Service (OSS)

6.1. 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.

OSS is an immediately available storage solution that has unlimited storage capacity. 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 texts, 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:

- 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, and configure ACL for the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using SDKs or calling RESTful API operations in your application.

6.2. Usage notes

Before you use OSS, you must understand the following content:

To allow other users to use all or part of OSS features, you must create RAM users and grant permissions to the users by configuring RAM policies.

Before you use OSS, you must also understand the following limits.

Item	Limit
Bucket	<ul style="list-style-type: none"> • You can create up to 100 buckets. • After a bucket is created, its name and region cannot be modified.
Upload objects	<ul style="list-style-type: none"> • Objects larger than 5 GB cannot be uploaded by using the following modes: console upload, simple upload, form upload, or append upload. To upload an object that is larger than 5 GB, you must use multipart upload. The size of an object uploaded by using multipart upload cannot exceed 48.8 TB. • If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.
Delete objects	<ul style="list-style-type: none"> • Deleted objects cannot be recovered. • You can delete up to 100 objects at a time in the OSS console. To delete more than 100 objects at a time, you must call an API operation or use an SDK.
Lifecycle	You can configure up to 1,000 lifecycle rules for each bucket.

6.3. Quick start

6.3.1. Log on to the OSS console

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

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press Enter.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Object Storage Service**.

6.3.2. Create buckets

Objects uploaded to OSS are stored in a bucket. Create a Bucket before you upload any objects to OSS.

Context

The attributes of a bucket include the region, ACL, and other metadata.

Procedure

1. **Log on to the OSS console.**
2. In the left-side navigation pane, click **Create Bucket** if no buckets are available. In the **Create OSS Bucket** dialog box that appears, configure parameters.

 **Note** In the left-side navigation pane, click the + icon next to **Buckets** if buckets exist. The **Create OSS Bucket** dialog box appears.

The following table describes the parameters used to create a bucket.

Parameters

Parameter	Description
Organization	Select an organization from the drop-down list for the bucket.
Resource Set	Select a resource set from the drop-down list for the bucket.

Parameter	Description
Region	<p>Select a region from the drop-down list for the bucket.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> ◦ The region of a bucket cannot be changed after the bucket is created. ◦ If you want to access OSS from your ECS instance over the internal network, select the region where your ECS instance is deployed. </div>
Cluster	Select a cluster for the bucket. Two OSS clusters can be deployed in Apsara Stack.
Bucket Name	<p>Enter the name of the bucket.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> ◦ The bucket name must comply with the naming conventions. ◦ The bucket name must be globally unique among all the existing buckets in Alibaba Cloud OSS. ◦ The bucket name cannot be changed after the bucket is created. </div>
Storage Class	Currently, only Standard buckets are supported.
Capacity	<p>Set the capacity of the bucket:</p> <ul style="list-style-type: none"> ◦ Unlimited: The capacity is unlimited. ◦ Custom: Select this option to set the capacity of the bucket. Valid values: 0 to 2000000. Unit: TB.
Access Control List (ACL)	<p>Set the ACL of the bucket. The following options are available:</p> <ul style="list-style-type: none"> ◦ Private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access objects in the bucket without authorization. ◦ Public Read: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users can only read objects in the bucket. ◦ Public Read/Write: Any users, including anonymous users can read and write objects in the bucket. Fees incurred by such operations are paid by the owner of the bucket. Exercise caution when you configure this option. <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note After a bucket is created, you can modify its ACL. For more information, see Modify bucket ACLs.</p> </div>

Parameter	Description
Server-side Encryption	Configure server-side encryption: <ul style="list-style-type: none"> ◦ No: Server-side encryption is not performed. ◦ AES256: OSS server-side encryption uses AES256 to encrypt each object in the bucket with a different data key. CMKs used to encrypt data keys are rotated regularly. ◦ SM4: OSS server-side encryption uses SM4 to encrypt each object in the bucket with a different data key. CMKs used to encrypt data keys are rotated regularly.

3. Click **Submit**.

6.3.3. Upload objects

After you create a bucket, you can upload objects to it.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

You can upload an object of any format to a bucket. You can use the OSS console to upload an object up to 5 GB in size. To upload an object larger than 5 GB, use an SDK or call an API operation.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket to which you want to upload objects to go to the bucket details page.
3. Click the **Files** tab.
4. Click **Upload**. The **Upload** dialog box appears.
5. In the **Upload To** section, set the directory to which the object will be uploaded.
 - **Current**: Objects are uploaded to the current folder.
 - **Specified**: Objects are uploaded to the specified folder. OSS creates the specified folder automatically and uploads the object to it.

 **Note** For more information about folders, see [Create folders](#).

6. In the **File ACL** section, select the ACL of the object to upload. By default, an object inherits the ACL of the bucket to which it belongs.
7. Drag and drop one or more objects to upload to the **Upload** field, or click **Upload** to select one or more objects to upload.

 **Note**

- If the uploaded object has the same name as an existing object in the bucket, the existing object will be overwritten.
- During object upload, do not refresh or close the page. Otherwise, the upload queue will be interrupted and cleared.
- The name of the uploaded object must comply with the following conventions:
 - The name can contain only UTF-8 characters.
 - The name is case-sensitive.
 - The name must be 1 to 1,023 bytes in length.
 - The name cannot start with a forward slash (/) or backslash (\).

8. After the object is uploaded, refresh the Files tab to view the uploaded object.

6.3.4. Obtain object URLs

You can obtain the URL of an object uploaded to a bucket. This URL can be used to share or download the object.

Prerequisites

- A bucket is created. For more information about how to create a bucket, see [Create buckets](#).
- Objects are uploaded to the bucket. For more information about how to upload objects, see [Upload objects](#).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket that stores the target object to go to the bucket details page.
3. Click the **Files** tab. The list of objects appears.
4. Click the name of the target object. In the **Preview** dialog box that appears, click **Copy File URL** under the URL field. You can also choose **More > Copy File URL** in the Actions column corresponding to the object. In the dialog box that appears, click **Copy**. You can send the URL to other users so that they can view or download the object.

6.4. Buckets

6.4.1. View bucket information

You can view the details of created buckets in the OSS console.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the target bucket.
3. On the bucket details page that appears, click the **Overview** tab. View the bucket domain names and basic settings.

6.4.2. Delete buckets

You can delete buckets in the OSS console.

Prerequisites

All objects and parts stored in the bucket are deleted. For more information about how to delete objects and parts, see [Delete objects](#) and [Manage parts](#).

 **Warning** Deleted objects, parts, and buckets cannot be recovered. Exercise caution when you delete objects, parts, and buckets.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of a bucket to go to the bucket details page.
3. On the bucket details page that appears, click **Delete Bucket** in the upper right corner. In the message that appears, click **OK**.

6.4.3. Modify bucket ACLs

You can modify the access control list (ACL) of a bucket in the OSS console to control access to the bucket.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

OSS provides ACL to control access to buckets. By default, the ACL of a bucket is private when you create the bucket. You can modify the ACL of a bucket after the bucket is created.

OSS provides ACL for buckets. The following ACLs are available for a bucket:

- **Private:** Only the owner or authorized users of the bucket can read and write the object.
- **Public read:** Only the owner or authorized users of this bucket can write the object. Other users, including anonymous users can only read the object.
- **Public read/write:** Any users, including anonymous users can read and write the object. Fees incurred by such operations are paid by the owner of the bucket. Exercise caution when you configure this option.

 **Warning** If you set ACL to public read or public read/write, other users can directly read the data in the bucket without authentication, resulting in security risks. For data security reasons, we recommend that you set the bucket ACL to private.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket for which you want to modify ACL to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Access Control List (ACL)** section.
4. Click **Configure**. Modify the bucket ACL.
5. Click **Save**.

6.4.4. Configure static website hosting

You can configure static website hosting in the OSS console so that users can access the static website by using the bucket domain name.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

Static website hosting is not enabled if the default pages are not specified.

After the default homepage is configured, the default homepage is displayed if you access the root domain name of the static website or any URL that ends with a forward slash (/) under this domain name.

Procedure

1. [Log on to the OSS console.](#)
2. In the left-side navigation pane, click the name of a bucket to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Static Pages** section.
4. Click **Configure**. Configure the following parameters:
 - **Default Homepage:** Specify the name of the index document that links to the index page. The index page functions similar to index.html. Only HTML objects in the root folder can be used. The default homepage is disabled if you do not specify this parameter.
 - **Default 404 Page:** Specify the name of the error document that links to the error page displayed when the requested resource does not exist. Only HTML, JPG, PNG, BMP, or WebP objects in the root folder can be used. Default 404 Page is disabled if you do not specify this parameter.
5. Click **Save**.

6.4.5. Configure hotlink protection

You can configure hotlink protection for a bucket in the OSS console to prevent unauthorized domain names from accessing the data in your bucket.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

OSS provides hotlink protection to prevent other domain names from accessing your data in OSS. You can configure the Referer field in the HTTP header to implement hotlink protection. You can configure a Referer whitelist for a bucket and configure whether to allow access requests that have an empty Referer field in the OSS console. For example, you can add `http://www.aliyun.com` to the Referer whitelist for a bucket named `oss-example`. Then, requests whose Referer field is set to `http://www.aliyun.com` can access the objects in the `oss-example` bucket.

Procedure

1. [Log on to the OSS console.](#)
2. In the left-side navigation pane, click the name of the bucket to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Hotlink Protection** section.
4. Click **Configure**. Configure the following parameters:
 - **Referer Whitelist:** Add URLs to the whitelist. Referers are typically in URL format. Separate multiple Referers with break lines. You can use question marks (?) and asterisks (*) as wildcard characters.
 - **Allow Empty Referer:** Specify whether to allow requests whose Referer field is empty. If you do not allow empty Referers, only HTTP or HTTPS requests which include the corresponding Referer field value can access the objects in the bucket.
5. Click **Save**.

6.4.6. Configure logging

You can enable or disable bucket logging in the OSS console.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

You can store access logs in the current bucket or in a new bucket.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket to go to the bucket details page.
3. On the bucket details page that appears, click the **Basic Settings** tab. Find the **Logging** section.
4. Click **Configure**. Turn on **Logging**. Select **Destination Bucket** and set **Log Prefix**.
 - **Destination Bucket:** Select the name of the bucket in which access logs are to be stored from the drop-down list. You must be the owner of the selected bucket and the bucket must be in the same region as the bucket for which logging is enabled.
 - **Log Prefix:** Enter the prefix and folder where the access logs are stored. If you specify *log/<TargetPrefix>*, the access logs are stored in the *log/* directory.
5. Click **Save**.

6.4.7. Configure CORS

You can configure cross-origin resource sharing (CORS) in the OSS console to enable cross-origin access.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

OSS provides CORS over HTML5 to implement cross-origin access. When OSS receives a cross-origin request (or an OPTIONS request) for a bucket, OSS reads the CORS rules of the bucket and checks the relevant permissions. OSS matches the rules one by one. When OSS finds the first match, OSS returns a corresponding header. If no match is found, OSS does not include any CORS header in the response.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Cross-Origin Resource Sharing (CORS)** section. Click **Configure**.
4. Click **Create Rule**. In the **Create Rule** dialog box that appears, configure the following parameters.

Parameter	Required	Description
Sources	Yes	Specifies the sources from which you want to allow cross-origin requests. You can configure multiple origins and separate them with break lines. Each origin can contain only one asterisk (*) wildcard. If Sources is set to asterisk (*), all cross-origin requests are allowed.
Allowed Methods	Yes	Specifies the cross-origin request methods that are allowed.

Parameter	Required	Description
Allowed Headers	No	Specifies the allowed headers in a cross-origin request. Allowed headers are case-insensitive. You can configure multiple headers and separate them with break lines. Each allowed header can contain only one asterisk (*) wildcard. Set this parameter to an asterisk (*) if there are no special requirements.
Exposed Headers	No	Specifies the list of headers that can be exposed to the browser. The headers are the response headers that allow access from an application such as XMLHttpRequest in JavaScript. No asterisk (*) wildcards are allowed.
Cache Timeout (Seconds)	No	Specifies the time the browser can cache the response to a preflight (OPTIONS) request to a specific resource.

 **Note** You can configure up to 10 rules for each bucket.

5. Click OK.

6.4.8. Manage lifecycle rules

You can define and manage lifecycle rules for a bucket in the OSS console.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

You can define a rule for a full set or a subset by specifying the prefix keyword of objects in a bucket. A rule applies to all objects that match the rule. You can manage lifecycle rules to perform operations, such as object management and automatic part deletion.

Notice

- If an object matches a rule, data of the object is deleted within two days from the effective date.
- Data that is deleted based on a lifecycle rule cannot be recovered. Configure a rule only when necessary.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket to go to the bucket details page.
3. Click the **Basic Settings** tab. Find the **Lifecycle** section. Click **Configure**.
4. Click **Create Rule**. In the **Create Rule** dialog box that appears, configure the following parameters:
 - **Status**: Configure the status of the rule: **Enabled** or **Disabled**.
 - **Applied To**: You can select **Files with Specified Prefix** or **Whole Bucket**. **Files with Specified Prefix** indicates that this rule applies to objects whose names contain a specified prefix. **Whole Bucket** indicates that this rule applies to all objects in the bucket.

 **Note** If you select **Files with Specified Prefix**, you can configure multiple lifecycle rules that have different prefixing configurations for objects. If you select **Whole Bucket**, only one lifecycle rule can be configured. In addition, if you have created a rule that has **Files with Specified Prefix** configured, you cannot create another rule that has **Whole Bucket** configured for the same bucket.

- **Prefix:** If you set **Applied To** to **Files with Specified Prefix**, you must enter the prefix of the objects to which to apply the rule. If you want to match objects whose names start with `img`, enter `img`.
- **File Lifecycle:** Configure operations to perform on expired objects. You can select **Validity Period (Days)**, **Expiration Date**, or **Disabled**.
 - **Validity Period (Days):** Specify the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted. If you set **Validity Period (Days)** to 30, objects that are last modified on January 1, 2016 are scanned for by the backend application and deleted on January 31, 2016.
 - **Expiration Date:** Specify the date before which parts that are last modified expire and the operation to perform on these parts after they expire. If you select **Delete** and set **Expiration Date** to 2012-12-21, the backend application scans for objects that are last modified before December 21, 2012 and delete those objects.
 - **Disabled:** The automatic object deletion function is not enabled.
- **Delete:** If you select **Validity Period (Days)** or **Expiration Date** for **File Lifecycle**, you can select **Delete** to delete objects based on the validity period or expiration time. If you select **Disabled**, the rule becomes invalid.
- **Part Lifecycle:** Configure the delete operation to perform on expired parts. You can select **Validity Period (Days)**, **Expiration Date**, or **Disabled**.
 - **Validity Period (Days):** Specify the number of days within which parts can be retained after they are last modified. After the validity period, expired parts are deleted. If you set **Validity Period (Days)** to 30, the backend application scans for parts that are last modified before January 1, 2016 and deletes them on January 31, 2016.
 - **Expiration Date:** Specify the date before which parts that are last modified expire and the operation to perform on these parts after they expire. If you set **Expiration Date** to 2012-12-21, parts that are last modified before this date are scanned for and deleted by the backend application.
 - **Disabled:** The automatic part deletion function is not enabled.
- **Delete:** If you select **Validity Period (Days)** or **Expiration Date** for **Part Lifecycle**, you can select **Delete** to delete parts based on the validity period or expiration time. If you select **Disabled**, the rule becomes invalid.

 **Notice** In each lifecycle rule, you must configure at least object lifecycle or part lifecycle. In other words, you must select **Delete** or configure conversion actions for object lifecycle or select **Delete** for part lifecycle.

5. Click **OK**.

6.4.9. Configure storage quota

You can specify the storage quota of a bucket in the OSS console to limit the bucket size. You can specify the storage quota of a bucket when you create the bucket. You can also configure the storage quota of a created bucket. This topic describes how to configure the storage quota of a created bucket.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

After a storage quota is configured for a bucket, if the bucket size exceeds the quota, write operations, including PutObject, MultipartUpload, CopyObject, PostObject, AppendObject, cannot be performed on the bucket. Before configuring the quota of a bucket, ensure that it does not affect your business.

 **Notice** In general, OSS takes one hour to determine whether the bucket size exceeds the quota. In some cases, more than one hour is required for OSS to detect that the bucket size exceeds the quota.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket for which you want to configure storage quota to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Storage Quota** section.
4. Click **Configure**. Turn on **Storage Quota** and set the **Storage Quota** value.
 - Units: TB or GB.
 - Valid values: -1 to 2000000

The default value is -1, indicating that the bucket size is not limited.
5. Click **Save**.

6.4.10. Configure cluster-disaster recovery

In cluster-disaster recovery mode, buckets with the same name are replicated. Cluster-based disaster recovery is automatically enabled based on configurations made when the cluster is created. In other words, after a primary bucket is created, a secondary bucket with the same name is automatically created. Information stored in the primary bucket is automatically synchronized to the secondary bucket. By default, this function is enabled.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of a bucket for which you want to configure cluster-disaster recovery to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Cluster-disaster Recovery** section.
4. Click **Configure**. Turn on or turn off **Cluster-disaster Recovery**.
5. Click **Save**.

6.4.11. Configure CRR

Cross-region replication (CRR) enables the automatic and asynchronous (near real-time) replication of objects across buckets in different OSS regions. Operations such as the creation, overwriting, and deletion of objects can be synchronized from the source bucket to the destination bucket.

Prerequisites

The steps described in [Create buckets](#) are performed or a bucket is created in the region.

Context

This feature meets the requirements of geo-disaster recovery or data replication. Objects in the destination bucket are extra replicas of objects in the source bucket. They have the same object names, object content, and object metadata such as the creation time, owner, user metadata, and object ACL.

Procedure

1. Log on to the OSS console.
2. In the left-side navigation pane, click the name of a bucket for which you want to configure CRR to go to the bucket details page.
3. On the bucket details page, click the **Basic Settings** tab. Find the **Cross-Region Replication** section.
4. Click **Enable**. In the **Cross-Region Replication** dialog box that appears, configure the parameters described in the following table.

Parameter	Description
Source Region	The region where the current bucket is located.
Source Bucket	The name of the current bucket.
Destination Region	Select the region where the destination bucket is located. The source and destination buckets for CRR must be located in different regions. Data cannot be synchronized between buckets located within the same region.
Destination Bucket	Select the destination bucket. The two buckets with CRR enabled cannot synchronize data with other buckets. If you synchronize data from Bucket A to Bucket B, neither Bucket A nor Bucket B can synchronize data with other buckets.
Applied To	Select the source data to synchronize. <ul style="list-style-type: none"> ◦ All Files in Source Bucket: synchronizes all objects from the source bucket to the destination bucket. ◦ Files with Specified Prefix: synchronizes the objects whose names contain the specified prefix from the source bucket to the destination bucket. For example, if you have a folder named <i>management/</i> in the root folder of a bucket and a subfolder named <i>abc/</i> in <i>management/</i>, when you want to synchronize objects in the <i>abc/</i> subfolder, enter <i>management/abc/</i> as the prefix. You can specify up to five prefixes.
Operations	Select the synchronization policy. <ul style="list-style-type: none"> ◦ Add/Change: synchronizes only added or changed data from the source bucket to the destination bucket. ◦ Add/Delete/Change: synchronizes all data changes such as the creation, overwriting, and deletion of objects from the source bucket to the destination bucket.
Replicate Historical Data	Specify whether to synchronize historical data before you enable CRR. <ul style="list-style-type: none"> ◦ Yes: synchronizes historical data to the destination bucket. <div style="background-color: #e6f2ff; padding: 5px; margin: 5px 0;"> <p> Notice When historical data is synchronized, objects in the source bucket may overwrite objects in the destination bucket if these objects have the same name. Before you select this option, ensure that the data is consistent.</p> </div> <ul style="list-style-type: none"> ◦ No: synchronizes only objects that you want to upload or update after CRR is enabled to the destination bucket.

5. Click **OK**.

 Note

- After the configuration is complete, it may take three to five minutes for CRR to take effect. Synchronization information is displayed after the source bucket is synchronized.
- In CRR, data is asynchronously (near real-time) replicated. It takes several minutes to several hours for the data to be replicated to the destination bucket based on the amount of data.

6.4.12. Access OSS through custom domain names

You can bind a custom domain name to your bucket and add a CNAME record that points to the public endpoint of your bucket. After the CNAME record is added, you are directed to your bucket when you access the custom domain name.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Step 1: Bind the custom domain name to the bucket

You can use OSS SDK for Java to bind the custom name to the bucket. For the sample code, visit [Bind custom domain name](#).

Step 2: Configure the CNAME record

The domain name of Alibaba Cloud is used as an example to show how to configure the CNAME record.

1. Log on to the DNS console.
2. On the Manage DNS page, click **Configure** in the Actions column corresponding to the target domain name.
3. On the DNS Settings page, click **Add Record**. In the Add Record dialog box, configure parameters listed in the following table.

Parameter	Description
Type	Select the type of the record to direct requests. In this example, select <i>CNAME</i> .
Host	Enter the host record based on the prefix of the domain name. Examples: <ul style="list-style-type: none"> ○ If the domain name is <code>www.aliyun.com</code>, enter <code>www</code>. ○ If the domain name is <code>aliyun.com</code>, enter <code>@</code>. ○ If the domain name is <code>abc.aliyun.com</code>, enter <code>abc</code>. ○ If the domain is a second-level domain such as <code>a.aliyun.com</code> or <code>b.aliyun.com</code>, enter an asterisk (*).
ISP Line	Select the ISP line used to resolve the domain name. We recommend that you select <i>Default</i> to allow the system to select the optimal line.
Value	Enter the value of the record based on the selected record type. In this example, enter the public endpoint of the bucket.
TTL	Select the update interval of the record. In this example, select the default value.

4. Click **OK**.

 **Note** A new CNAME record takes effect immediately. It takes up to 72 hours for the modified CNAME record to take effect.

Verify CNAME status

After a CNAME record is configured, the period required for the record to take effect varies with different DNS providers. You can run the `ping` or `lookup` command to access the added domain name. If the access is directed to the bucket endpoint, the CNAME configuration has taken effect.

6.5. Objects

6.5.1. Search for objects

You can search buckets or folders for objects whose names contain a specified prefix in the OSS console.

Prerequisites

- A bucket is created. For more information about how to create a bucket, see [Create buckets](#).
- Objects are uploaded to the bucket. For more information about how to upload objects, see [Upload objects](#).

Context

When you search for objects based on a prefix, search strings are case-sensitive and cannot contain forward slashes (/). The search range is limited to the root directory of the current bucket or the objects in the current folder (excluding subfolders and the objects in them).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the bucket to go to the bucket details page.
3. Click the **Files** tab.
4. On the right side of the Files tab, enter the prefix to search in the search box and press Enter or click the search icon to search for related objects.

The names of the objects and folders that are stored in the current folder and match the prefix are listed.

 **Note** To search for objects in a specified folder, open the folder and enter the prefix in the search box. The names of objects and subfolders in the folder that match the prefix are listed.

6.5.2. Configure object ACL

You can configure the ACL of an object in the OSS console to control access to the object.

Prerequisites

- A bucket is created. For more information about how to create a bucket, see [Create buckets](#).
- Objects are uploaded to the bucket. For more information about how to upload objects, see [Upload objects](#).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the target bucket to go to the bucket details page.
3. Click the **Files** tab.
4. On the Files tab, click the name of the target object. The **Preview** dialog box appears.
5. Click **Set ACL** on the right side of **File ACL**. The **Set ACL** dialog box appears. The ACLs that you can select are described as follows:

- **Inherited from Bucket:** The ACL of each object is the same as that of the bucket.
- **Private:** Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
- **Public Read:** Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket.
- **Public Read/Write:** Any users, including anonymous users can read and write objects in the bucket. Fees incurred by such operations are paid by the owner of the bucket. Configure this option only when necessary.

 **Note** You can also choose **More > Set ACL** in the Actions column corresponding to the target object to open the Set ACL dialog box.

6. Click **OK**.

6.5.3. Create folders

You can create a folder in a bucket in the OSS console.

Prerequisites

A bucket is created. For more information about how to create a bucket, see [Create buckets](#).

Context

OSS does not use traditional folders. All elements are stored as objects. A folder is an object whose size is 0 and has a name that ends with a forward slash (/). A folder is used to sort objects of the same type and process them at a time. The OSS console displays objects that end with a forward slash (/) as folders. These objects can be uploaded and downloaded. You can use OSS folders in the OSS console the way you use folders in Windows.

 **Note** The OSS console displays any objects whose names end with a forward slash (/) as folders, regardless of whether these objects contain data. You can download these objects only by calling an API operation or by using an SDK.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the target bucket to go to the bucket details page.
3. Click the **Files** tab. On the page that appears, click **Create Folder**.
4. In the **Create Folder** dialog box that appears, enter the folder name. The folder name must comply with the following conventions:
 - The name can contain only UTF-8 characters. The name cannot contain emojis.
 - The name cannot start with a forward slash (/) or backslash (\). The name cannot contain consecutive forward slashes (/). You can use forward slashes (/) in a folder name to quickly create a subfolder.
 - A subfolder cannot contain two consecutive periods (..) in its name.
 - The folder name must be 1 to 254 characters in length.
5. Click **OK**.

6.5.4. Delete objects

You can delete uploaded objects in the OSS console.

Context

You can delete one or more objects at a time. A maximum of 100 objects can be deleted at a time. You can use SDKs or call an API operation to delete a specific object or more than 100 objects.

 **Notice** Deleted objects cannot be recovered. Exercise caution when you delete objects.

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the target bucket to go to the bucket details page.
3. Click the **Files** tab.
4. Select one or multiple objects that you want to delete in the object list, and choose **Batch Operation > Delete**. You can also choose **More > Delete** in the Actions column corresponding to the target object.
5. In the message that appears, click **OK**.

6.5.5. Manage parts

When you upload an object in multipart upload mode, the object is split into several smaller parts. After all of the parts are uploaded to the OSS server, you can call `CompleteMultipartUpload` to combine them into a complete object. We recommend that you delete unnecessary parts on a regular basis.

Context

Parts are generated in multipart upload tasks and cannot be read until they are combined into a complete object. To save storage space in the bucket, you can configure lifecycle rules to manage unnecessary parts that are generated when multipart upload tasks fail. For more information, see [Manage lifecycle rules](#).

Procedure

1. [Log on to the OSS console](#).
2. In the left-side navigation pane, click the name of the target bucket to go to the bucket details page.
3. Click the **Files** tab. On the page that appears, click **Parts**.
4. In the **Parts** dialog box that appears, delete parts.
 - To delete all parts in the bucket, select all parts and click **Delete All**.
 - To delete specified parts, select the parts that you want to delete and click **Delete**.
5. In the message that appears, click **OK**.

7. Apsara File Storage NAS

7.1. What is Apsara File Storage NAS?

NAS is a cloud service that provides file storage for compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

NAS supports multiple standard file access protocols. NAS is a distributed file system that offers a variety of benefits. These benefits include unlimited capacity, scalable performance, shared access, high reliability, and high availability. You can work with NAS without the need to modify existing applications. Compared with traditional user-created data stores, NAS helps you reduce a large number of maintenance costs and mitigate data security risks. You can mount an NAS file system on multiple compute nodes at the same time. This helps you reduce a large number of costs in data transmission and synchronization.

You can perform the following operations on an NAS file system:

- Create NAS file systems and mount points.
- Create permission groups for NAS file systems and add rules to permission groups. This allows access to file systems from specific IP addresses or CIDR blocks. This also allows you to grant different levels of access permissions to IP addresses or CIDR blocks.
- Mount file systems on compute nodes. These compute nodes include ECS instances and ACK nodes. NAS allows you to access file systems by using the standard Network File System (NFS) and Server Message Block (SMB) protocols. You can also use POSIX-based APIs to access file systems.
- Manage file systems, mount points, and permission groups in the NAS console.
- Call NAS API operations to manage file systems.

7.2. Precautions

Before you use NAS, you must familiarize yourself with the following limits.

Limits on file systems

- Maximum number of files in a single file system: 1 billion.
- Maximum name length: 255 bytes.
- Maximum size of a single file: 32 TB.
- Maximum directory depth: 1,000 levels deep.
- Maximum capacity of a single file system: 10 PB for NAS Capacity and 1 PB for NAS Performance.
- Maximum number of compute nodes on which you can mount a single file system: 10,000. Note: The file system allows simultaneous access from the 10,000 compute nodes.
- Maximum size of a protocol packet: 4 MB.
- Maximum number of Change Notify requests: 512.

Limits on NFS clients

Limits on the usage of NFS clients are listed as follows.

- You can open a maximum of 32,768 files at a time on an NFS client. Files in the list folder and its subfolders are not counted as part of the total number of open files.
- Each unique mount on an NFS client can acquire a maximum of 8,192 locks across a maximum of 256 unique file or process pairs. For example, a single process can acquire one or more locks on 256 separate files, or 8 processes can each acquire one or more locks on 32 files.
- We recommend that you do not use an NFS client in a Windows environment to access an NFS file system.

Limits on SMB clients

Each file or folder can be opened a maximum of 8,192 times in parallel across compute nodes that each have a file system mounted and users that share access to each of these file systems. This represents a maximum of 8,192 active file handlers for each file system. A maximum of 65,536 active file handlers can exist on a file system.

Limits on the NFS protocol

- NAS supports the NFSv3 and NFSv4 protocols.
- NFSv4.0 does not support the following attributes: `FATTR4_MIMETYPE`, `FATTR4_QUOTA_AVAIL_HARD`, `FATTR4_QUOTA_AVAIL_SOFT`, `FATTR4_QUOTA_USED`, `FATTR4_TIME_BACKUP`, and `FATTR4_TIME_CREATE`. If one of the preceding attributes is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on a client that has the file system mounted.
- NFSv4.1 does not support the following attributes: `FATTR4_DIR_NOTIF_DELAY`, `FATTR4_DIR_NOTIF_DELAY`, `FATTR4_DACL`, `FATTR4_SACL`, `FATTR4_CHANGE_POLICY`, `FATTR4_FS_STATUS`, `FATTR4_LAYOUT_HINT`, `FATTR4_LAYOUT_TYPES`, `FATTR4_LAYOUT_ALIGNMENT`, `FATTR4_FS_LOCATIONS_INFO`, `FATTR4_MDSTHRESHOLD`, `FATTR4_RETENTION_GET`, `FATTR4_RETENTION_SET`, `FATTR4_RETENT_EVT_GET`, `FATTR4_RETENT_EVT_SET`, `FATTR4_RETENTION_HOLD`, `FATTR4_MODE_SET_MASKED`, `FATTR4_FS_CHARSET_CAP`. If one of the preceding attributes is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on a client that has the file system mounted.
- NFSv4 does not support the following operations: `OP_DELEGPURGE`, `OP_DELEGRETURN`, and `NFS4_OP_OPENATTR`. If one of the preceding operations is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on a client that has the file system mounted.
- NFSv4 does not support delegations.
- The following issues are related to user IDs (UIDs) and group IDs (GIDs):
 - On Linux, mappings between UIDs or GIDs and usernames or group names are defined in configuration files. For NFSv3 file systems, if the mapping between an ID and a name is defined in a configuration file, the name is displayed. If no mapping can be found for a UID or GID, the UID or GID is displayed.
 - For NFSv4 file systems, the usernames and group names are displayed as `nobody` for all files if the version of a Linux kernel is earlier than 3.0. If the kernel version is later than 3.0, the rule used by NFSv3 file systems applies to display files.

 **Notice** If a file or directory is stored on an NFSv4 file system and the Linux kernel version is earlier than 3.0, we recommend that you do not use the `chown` or `chgrp` command. If you use either one of the commands, the UID and GID of the file or directory will change to `nobody`.

Limits on the SMB protocol

- NAS supports protocols including SMB 2.1 or later and operating systems including Windows 7, and Windows Server 2008 R2 or later. However, NAS does not support Windows Vista, or Windows Server 2008 or earlier. Compared with SMB 2.1 or later, SMB 1.0 has lower performance and functionality. Furthermore, Windows products that support SMB 1.0 are no longer offered or supported.
- Extended file attributes and client-side caching based on leases.
- Input/output control (IOCTL) or file system control (FSCTL) operations, such as creating sparse files, compressing files, retrieving NIC status, and creating reparse points.
- Alternate data streams.
- Identity authentication provided by Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).
- Several features provided by SMB 3.0 or later, such as SMB Direct, SMB Multichannel, SMB Directory Leasing, and persistent handles.
- Access control lists (ACLs) on files or directories.

7.3. Quick start

7.3.1. Log on to the Apsara File Storage NAS console

This topic describes how to log on to the Apsara File Storage NAS console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, select **Products** and click **Apsara File Storage NAS**.

7.3.2. Create a file system

This topic describes how to create a file system in the Apsara File Storage NAS console.

Context

Before you create a file system, you must note the following limitations:

- You can use an Alibaba Cloud account to create a maximum of 1,000 file systems.
- The maximum capacity of a NAS Performance file system is 1 PB. The maximum capacity of a NAS Capacity file system is 10 PB.

If you want to increase the maximum storage capacity, we recommend that you contact Alibaba Cloud Technical Support.

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List** and click **Create File System**.
3. In the **Create File System** dialog box, set the required parameters.

Create File System

Region

*Organization:

*Resource Set:

*Region:

Basic Settings

File System Name:

The value must be 2 to 256 characters in length and start with a letter or a C

Storage Configurations

*Storage Type:

After the file system is created, you cannot change the storage type.

*Protocol Type:

After the file system is created, you cannot change the protocol type.

*Capacity (TB):

The following table lists the required parameters.

Parameter	Description
Region	Select a region where you need to create a file system.
Organization	Select an organization from the drop-down list for the instance.
Resource Set	Select a resource set from the drop-down list for the instance.
File System Name	The name of the file system. The name must be 2 to 256 characters in length and can contain letters, digits, and special characters. These special characters include underscores (_) and hyphens (-). The name must start with a letter and cannot start with http:// or https://.
Storage Type	The storage type. Select Performance or Capacity based on your business requirements. The maximum capacity of an NAS Performance file system is 1 PB. The maximum capacity of an NAS Capacity file system is 10 PB.
Protocol Type	The protocol type. Select NFS or SMB based on your business requirements. We recommend that you mount Network File System (NFS) file systems on Linux clients and Server Message Block (SMB) file systems on Windows clients.

Parameter	Description
Capacity (TB)	The capacity of the file system. <ul style="list-style-type: none">○ The capacity of an NAS Performance file system ranges from 0.5 TB to 1024 TB.○ The capacity of an NAS Capacity file system ranges from 0.5 TB to 10240 TB.

4. Click **OK** to complete the creation.

7.3.3. Create a permission group and add rules

This topic describes how to create a permission group and add rules to the permission group in the Apsara File Storage NAS console.

Context

In NAS, each permission group represents a whitelist. You can add rules to a permission group to allow access to a file system from specific IP addresses or CIDR blocks. You can also grant different access permissions to different IP addresses or CIDR blocks.

 **Note** You can use an Alibaba Cloud account to create a maximum of 100 permission groups. If you want to increase the limit, we recommend that you contact Alibaba Cloud Technical Support.

Creates a permission group

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > Permission Group** and click **Create Permission Group**.
3. In the **Create Permission Group** dialog box, specify the required parameters.

Create Permission Group

Region

*Organization:

*Resource Set:

*Region:

Basic Settings

*Name:
The value must be 3 to 64 characters in length and can contain letters, digits, and hyphens (-).

*Network Type:

Description:
The value must be 2 to 128 characters in length and start with a letter or a Chinese character. It cannot contain spaces, and cannot start with string http:// or https://.

The following table lists the required parameters.

Parameter	Description
Organization	The organization to which the permission group belongs.
Resource Set	The resource set to which the permission group belongs.
Region	The region where you want to create the permission group.
Name	The name of the permission group. The name must be 3 to 64 characters in length and can contain letters, digits, and hyphens (-).
Network Type	The network type. Select Classic Network or VPC based on your business requirements.

- Click **OK** to complete the creation of the permission group.

Create a rule

- Log on to the [Apsara File Storage NAS console](#).
- On the **Permission Group** page, find the target permission group and click **Manage**.
- Click **Add Rule**.
- In the **Add Rule** dialog box that appears, specify the required parameters.

Add rules
✕

* Authorized address ?

* Read and write permissions

* User permissions

* Priority ?

The following table lists the required parameters.

Parameter	Description
Authorization Address	Specifies the authorized object to which the rule applies. You can specify an IP address or CIDR block. Only IP addresses are available for permission groups of the classic network type.
Read/Write Permission	Specifies whether to allow read-only or read/write access to the file system from the authorized object. Valid values: Read-only and Read/Write.
User Permission	<p>Specifies whether to limit a Linux user's access to a file system.</p> <ul style="list-style-type: none"> ◦ Do not limit root users (no_squash): allows access to a file system from root users. ◦ Limit root users (root_squash): denies access to a file system from root users. All root users are treated as nobody users. ◦ Limit all users (all_squash): denies access to a file system from all users including root users. All users are treated as nobody users. <p>The nobody user is created by default on Linux. The user has only the most basic permissions and can access only the open content of servers. This feature offers high security.</p>
Priority	When multiple rules are applied to an authorized object, the rule with the highest priority takes effect. Valid values: 1 to 100, in which 1 is the highest priority.

5. Click **OK** to complete the creation of the rule.

7.3.4. Add a mount target

This topic describes how to add a mount target. After an Apsara File Storage NAS file system is created, you must add a mount target to the file system. Then, you can use the mount target to mount the file system on compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add](#)

rules.

Context

A mount target is an endpoint that resides in a VPC or classic network. Each mount target corresponds to a file system. Mount targets of the VPC and classic network types are available for NAS file systems.

Note You use a mount target to mount a file system on multiple compute nodes for shared access. These compute nodes include ECS instances and ACK nodes.

Procedure

1. Log on to the Apsara File Storage NAS console.
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. On the **Mount Target** tab, click **Add Mount Target**.
5. In the **Add Mount Target** dialog box that appears, specify the required parameters.

Mount Target Type: includes **VPC** and **Classic Network**.

Note Mount targets of the classic network type allow access only from ECS instances that belong to the same Alibaba Cloud account as the mount targets.

Add mount point
✕

File system 1b45449e09

* Mount point type VPC ▼

?

* VPC network ? vpc-1goag4colq5uavdymt9r5(172.31.0.0/16) ▼

* Switch ? vsw-1gopa1uzsei6mult75xno(172.31.144.0/20) ▼

* Permission Group nas-permission-01 ▼

?

OK
Cancel

- o If you want to create a mount target of the VPC type, specify the following parameters.

Parameter	Description
VPC	The VPC. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> <p>Note The VPC you specify must be the same as the VPC where the compute nodes reside. These compute nodes include ECS instances and ACK nodes.</p> </div>
VSwitch	The VSwitch.

Parameter	Description
Permission Group	The permission group.

- If you want to create a mount target of the classic network type, specify the following parameters.

Parameter	Description
Permission Group	The permission group.

6. Click **OK** to complete the configuration.

7.3.5. Mount an NFS file system

This topic describes how to mount a Network File System (NFS) file system. Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. This topic takes a mount target of the VPC type as an example. For more information, see [Add a mount target](#).
 - If you create a mount target of the VPC type for a file system, you can mount the file system only on ECS instances that reside in the same VPC as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the VPC that hosts the ECS instances.
 - If you create a mount target of the classic network type for a file system, you can mount the file system only on ECS instances that belong to the same Alibaba Cloud account as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the private network that hosts the ECS instances.
- A compute node is created. This topic takes a Linux ECS instance as an example.

Step 1: Install an NFS client

Before you mount an NFS file system on a Linux ECS instance, you must install an NFS client. If an NFS client is installed, skip this step.

1. Log on to the Linux ECS instance. For more information, see the [Quick start > Connect to an ECS instance](#) topic of the *ECS User Guide*.
2. Install the NFS client.

- If CentOS, RHEL, or Aliyun Linux runs on the ECS instance, use the following command to install the NFS client.

```
sudo yum install nfs-utils
```

- If Ubuntu or Debian runs on the ECS instance, use the following commands to install the NFS client.

```
sudo apt-get update
```

```
sudo apt-get install nfs-common
```

Step 2: Mount an NFS file system

1. Log on to the Linux ECS instance. For more information, see the [Quick start > Connect to an ECS instance](#) topic of the *ECS User Guide*.
2. Mount the NFS file system.

Use the following command to mount the NFS file system. In the command, replace `file-system-id.region.nas.aliyuncs.com:/mnt` with a value that is specific to your environment.

- To mount an NFSv4 file system, use the following command.

```
sudo mount -t nfs -o vers=4.0,minorversion=0,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/mnt
```

- To mount an NFSv3 file system, use the following command.

```
sudo mount -t nfs -o vers=3,nolock,proto=tcp,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/mnt
```

Mount parameters

Parameter	Description
<code>file-system-id.region.nas.aliyuncs.com:/mnt</code>	<p>Specifies the mount target of the NAS file system, the forward slash (/) following the mount target specifies the root directory of the NAS file system, and /mnt specifies a local directory that resides on the Linux ECS instance. You must replace the example values based on your business requirements.</p> <ul style="list-style-type: none"> ■ The mount target, for example, <code>file-system-id.region.nas.aliyuncs.com</code>. To obtain information about a mount target, follow these steps. Log on to the NAS console, find the target system, click Manage next to the file system to go to the Details page. The Details page shows information about the mount target. ■ The directory of the NAS file system: specifies the root directory (/) or a subdirectory (/sub1). If a subdirectory is specified, make sure that the subdirectory exists. ■ The local directory on which you want to mount a file system: specifies the root directory (/) or a subdirectory (/mnt) of a system such as Linux. If a subdirectory is specified, make sure that the subdirectory exists.
<code>vers</code>	The version of the file system. Only NFSv3 and NFSv4 are available.
Mount option	<p>When you mount a file system, multiple mount options are available. Separate multiple mount options with commas (.). For more information, see the following Mount options.</p> <p>Note When you specify mount options, take note of the following items.</p> <ul style="list-style-type: none"> ■ To avoid a decrease in performance, we recommend that you specify the maximum value (1048576) for both the <code>rsize</code> mount option and the <code>wsiz</code> mount option. ■ If you need to modify the <code>timeo</code> mount option, we recommend that you specify a minimum of 150 for the mount option. The <code>timeo</code> mount option is measured in deciseconds (tenths of a second). For example, a value of 150 indicates 15 seconds. ■ To avoid data inconsistency, we recommend that you do not use the <code>soft</code> mount option. Use caution with the <code>soft</code> mount option. ■ We recommend that you use the default values for other mount options. For example, a decrease in performance may occur due to changes in some mount options. These mount options include the size of the read or write buffer or the use of attribute caching.

Mount options

Option	Description
rsize	Specifies the maximum number of bytes in each read request that the NFS client can receive. Recommended value: 1048576.
wsize	Specifies the maximum number of bytes in each write request that the NFS client can send. Recommended value: 1048576.
hard	Specifies that applications must stop accessing a file system when the file system is unavailable, and wait until the file system is available. We recommended that you use the hard mount option.
timeo	Specifies the time in deciseconds (tenths of a seconds) that the NFS client waits before it retries an NFS request. Recommended value: 600.
retrans	Specifies the number of times the NFS client retries a request. Recommended value: 2.
noresvport	Specifies that the NFS client uses a different TCP source port for a new network connection to ensure data integrity. We recommend that you use the noresvport mount option.

- Use the `mount -l` command to view the mount result.

The following figure shows an example of a successful mount.

```

[root@ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvdc1     100G  100G   0% 100% /
tmpfs           16G   0%  16G   0% /dev/shm
tmpfs           16G   0%  16G   0% /run
tmpfs           16G   0%  16G   0% /tmp
tmpfs           16G   0%  16G   0% /var/tmp

```

After a file system is mounted, you can use the `df -h` command to view the size of the file system.

- After you mount an NAS file system on an ECS instance, you can access the file system from the ECS instance.

You can access the file system in the same way you access a local directory. The following figure shows an example.

```

[root@iZwe5f6ow1q4t21d4g0f16q2 ~]# mkdir /mnt/dir1
[root@iZwe5f6ow1q4t21d4g0f16q2 ~]# mkdir /mnt/dir2
[root@iZwe5f6ow1q4t21d4g0f16q2 ~]# touch /mnt/file1
[root@iZwe5f6ow1q4t21d4g0f16q2 ~]# echo 'some file content' > /mnt/file2
[root@iZwe5f6ow1q4t21d4g0f16q2 ~]# ls /mnt
dir1 dir2 file1 file2 tmp

```

7.3.6. Mount an SMB file system

This topic describes how to mount a Server Message Block (SMB) file system. Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes such as Elastic Compute Service (ECS) instances.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. This topic takes a mount target of the Classic Network type as an example. For more information, see [Add a mount target](#).
 - If you create a mount target of the VPC type for a file system, you can mount the file system only on ECS instances that reside in the same VPC as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the VPC that hosts the ECS instances.
 - If you create a mount target of the Classic Network type for a file system, you can mount the file system

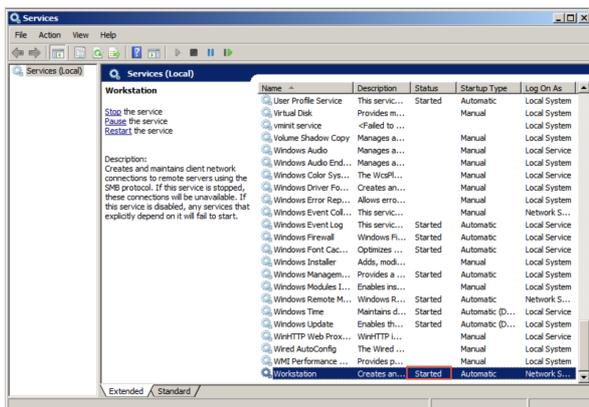
only on ECS instances that belong to the same Alibaba Cloud account as the mount target. You can specify a permission group for the mount target. Then, you can add several rules to the permission group. The authorization address of a rule must match the IP range of the private network that hosts the ECS instances.

4. An ECS instance is created. This topic takes a Windows ECS instance as an example.

5. The following Windows services are started:

- Workstation
 - a. Choose **All Programs > Accessories > Run**, or press **Win+R** and enter **services.msc** to open the Services console.
 - b. Find the Workstation service and ensure that the service is **Running** and the startup type is **Automatic**.

The default state for the Workstation service is Running.

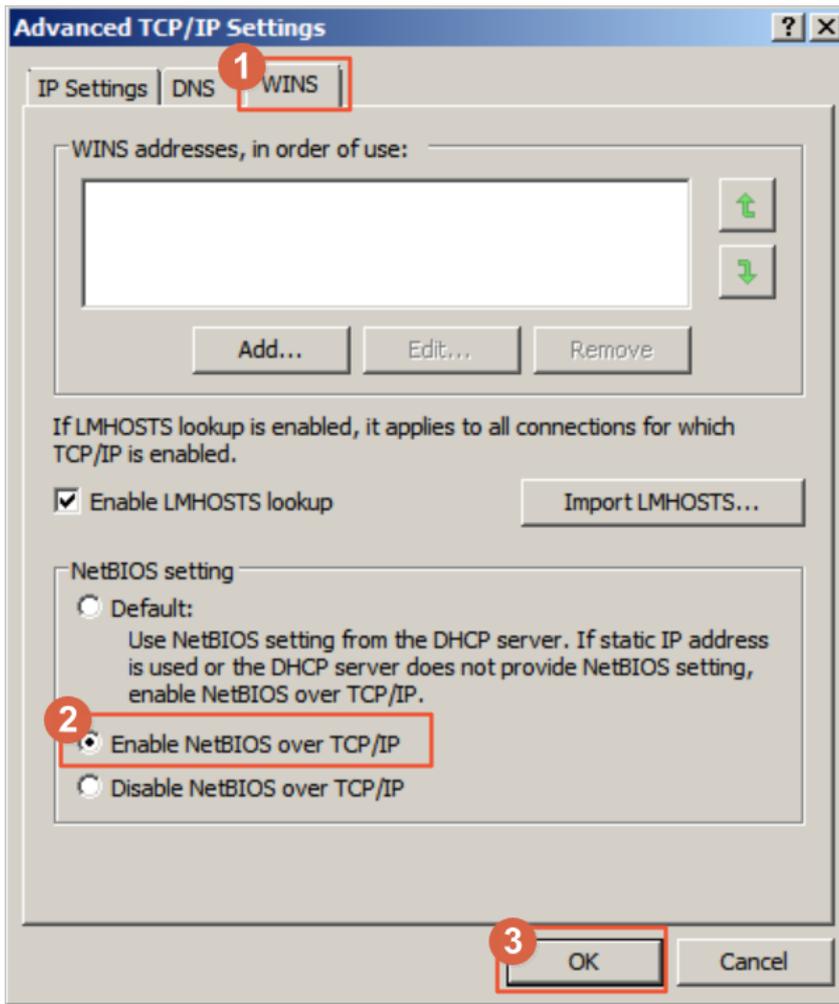


- TCP/IP NetBIOS Helper

Follow these steps to start the TCP/IP NetBIOS Helper service:

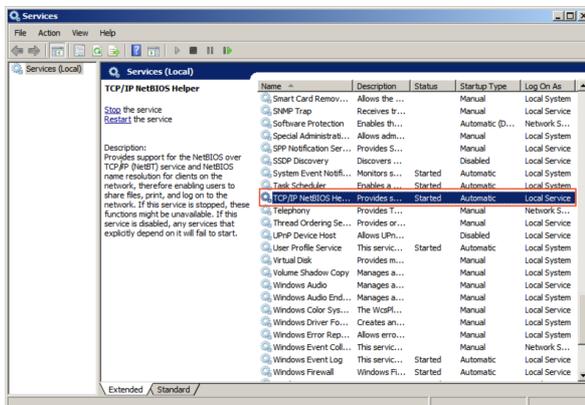
- a. Double-click **Network and Sharing Center** and right-click the active network connection.
- b. Click **Properties** to open the **Local Area Network Properties** dialog box. Double-click **Internet Protocol Version 4 (TCP/IPv4)** to open the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, and then click **Advanced**.

- c. In the Advanced TCP/IP Settings dialog box, choose WINS > Enable NetBIOS over TCP/IP.



- d. Choose All Programs > Accessories > Run, or press Win+R and enter services.msc to open the Services console.
- e. Find the TCP/IP NetBIOS Helper service and make sure that the service is Running and the startup type is Automatic.

The default state for the TCP/IP NetBIOS Helper service is Running.



Procedure

1. Log on to a Windows ECS instance. For more information, see the Quick start > Connect to an ECS instance topic of the ECS User Guide .

- Open the command prompt and use the following command to mount the file system.

```
net use D: \\file-system-id.region.nas.aliyuncs.com\myshare
```

The syntax of the command is `net use <the letter of a local drive> \\<the domain name of a mount target>\myshare .`

- The letter of a local drive: specifies a drive on which you need to mount a file system. You can specify the target drive based on your business requirements.

Note The target drive must be different from any existing drives.

- The domain name of the mount target: When you create a mount target for a file system, the domain name of the mount target is automatically generated. You can replace the domain name based on your business requirements. To obtain the mount target of the file system, follow these steps. Log on to the NAS console, find the target file system, and click **Manage** to go to the Details page.
 - myshare: specifies the name of an SMB share. You cannot change the name.
- Use the `net use` command to check mount results.

The following figure shows an example of a successful mount.



```
C:\Users\Administrator>net use
New connections will be remembered.

Status      Local      Remote                                           Network
-----
OK          D:         \\6... .nas.aliyuncs.com\myshare                Microsoft Windows Network

The command completed successfully.
```

- After you mount an NAS file system on an ECS instance, you can access the file system from the ECS instance.

7.4. File systems

7.4.1. View the details of a file system

This topic describes how to view the details of a file system. The details include the information about the file system and information about the attached mount targets.

Prerequisites

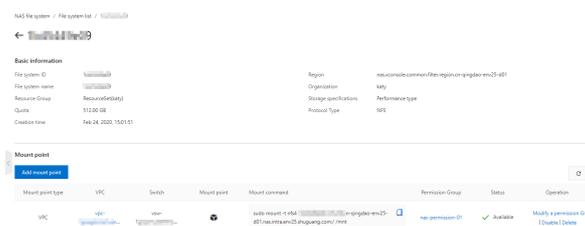
A file system is created. For more information, see [Create a file system](#).

Procedure

- Log on to the [Apsara File Storage NAS console](#).
- Choose **NAS > File System List**.
- Find the target file system and click **Manage** to go to the Details page of the file system.

The following sections are available on the Details page of the file system.

- Basic Information:** includes information about the file system. The information includes the ID, region, protocol type, and storage type.
- Mount Target:** includes mount targets for the file system. You can manage mount targets in the section.



7.4.2. Delete a file system

This topic describes how to delete a file system in the Apsara File Storage NAS console.

Prerequisites

A file system is created. For more information, see [Create a file system](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List**.
3. Find the target file system, and click **Manage**.

Note

- Before you can delete a file system, you must remove all mount targets from the file system.
- Use caution when you delete a file system. After a file system is deleted, the data on the file system cannot be restored. We recommend that you ensure that all data is backed up.

4. In the **Delete File System** dialog box, click **OK** to complete the deletion.

7.5. Mount targets

7.5.1. View mount targets

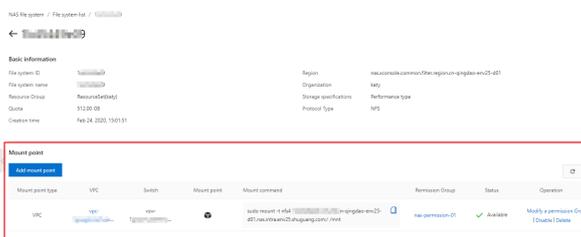
This topic describes how to view mount targets in the Apsara File Storage NAS console.

Context

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. For more information, see [Add a mount target](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. In the **Mount Target** section, view mount targets in the file system.



7.5.2. Enable or disable a mount target

This topic describes how to enable or disable a mount target. You can control access to a mount target from clients by enabling or disabling the mount target.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. For more information, see [Add a mount target](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. After you find the mount target that you want to disable or enable, you can perform the following operations:
 - **Disable the mount target.** Click **Disable**. In the Disable Mount Target dialog box, click **OK** to deny access to the mount target from clients.
 - **Enable the mount target.** Click **Enable**. In the Enable Mount Target dialog box, click **OK** to allow access to the mount target from clients.



7.5.3. Delete a mount target

This topic describes how to delete a mount target in the Apsara File Storage NAS console.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. For more information, see [Add a mount target](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List**.
3. Find the target file system and click **Manage**.
4. Find the mount target that you want to delete and click **Delete**.

Note Use caution when you delete a mount target. After you delete a mount target, the mount target cannot be restored.

5. In the Delete Mount Target dialog box, click **OK**.

7.5.4. Modify the permission group of a mount target

You must add a permission group to each mount target. You can modify the permission group of a mount target in the Apsara File Storage NAS console.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group and a rule are created. For more information, see [Create a permission group and add rules](#).
- A mount target is created. For more information, see [Add a mount target](#).

Procedure

1. **Log on to the Apsara File Storage NAS console.**
2. **Choose NAS > File System List.**
3. **Find the target file system, and click **Manage**.**
4. **Find the mount target that you want to modify and click **Modify Permission Group**.**



5. **In the **Modify Permission Group** dialog box, change the permission group and click **OK**.**

7.6. Permission groups

7.6.1. View permission groups

This topic describes how to view permission groups in the Apsara File Storage console.

Prerequisites

A permission group is created. For more information, see [Create a permission group and add rules](#).

Procedure

1. **Log on to the Apsara File Storage NAS console.**
2. **Choose **NAS > Permission Group** and view permission groups in the region.**



7.6.2. Delete a permission group

This topic describes how to delete a permission group in the Apsara File Storage NAS console.

Prerequisites

A permission group is created. For more information, see [Create a permission group and add rules](#).

Operating system

1. **Log on to the Apsara File Storage NAS console.**
2. **Choose **NAS > Permission Group**.**
3. **Find the target permission group and click **Delete**.**

Note Permission groups in use cannot be deleted. Before you can delete a permission group, you must remove the permission group from the linked mount target.

4. **In the **Delete Permission Group** dialog box, click **OK**.**

7.6.3. Manage permission group rules

This topic describes how to manage permission group rules in the Apsara File Storage NAS console. The management includes viewing the details of rules, modifying rules, and deleting rules.

Prerequisites

A permission group and a permission group rule are created. For more information, see [Create a permission group and add rules](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > Permission Group**.
3. Find the target permission group and click **Manage**.
4. On the Rules page, you can perform the following operations:
 - View all rules for the permission group.



- Modify a rule. Find the target rule and click **Edit** to modify the details of the rule. The details include the authorization address, read/write permissions, user permission, and priority.
- Delete a rule. Find the target rule and click **Delete**. In the Delete Rule dialog box, click **OK**.

7.7. Manage quotas

This topic describes how to use the Alibaba Cloud `quota_tool` tool to manage quotas on Elastic Compute Service (ECS) instances that have Apsara File Storage NAS file systems mounted. You can configure, view, and cancel quotas on these ECS instances.

Prerequisites

An NFS file system of the NAS Capacity or NAS Performance type is mounted on an ECS instance. For more information, see [Mount an NFS file system](#).

Context

NAS allows you to view and manage directory-level quotas. Directory-level quotas specify the maximum number of files in each directory and the maximum storage space for these files.

From the perspective of the application scope, quotas are sorted into quotas for all users and quotas for a single user or group. Quotas for all users specify the maximum storage space for files that all users can create in a directory. Quotas for a single user or group specify the maximum storage space for files that a user or group can create in a directory.

From the perspective of the restriction level, quotas are sorted into statistical quotas and restriction quotas. Statistical quotas collect only the usage of storage space. You can query and view statistical data. Restriction quotas specify the maximum capacity of storage space for files that you can create in a directory. If the limit is exceeded, you may fail to create a file or subdirectory, append data to a file, or perform other operations.

Notice

- Only statistical quotas are available.
- NAS performs asynchronous calculation for quotas at the backend. When you use the `quota_tool` tool to query statistical data about quotas, the process requires a period of time to complete. In most cases, the time period is about 5 to 15 minutes.

Configure quotas

This topic uses the `/mnt` directory as an example.

1. Log on to an Elastic Compute Service (ECS) instance by using a root account. You can use the `quota_tool` tool on an ECS instance that has a NAS file system mounted. You must run the tool with the root permissions. The following describes how to use the `quota_tool` tool on the ECS instance.
2. Use the following command to download the `quota_tool` tool.

```
wget https://nasimport.oss-cn-shanghai.aliyuncs.com/quota_tool_v1.0 -O quota_tool
```

3. Use the following command to grant the execute permission to the quota_tool tool.

```
sudo chmod a+x quota_tool
```

4. Configure quotas.

Note For each file system, you can configure quotas only for a maximum of 10 directories.

The syntax of the command that you use to configure quotas is `sudo ./quota_tool set --dir [DIR] [OPTION]` .

Parameter	Description
--dir [DIR]	Specifies the directory for which you want to configure quotas. For example, --dir /mnt/data/.
OPTION	<p>Specifies the required options.</p> <p>Note When you specify options, you must follow these rules: 1. the --accounting option is required. 2. One of the --alluser, --uid, and --gid options is required. .</p> <ul style="list-style-type: none"> ○ --accounting: specifies a statistical quota. ○ --alluser: specifies a directory-level quota for all users. ○ --uid: specifies the UID of a user. For example, --uid 505 indicates the quota is configured only for the user whose UID is 505. ○ --gid: specifies the GID of a group. For example, --gid 1000 indicates the quota is configured only for the group whose GID is 1000.

The following examples describe how to configure quotas.

- Use the following command to configure a statistical quota for the /mnt/data/ directory to limit the total number of files that reside in a directory.

```
sudo ./quota_tool set --dir /mnt/data/ --accounting --alluser
```

- Use the following command to configure a statistical quota for the /mnt/data/ directory to limit the total number of files that can be created by the user whose UID is 505.

```
sudo ./quota_tool set --dir /mnt/data/ --accounting --uid 505
```

Query quotas

After you configure a quota for an NAS directory, you can query statistical data about the quota for the directory.

1. Log on to an Elastic Compute Service (ECS) instance by using a root account.
2. Use the following command to query quotas.

```
sudo ./quota_tool get --dir /mnt/data/ --all
```

In the preceding command, the --all parameter is optional. If you specify the parameter, statistical data about all quotas that are configured for the file system returns.

Note

- The first time you query a quota, a state called Initializing appears. After the Initializing process is complete, you can query the quota and a result showing success appears. The duration of the initialization process is based on the number of files and subdirectories in a directory.
- After the initialization process is complete, you can query quotas daily. A delay of 5 to 10 minutes may occur before the expected FileCountReal and SizeReal appear. This occurs due to the asynchronous calculation for quotas at the backend.

```
{
  "Reports" : [
    {
      "Path" : "/mnt/data",
      "Report" : [
        {
          "FileCountLimit" : "Empty",
          "FileCountReal" : "2",
          "Gid" : "All",
          "Quotatype" : "Accounting",
          "SizeLimit" : "Empty",
          "SizeReal" : "4KB",
          "Uid" : "All"
        }
      ],
      "ReportStatus" : "Success"
    }
  ],
  "Status" : 0
}
```

The following table lists parameters that are included in a response in the JSON format.

Parameter	Description
Path	Indicates a directory for which you query a quota.
Report	Includes all information about a quota that is specified for a directory, for example, UID and GID.
ReportStatus	The state for the query of a quota.
FileCountLimit	Indicates the limit for the number of files. A value of Empty indicates no limit.
FileCountReal	Indicates the total number of files including subdirectories, files, and special files that reside in a directory.
QuotaType	Accounting indicates a statistical quota and Force indicates a restriction quota.
Uid	Indicates the UID of a user. A value of All indicates all users.
Gid	Indicates the GID of a group. A value of All indicates all groups.
SizeLimit	Indicates the maximum capacity of files that reside in a directory. A value of Empty indicates no limit.
SizeReal	Indicates the total capacity of files that reside in a directory.

Cancel quotas

You can cancel a quota.

1. Log on to an ECS instance.
2. Use the following command to cancel quotas.

The syntax of the command that you can use to cancel quotas is `sudo ./quota_tool cancel --dir [DIR] [OPTION]`

.

Parameter	Description
--dir [DIR]	Specifies the directory for which you want to cancel quotas, for example, -dir /mnt/data/.
OPTION	<p>Specifies the required options.</p> <p> Note When you specify the OPTION parameter, one of the --alluser, --uid, and --gid options is required.</p> <ul style="list-style-type: none"> ○ --alluser: specifies a directory-level quota for all users. ○ --uid: specifies the UID of a user. For example, --uid 505 indicates that the quota is canceled for the user whose UID is 505. ○ --gid: specifies the GID of a group. For example, --gid 505 indicates that the quota is canceled for the group whose GID is 505.

The following examples describes how to cancel quotas.

- If you have configured a quota for the /mnt/data/ directory, use the following command to cancel the quota for the user whose UID is 100.

```
sudo ./quota_tool cancel --dir /mnt/data/ --uid 100
```

- If you have configured a quota for the /mnt/data/ directory, use the following command to cancel the quota for all users.

```
sudo ./quota_tool cancel --dir /mnt/data/ --alluser
```

7.8. Create and manage a unified namespace

This topic describes how to create a unified namespace and add mount targets in the Apsara File Storage NAS console. The topic also describes how to add, remove, and modify file systems in a namespace, view namespace details, and enable the cross-domain mount orchestration feature.

Features

A unified namespace allows you to mount multiple file systems in a NAS cluster by using a single domain name. You do not need to maintain multiple mount targets and directories.

You can create mount targets for a namespace. You can use a unified namespace to manage multiple file systems in the same way as you manage a single file system.

A unified namespace contains a virtual root directory in which file systems are the first-level subdirectory. You can mount the file systems by using their mount targets even if you add the file systems to a unified namespace.

Limits

A unified namespace has the following limits:

- You can add a maximum of 1,000 file systems to each unified namespace.
- The mapping name of a file system in a unified namespace cannot exceed 255 characters in length. It can contain only letters, digits, and the following special characters:

.+-_()<>@#

 **Note** The mapping name of a file system in a namespace cannot exceed 255 characters. It can contain only letters, digits, and the following special characters:

.+-_()<>@#

- The number of mount targets that are created for a namespace must be the same as that of file systems.
- You can create a maximum of 20 namespaces in each region.
- You can mount a unified namespace only by using the NFSv3 protocol.
- The file systems that are added to a namespace must belong to the same user and cluster as the namespace. The storage type, protocol type, and encryption type of the file systems must be the same.
- The mapping name of a file system must be unique in each unified namespace.
- File systems can be mapped only to first-level subdirectories in a unified namespace. You cannot modify the access permissions, owner, or access control list (ACL).

Procedure

- Create a unified namespace and add mount targets.

Log on to the Apsara File Storage NAS console. In the left-side navigation pane, choose **Unified Namespace > Unified List of Namespaces**. On the page that appears, you can create a unified namespace and add mount targets.

 **Note** To identify CIDR blocks when you mount unified namespaces across regions, we recommend that you use different VPC CIDR blocks. For example, you can use `192.168.0.0/16` for Region 1 and `172.16.0.0/16` for Region 2. For more information, see [Cross-domain mount orchestration](#).

- Add a file system in a unified namespace.

After you create a unified namespace, you can add file systems and set the mapping name.

 **Note** The mapping name is the name of the virtual directory.

- Remove a file system from a unified namespace.

You can remove an existing file system from the unified namespace.

 **Note** The file system is not deleted but removed from the list of file systems in the namespace.

- Modify the name of the file system mapping.

You can also modify the name of the file system mapping in a namespace.

 **Note** If file systems in a namespace are in different mount modes, modification of the mapping name may cause connection failure.

- View the details of a unified namespace.

The details of a namespace are divided into three types:

- Properties
- Mount target list

 **Note** You can create or delete mount targets.

- File system list

 **Note** You can add or remove file systems.

Cross-domain mount orchestration

In traditional solutions, you can add file systems to a namespace only when the file systems and the namespace are in the same region. To resolve the issue, NAS provides the cross-domain mount orchestration feature. To use this feature, perform the following operations:

- Create a unified namespace and add mount targets.
- Map the file systems to the local directory tree of a Network File System (NFS) client by using the feature.
- After the orchestration is complete, specify the root directory to generate an automatic mount script.

 **Notice** The added mount targets or attached directories cannot be modified after they are specified. You can remove the mount targets. However, if you add the mount targets again, you can only attach them to the original root directory.

An NFS client runs the automatic mount script to mount the domain names to the local directory. This allows NFS to access file systems across regions. The local directory for each unified namespace is: `<user-defined Root Directory>/<mapping path for a namespace>` .

Different regions have different VPCs. To enable the cross-domain mount orchestration feature, you must establish connections between different VPCs. In the following example, two VPCs in two regions are used to demonstrate how to establish VPC connections and how to mount file systems on an Elastic Compute Service (ECS) instance.

1. Create two VPCs. [Log on to the Apsara File Storage NAS console](#). In the top navigation bar, choose **Products > Networking > Virtual Private Cloud** to go to the VPC console. Create VPCs for Region 1 and Region 2. Configure the following CIDR blocks:
 - Region 1: `192.168.1.0/24` (skvpc1)
 - Region 2: `192.168.2.0/24` (skvpc2)
2. Configure a peering connection across regions. Configure a peering connection between the VPCs of the two regions.

Configure a peering connection in Region 1 to connect skvpc1 and skvpc2.

- i. Configure the route table for skvpc1.

Send the CIDR block `192.168.2.0/24` to skvpc1 and add the CIDR block to the route table of skvpc1.

- a. In the left-side navigation pane, choose **VPC > Route Tables**. On the page that appears, find skvpc1 and click **Manage**.
- b. Click **Add Route Entry** and set the required parameters. Click **Create VPC-to-VPC Connection** to configure a peering connection between skvpc1 and skvpc2.

-  **Note**
- **Destination CIDR Block** must be the same as the CIDR block of skvpc2.
 - Select **Router Interface (To VPC)** from the **Next Hop Type** drop-down list.
 - If you configure a peering connection across regions for the first time, no VPCs are available.

- ii. Create a VPC-to-VPC connection

Click **Create VPC-to-VPC Connection** to go to the **Create Peering Connection** page. Specify the source VPC ID, destination VPC ID, and bandwidth based on your business requirements.

After the VPC-to-VPC connection is created, the **Express Connect** page appears. If the initiator and acceptor are in the **Activated** state, the connection between skvpc1 and skvpc2 is established.

- iii. View VPC-to-VPC connection.

Return to the **Add Route Entry** page. Click **Refresh**. The VPC-to-VPC connection that you have created appears in the VPC list.

Use the preceding procedure to configure a peering connection in Region 2 to connect skvpc1 and skvpc2.

Select the VPC-to-VPC connection from the VPC list and click **OK** to add a route entry.

When you add a route entry for Region 2, you can select the VPC-to-VPC connection from the VPC list. `vpc-6b xxx42t` is the ID of `skvpc1`.

3. Add rules for NAS access groups. In the top navigation bar, choose **Products > Storage > Apsara File Storage NAS**. In the left-side navigation pane, choose **File System > Access Group > NasTimor**, and click **Management Rules**. On the page that appears, click **Add Rules**. In the dialog box that appears, set the authorized address, read/write permissions, user permissions, and priority. For example, add the IP address segment of `skvpc2`.
4. Create a mount target. In Region 2, you can create a mount target for `skvpc2`. The mount target is used to mount file systems on a specified namespace. In Region 1, you can create a mount target for `skvpc1`. The mount target is used to mount file systems on a specified namespace.
5. Create an ECS instance. For example, if you select `skvpc2` when you create an ECS instance, you can mount file systems on the ECS instance. The following figure shows how to select `skvpc2` when you create an ECS instance.

Note In most cases, each ECS instance uses an independent VPC. You do not need to establish connections between VPCs that are used by mount targets of a namespace in two regions. Instead, you must establish connections between the VPC that is used by mount targets of the namespace and of the VPC that is used by the ECS instance.

7.9. Manage the file lifecycle

This topic describes how to use the Apsara File Storage NAS console to manage the file lifecycle, configure lifecycle management policies, and store cold data as Infrequent Access (IA) data. In the NAS console, you can create, view, and modify lifecycle management policies. You can also query the usage of primary storage and IA storage.

Prerequisites

A Network File System (NFS) file system of the NAS Capacity or NAS Performance type is mounted on an Elastic Compute Service (ECS) instance. For more information, see [Mount an NFS file system](#).

Context

NAS provides the lifecycle management feature that allows you to manage cold data. You can configure lifecycle management policies to transfer infrequently accessed data to a specified OSS bucket. This allows you to reduce the storage cost. You can still access the data that is stored in the OSS bucket.

Limits

The lifecycle management feature supports only NFS file systems. Server Message Block (SMB) file systems or file systems whose data is encrypted are not supported.

Additional considerations

The lifecycle management feature allows you to transfer cold data to a specified OSS bucket. You must be the owner of the specified file system and OSS bucket.

- You cannot delete the OSS bucket before you transfer cold data to the original file system. If you delete the file system, data may be lost and NAS clusters may be affected.
- You cannot revoke Resource Access Management (RAM) permissions on NAS.
- We recommend that you minimize the permissions of OSS bucket to prevent data leakage.
- We recommend that you use an independent OSS bucket for NAS. This reduces the risk of deleting cold data by accident if you store the cold data with other business data in the same OSS bucket.

Preparations

1. Authorize NAS by using RAM.

You must grant permissions to NAS before NAS can access the specified OSS bucket to read and write data.

 **Notice** When you create a RAM role, the organization that you specify is the department where the file system resides, for example, bms.

You need to grant permissions to NAS only once for an organization.

- i. Log on to the [Apsara Stack Cloud Management \(ASCM\) console](#).
 - ii. Choose **Configurations > RAM Authorization > RAM Roles > Create RAM Role** to activate OSS authorization for the user.
 - iii. View the authorization status of bms in the role list to confirm that `AliyunNASTieringRole` is on the list.
2. Create an OSS bucket. You can use an existing OSS bucket or create an OSS bucket. You can set different OSS buckets for each file system.

 **Note** If you use an existing OSS bucket, note the following information:

- The bucket is in the first cluster in the region. The domain name of the first cluster starts with `oss-` whereas the names of the other clusters start with `oss.xxxx`.
- Make sure that the bucket and the file system belong to the same department.

Log on to the [ASCM console](#). In the top navigation bar, choose **Products > Object Storage Service** to go to the OSS console. In the left-side navigation pane, click the **add** icon. Create an OSS bucket for the RAM user.

 **Note** When you create an OSS bucket, note the following information:

- Select **standard** from the storage class drop-down list, and the first cluster from the Cluster drop-down list. The domain name of the first cluster starts with `oss-` whereas the names of the other clusters start with `oss.xxxx`.
- Make sure that the bucket and the file system belong to the same department.

Procedure

After the preparations are complete, you can use the lifecycle management feature in the NAS console. Make sure that an NFS file system of the NAS Capacity or NAS Performance type is mounted on an ECS instance. For more information, see [Mount an NFS file system](#).

1. Create a lifecycle management policy
 - i. Log on to the [ASCM console](#). In the top navigation bar, choose **Products > Apsara File Storage NAS**.
 - ii. In the left-side navigation pane, click **Life Cycle Management**. On the page that appears, click **Create Policy** to create a lifecycle management policy.

After you create a lifecycle management policy for a specific directory of the file system, NAS transfers cold data to the specified OSS bucket. You can set the following parameters:

- **Policy Name:** Specify a unique policy name.
- **File System:** Specify a file system.
- **Directory Path:** Specify a directory path on the instance.
You can enter a forward slash (/) to indicate the root directory.
If you select **Recursive Subdirectory**, all subdirectories in the directory are recurred.
- **Management Rules:** Specify a rule to manage files that have not been accessed more than 14, 30, 60, or 90 days.
- **OSS Bucket:** Specify the OSS bucket to which cold data is transferred.

 **Note** Cold data in a file system must be transferred to an OSS bucket.

Only the OSS buckets of the organization where the file system resides are displayed in the list.

2. View the lifecycle management policies

Log on to the ASCM console. In the left-side navigation pane, click **Life Cycle Management**. On the page that appears, you can view the lifecycle management policies. You can also screen the policies by file system ID.

3. Modify a lifecycle management policy

Log on to the ASCM console. In the left-side navigation pane, click **Life Cycle Management**. On the page that appears, you can modify the lifecycle management policies. You can set the following parameters:

- Recursive Subdirectory
- Management Rules

4. Query the usage of primary storage and IA storage

You can query the primary storage and IA storage usage of the file systems that have lifecycle management policies configured. Log on to the ASCM console. In the left-side navigation pane, click **File System List**.

7.10. Directory-level ACLs that grant the read and write access

7.10.1. Overview

Apsara File Storage NAS supports NFSv4 access control lists (ACLs) and Portable Operating System Interface (POSIX) ACLs. This topic describes POSIX ACLs and NFSv4 ACLs. It also lists precautions for using these ACLs.

Access control and user management are important for enterprise-level users who want to share files between different users and groups by using a shared file system. To control access to different files and directories, you can grant users and groups different types of access. NAS provides Network File System (NFS) ACLs to allow you to meet specific requirements. An ACL consists of one or more access control entries (ACEs) that each grant a user or group one or more permissions to access a file or directory.

The NFSv3 protocol includes the extended support for POSIX ACLs. POSIX ACLs extend the support for access control over file mode creation masks. You can grant permissions to specific users and groups besides users of the owner, group, and other classes. Permissions can also be inherited from parent objects. For more information, see [acl - Linux man page](#).

The NFSv4 protocol includes extended support for NFSv4 ACLs that provide more fine-grained access control than POSIX ACLs do. For more information, see [nfs4_acl - Linux man page](#).

You can mount an NFSv3 file system that has NFSv4 ACLs applied. These NFSv4 ACLs will then be converted into POSIX ACLs. You can also mount an NFSv4 file system that has POSIX ACLs applied. These POSIX ACLs will then be converted into NFSv4 ACLs. If you use NFS ACLs, we recommend that you mount NFSv4 file systems and control access by using NFSv4 ACLs rather than file mode creation masks or POSIX ACLs. This is because: NFSv4 ACLs and POSIX ACLs are not fully compatible. The interaction between ACLs and file mode creation masks is not in an ideal state. The file systems that are mounted by using the NFSv3 do not support locks. For more information about NFS ACL features, see [Features](#).

Precautions for using POSIX ACLs

- We recommend that you use the default inheritance method that allows a subdirectory or file to inherit the same ACL from the parent directory. This allows you to avoid configuring another ACL when you create a new file or subdirectory in the parent directory.
- We recommend that you retain a minimum number of ACEs because a file system needs to scan all ACEs each time it performs permission verification. Abuse of ACLs may diminish the performance of file systems.
- Use caution when you configure ACLs by using the recursive method (`setfacl -R`). Large amounts of metadata are produced when you perform a recursive operation on a directory that contains a large number of files and subdirectories. This may affect your businesses.
- Before you configure ACLs, we recommend that you manage groups and related permissions. For example,

you can add a user to one or more groups. If you want to add, remove, or modify permissions for a user, we recommend that you move the user to a group that has the required permissions. You do not need to modify the ACL of a group as long as the structure of groups remains unchanged. We recommend that you configure ACLs for groups rather than single users. This provides a simple and effective time-saving method to control access and ensure the better organization of permissions.

- You can apply a POSIX ACL to multiple objects that resides on different clients. In such cases, you must ensure that the ACL you apply to each object is the same. Apsara File Storage NAS stores user IDs (UIDs) and group IDs (GIDs) at the backend. You must ensure that the mappings between a username or group name and a UID or GID are the same.
- We recommend that you grant the least permissions to the other class because all users have the permissions that are granted to the other class. A potential security vulnerability may be exposed if the other class has more permissions than any ACE.
- We recommended that you configure the least permissions for the other class. Before creating files or directories, you can use the `umask 777` command to configure the file mode creation mask. This command sets the file mode creation mask to 000 when the mask is used as a parameter to create a new file or directory. This ensures that the newly created file or directory has the least permissions. For more information, see [umask and the default file mode creation mask](#).
- After you enable POSIX ACLs, the semantics of the other class for the POSIX ACL are equal to the semantics of the `EVERYONE@` principal. The semantics of the other class for the file mode creation mask are also equal to the semantics of the `EVERYONE@` principal. When a system performs permission verification, the system treats the other class the same as the `EVERYONE@` principal.

Precautions for using NFSv4 ACLs

- Use UIDs or GIDs such as UID 1001 to configure ACLs.
- We recommend that you do not configure the file mode creation mask after you configure an NFSv4 ACL.
- The `nfs4_setfacl` command provides `-a`, `-x`, `-m`, and other options. You can use these options to add, remove, or modify ACEs. However, we recommend that you use `nfs4_setfacl -e <file>` the command to edit an ACL in an interactive mode.
- We recommend that you configure the least permissions for the `EVERYONE@` principal because NFSv4 ACLs only support allow rather than deny ACEs. A potential security vulnerability may be exposed if the `EVERYONE@` principal has more permissions than other ACEs.
- NFSv4 ACLs have fine-grained permissions. In most cases, it is unnecessary to subdivide permissions at such a fine-grained level. For example, if you have the write access (`w`) to a file but do not have the append-only (`a`) access, an error may occur when you write data to the file. The same issue occurs for a directory. To avoid unexpected permission errors, we recommend that you specify a capital `w` (`W`) as a parameter when you use the `nfs4_setfacl` command to configure an ACL. The `nfs4_setfacl` command converts `W` to a full write access permission. For a file, `W` is expanded to `wadT`. For a directory, `W` is expanded to `wadTD`.
- We recommend that you use the default inheritance method that allows a subdirectory or file to inherit the same ACL from the parent directory. This allows you to avoid configuring another ACL when you create a new file or subdirectory in the parent directory.
- We recommend that you retain a minimum number of ACEs because a file system needs to scan all ACEs each time it performs permission verification. Abuse of ACLs may diminish the performance of file systems.
- Use caution when you configure ACLs by using the recursive method (`nfs4_setfacl -R`). Large amounts of metadata are generated when you perform a recursive operation on a directory that contains a large number of files and subdirectories. This may affect your businesses.
- Before you configure ACLs, we recommend that you manage groups and related permissions. For example, you can add a user to one or more groups. If you want to add, remove, or modify permissions for a user, we recommend that you move the user to a group that has the required permissions. You do not need to modify the ACL of a group as long as the structure of groups remains unchanged. We recommend that you configure ACLs for groups rather than single users. This provides a simple and effective time-saving method to control access and ensure the better organization of permissions.

7.10.2. Features

This topic describes the features of NFSv4 access control lists (ACLs) and POSIX ACLs.

Features of Apsara File Storage NAS NFSv4 ACLs

- Only access control entries (ACEs) of the allow type are supported. ACEs of the following types are not supported: deny, audit, and alarm.

Deny ACEs increase the complexity of access control. In most cases, complexity leads to confusion and increases potential security risks. As agreed by the industry, we recommend that you avoid using deny ACEs. For more information about why deny ACEs are not recommended, see [FAQ](#).

Audit and alarm ACEs are not available for NFS file systems. Instead, you can audit file systems and configure alerts based on auditing results in the NAS console.

- If no ACL is specified for a file or a directory, the default ACL that corresponds to the predefined file mode creation mask is applied.

```
touch file
```

```
[root@vbox test]# ls -l file
-rw-r--r--. 1 root root 0 May  6 14:27 file
```

```
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwatTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
```

- An ACL is an ordered list that contains and deduplicates ACEs. This scheme ensures that permissions defined in an ACL are clear and informative.

If you apply both a new ACE and an existing ACE to the same object and the existing ACE is inherited from the parent object, the permissions of the new ACE override the permissions of the existing ACE. For example:

- In most cases, ACEs that include the following principals are queued in sequence at the beginning of an ACL: OWNER@, GROUP@, and EVERYONE@. These ACEs take precedence over other ACEs.

```
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
A::1001:rwaxTnNcCy
```

- Add an ACE of the read and write permissions to the following ACL for a user principal named 1009. The ACE is placed after the ACE that is defined for a user principal named 1001 based on the predefined order.

```
[root@vbox test]# nfs4_setfacl -a A::1009:X file
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxTnNcCy
A::1009:xtcy
```

- Add a new ACE that includes the execute permission to the ACL for the user principal named 1009. The system automatically merges the execute permission into the existing ACE for the 1009 user principal.

```
[root@vbox test]# nfs4_setfacl -a A::1009:W file
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxTNcCy
A::1009:waxtTncCy
```

- Add the f and d inheritance flags to an ACE that includes a user principal named 1009. Then, the system splits the ACE into two ACEs. One ACE has an extra inheritance flag named i specified, which indicates an inherit-only ACE. The other ACE only applies to the file object without inheritance flags. If the inheritance type of an existing ACE matches the type for one of the two ACEs, the system combines the existing ACE with the ACE out of the two ACEs. The two matching ACEs are combined into one ACE.

```
[root@vbox test]# nfs4_setfacl -a A:fd:1009:R file
[root@vbox test]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxTNcCy
A::1009:rwaxTNcCy
A:fdi:1009:r
```

- All ACEs can be inherited.
 - For example, the OWNER@ principal has the write access, the GROUP@ principal has the read access, and the EVERYONE@ has no access to the dir directory.

```
[root@vbox nfs]# nfs4_getfacl dir
# file: dir
A::OWNER@:rwaDxtTnNcCy
A::GROUP@:rxscy
A::EVERYONE@:tncy
```

- ii. Add an ACE that grants a user principal named 1000 the read, write, and execute access to the dir directory. The f and d inheritance flags are also specified for the ACE.

```
[root@vbox nfs]# nfs4_setfacl -a A:fd:1000:rw dir
[root@vbox nfs]# nfs4_getfacl dir
# file: dir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rxscy
A::EVERYONE@:tcy
A::1000:rw
A:fdi:1000:rw
```

- iii. When you create a file or subdirectory in the dir directory, the file or the subdirectory automatically inherits all ACEs from the dir directory.

```
[root@vbox nfs]# touch dir/file
[root@vbox nfs]# nfs4_getfacl dir/file
# file: dir/file
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rwx
```

```
[root@vbox nfs]# mkdir dir/subdir
[root@vbox nfs]# nfs4_getfacl dir/subdir
# file: dir/subdir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rwaDxtcy
A::EVERYONE@:rwaDxtcy
A:fdi:1000:rwx
```

Note

- We recommend that you grant the least privileges to the `EVERYONE@` principal. Before you perform the following steps, we recommend that you run the `umask 777` command. This command ensures that no access to a file or directory is granted when the file or directory is created. For more information, see [Why doesn't umask change execute permissions on files?](#)
- When Linux calls functions to create files or directory, the predefined file mode creation mask is used as a request parameter. You can obtain the final ACL for a child object from the overlap of the inherited ACL (parent to child) and the file mode creation mask, as specified in the [RFC7530](#) standard. When you modify the group bits of a file mode creation mask based on the standard, permissions included in an ACL for each group must be less than or equal to permissions defined in group bits. However, this scheme results in an invalid inheritance for groups. For example, you create a file and the file attempts to inherit `A:RWX` from a parent object. However, the predefined file mode creation mask sets the group bits to `R`. The final permission for the file becomes `A:R`. In actual practice, we recommend that you only modify file mode creation masks for ACLs that include the following principals: `OWNER@`, `GROUP@`, and `EVERYONE@`. This prevents against potential issues and ensures that semantics are clear. To remove permissions for a group, we recommend that you remove the ACE that relates to the group.

- You need to manage mappings between usernames or group names and user IDs (UIDs) or group IDs (GIDs) across multiple independent instances.

NAS NFS adopts IP security groups rather than usernames to authenticate users. When you configure NFSv4 ACLs, UID or GID that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the `/etc/passwd` file and converts UID or GID into usernames or group names. You need to manage mappings between usernames or group names and UID or GID across multiple instances. You must ensure a username or group name is mapped to its UID or GID.

- NFSv4 ACLs can be printed by using extended attributes.

```
[root@vbox nfs]# getfattr -n system.nfs4_acl file
# file: file
system.nfs4_acl=0sAAAABgAAAAAAAAAAAAABYBhwAAAAZPV05FUKAAAAAAAAAAAAAAAAABIAhwAAAAZHUK9VUEAAAAA
AAAAAAAAABIAhwAAAAIFVkvS WU9ORUAAAAAAAAAAAAAAAAAAAAEAAAEEMT AwMAAAAAAAAAALAAAAAwAAAAQ
xMDAwAAAAAAAAAAEAAfGgQAAAABTEwMDAxAAAA
```

- Tools such as `cp` are supported for migrating NFSv4 ACLs.

NAS allows you to migrate NFSv4 ACLs by using the `cp`, `tar`, and `rsync` tools. For more information, see [How to preserve NFS v4 ACLs via extended attributes when copying file](#).

The following `cp --preserve=xattr file1 file2` command makes a copy of the file1 file as the file2 file while making a copy of the ACL of the file1 file for the file2 file. The `cp -ar dir1 dir2` command makes a copy of the dir1 directory as the dir2 directory while making a copy of the ACL of the dir1 directory for the dir2 directory.

 **Note** You may fail to migrate NFSv4 ACLs if the version of the `rsync` tool is earlier than 3.1.2.

```
[root@vbox nfs]# nfs4_getfacl file1
# file: file1
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
[root@vbox nfs]# cp --preserve=xattr file1 file2
```

```
[root@vbox nfs]# nfs4_getfacl file2
# file: file2
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
[root@vbox nfs]# cp -ar dir1 dir2
```

- Interaction between NFSv4 ACLs and file mode creation masks is supported. The modification for the ACL of an object may change the file mode creation mask of the object. The modification for the file mode creation mask of an object may change the ACL of the object.

For example, the file mode creation mask of the file object is 0666.

```
[root@vbox nfs]# ls -l file
-rw-rw-rw-. 1 root root 0 May 3 2019 file
[root@vbox nfs]# nfs4_getfacl file
# file: file
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
```

- If you add the execute permission to the file mode creation mask by modifying the owner bits, the execute permission is also added to the ACE that includes the OWNER@ principal.

```
[root@vbox nfs]# chmod u+x file
[root@vbox nfs]# ls -l file
-rwxrw-rw-. 1 root root 0 May 3 2019 file
[root@vbox nfs]# nfs4_getfacl file
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
```

- If you add the execute permission to an ACE that includes the GROUP@ principal, the execute permission is also added to the related file mode creation mask.

```
[root@vbox nfs]# nfs4_setfacl -a A::GROUP@:x file
[root@vbox nfs]# ls -l file
-rwxrwxrw-. 1 root root 0 May 3 2019 file
```

 **Note**

- In the interaction between ACLs and file mode creation masks, the EVERYONE@ principal is equal to the others class. When you modify the others class, the change also applies to the EVERYONE@ principal. This operation results in a slight impact on the semantics of permissions. For example, the current file mode creation mask is 177. After you run the `chmod o+r` command, all users that include the file owner and group members have the read permission. This occurs because the read permission is added to the related ACE that includes the EVERYONE@ principal. If no change is applied to the default file mode creation mask, the owner and group classes still have no read permission after you run the `chmod o+r` command.
- If no change is applied to NFSv4 ACLs, the others class of the file mode creation mask keeps the same semantics. If an NFSv4 ACL is changed, the semantics of the others class change to the semantics of the EVERYONE@ principal and the latest semantics remain. We recommend that you do not use file mode creation masks after using NFSv4 ACLs.

- Interaction between NFSv4 ACLs and POSIX ACLs is supported.

You can mount NFSv3 file systems that have NFSv4 ACLs applied. These NFSv4 ACLs will then be converted into POSIX ACLs. You can also mount NFSv4 file systems that have POSIX ACLs applied. These POSIX ACLs will then be converted into NFSv4 ACLs.

 **Note** The semantics of POSIX ACLs are different from the semantics of NFSv4 ACLs. For example, the inheritance rules that apply to POSIX ACLs do not differentiate files and directories. NFSv4 ACLs have more permissions than POSIX ACLs, which have only read, write, and execute permissions. We recommend that you use either NFSv4 ACLs or POSIX ACLs to prevent against potential issues.

For example, you configure an NFSv4 ACL for the dir0 directory. The permissions are listed as follows.

```
[root@vbox test] sudo nfs4_getfacl dir0
A::OWNER@:tTnNcCy
A::GROUP@:tnCy
A::EVERYONE@:tnCy
A:fdi:EVERYONE@:tnCy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tnCy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

You configure a POSIX ACL for the dir0 directory. The permissions are listed as follows.

```
[root@vbox test] sudo getfacl dir0
user::---
group::---
group:players:r-x
group:adminis:rwX
mask::rwX
other::---
default:user::---
default:group::---
default:group:players:r-x
default:group:adminis:rwX
default:mask::rwX
default:other::---
```

For example, you configure an NFSv4 ACL for the dir0/file file. The permissions are listed as follows.

```
[root@vbox test] sudo nfs4_getfacl dir0/file
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:g:19064:rxtncy
A:g:19065:rwaxTnNcCy
```

For example, you configure a POSIX ACL for the dir0/file file. The permissions are listed as follows.

```
[root@vbox test] sudo getfacl dir0/file
user::---
group::---
group:players:r-x
group:adminis:rwX
mask::rwX
other::---
```

- The number of NFSv4 ACLs is limited.

NAS supports a maximum of 100,000 ACLs that are different from one another in each file system. Each ACL contains a maximum of 500 ACEs.

 **Note** We recommend that you do not abuse ACLs and ACEs. This reduces the time and resources consumed for verifying permissions.

Features of NAS POSIX ACLs

- Permissions that are specified for the other class apply to all users.

Everyone includes the owner, group, and users that are related to each ACE. The other class is equal to the EVERYONE@ principal of an NFSv4 ACL.

 **Note** We recommend that you grant the least permissions to the other class in all cases.

For example, the following ACL is configured for the *myfile* file. Although the ACE contains a user named alice who does not have the write permission, the write permission propagates to the ACE because the permission is specified for the other class.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::rw-
user:alice:r--
group::r--
mask::r--
other::rw-
```

- Permissions that are configured by ACLs will not be changed after you run the `chmod` command.

 **Note** We recommend that you avoid modifying the file mode creation mask of a file that has a POSIX ACL applied. You can configure permissions for the file by modifying the POSIX ACL.

- For example, an ACE that grants the players group the read and write access to the *myfile* file.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::rw-
user:player:rw-
group::rw-
group:players:rw-
mask::rw-
other::---
```

- The `chmod g-w myfile` or `chmod u-w myfile` command does not change the permissions that are granted to the player user and the players group, which is different from the **POSIX ACL standard**. However, this ensures that permissions that are granted by POSIX ACLs to non-reserved users are the same after you modify permissions by using file mode creation masks. The non-reserved users include all users except for the users of the owner, group, and other classes.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::r--
user:player:rw-
group::r--
group:players:rw-
mask::rw-
other::---
```

- If the execute permission is not granted to the group and other classes of an ACL, the ACL has no execute permission.

The rule is predefined in Linux. The execute action is allowed by the backend of NAS. However, to make the execute permission in the ACL effective, you must grant the execute permission to the group or other class.

For example, if the group and other classes do not have the execute access to the *myfile* file, the player user cannot execute the file.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::rw-
user:player:r-x
group::r--
mask::r-x
other::r--
```

If you grant the execute permission to the group class, the execute permission also propagates to the player user.

```
[root@vbox 3]# getfacl myfile
# file: myfile
# owner: root
# group: root
user::rw-
user:player:r-x
group::r-x
mask::r-x
other::r--
```

- If you configure inheritable NFSv4 ACLs for directories, these settings may not conform to the POSIX ACL standard when these directories reside in NFSv3 file systems.

Inheritance rules that apply to files are different from those that apply to directories in NFSv4 ACLs. The same inheritance rules apply to both files and directories in POSIX ACLs.

 **Note** We recommend that you apply either NFS4 ACLs or POSIX ACLs to an NFS file system to prevent against potential issues.

- File mode creation masks cannot be modified.

The file mode creation mask of a POSIX ACL is yielded by the combination and interaction of permissions from all users and groups. The mask has no practical meaning and cannot be changed.

- You need to manage mappings between usernames or group names and user IDs (UIDs) or group IDs (GIDs) across multiple instances.

Apsara File Storage NAS NFS adopts IP security groups rather than usernames to authenticate users. When you configure POSIX ACLs, UIDs or GIDs that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the */etc/passwd* file and converts UIDs or GIDs into actual usernames or group names. You need to manage mappings between usernames or group names and UIDs or GIDs across multiple instances. You must ensure a username or group name is mapped to its related UID or GID.

- POSIX ACLs can be printed by using extended attributes.

```
[root@vbox nfs]# getfattr -n system.posix_acl_access file
# file: file
system.posix_acl_access=0sAgAAAAEAAAD/////AgAFACAEAAAEEAAA/////xAABQD/////IAABAP/////8=
```

- POSIX ACLs can be migrated by using tools such as cp.

NAS allows you to migrate POSIX ACLs by using the `cp`, `tar`, and `rsync` tools. For more information, see [How to preserve NFS v4 ACLs via extended attributes when copying file](#).

The following `cp --preserve=xattr file1 file2` command makes a copy of the file1 file as the file2 file while making a copy of the ACL of the file1 file for the file2 file. The `cp -ar dir1 dir2` command makes a copy of the dir1 directory as the dir2 directory while making a copy of the ACL of the dir1 directory for the dir2 directory.

 **Note** You may fail to migrate POSIX ACLs if the version of the `rsync` tool is earlier than 3.1.2.

```
[root@vbox nfs]# getfacl file1
user::---
user:player:r-x
group::---
mask::r-x
other::--x
[root@vbox nfs]# cp --preserve=xattr file1 file2
```

```
[root@vbox nfs]# getfacl file2
# file: file2
user::---
user:player:r-x
group::---
mask::r-x
other::--x
[root@vbox nfs]# cp -ar dir1 dir2
```

- The number of POSIX ACLs is limited.

NAS supports a maximum of 100,000 ACLs that are different from one another in each file system. Each ACL contains a maximum of 500 ACEs.

 **Note** We recommend that you do not abuse ACLs and ACEs. This reduces the time and resources consumed for verifying permissions.

FAQ

Why are deny ACEs not supported?

- The position of an ACE that resides in an ACL is important.

The sequence for ACEs that reside in an NFSv4 ACL is random. A deny ACE may be placed in any position of an NFSv4 ACL. For example, an ACL contains two ACEs: `A::Alice:r` and `D::Alice:r`. The position of the ACEs determines whether the user named Alice has the write permission.

 **Note** When you configure an ACL, you must consider the position of each ACE.

- The number of ACEs in an ACL experiences a sharp increase.

You may have difficulties to combine and deduplicate ACEs in an ACL because the sequencing for ACEs is not mandatory. The number of ACEs may increase up to tens or hundreds over a long period of time. To manage the final permissions that are produced by these ACEs, you need to check each ACE. The process to check is strenuous and time-consuming.

- The interactions between file mode creation masks and ACLs become more complex after deny ACEs are applied because deny features do not exist in file mode creation masks.

- If deny ACEs are available, you may need to add several ACEs to an ACL when the file mode creation mask is changed. For example, if you change the file mode creation mask to `-rw-rw-rw`, you need to add the following ACEs to an ACL. You must add the ACEs in sequence at the beginning of the ACL.

```
A::OWNER@:rw
D::OWNER@:x
A::GROUP@:rw
D::GROUP@:x
A::EVERYONE@:rw
D::EVERYONE@:x
```

- If deny ACEs are unavailable, you can sequence and deduplicate ACEs. You do not need to differentiate the `EVERYONE@` principal and the other class. You can modify an ACL with ease when the file mode creation mask is changed. In such cases, you only need to find ACEs that contain the `OWNER@`, `GROUP@`, and `EVERYONE@` principals and modify these ACEs as follows.

```
A::OWNER@:rw
A::GROUP@:rw
A::EVERYONE@:rw
```

- Conversions between NFSv4 ACLs and POSIX ACLs are not supported in some cases.
 - POSIX ACLs do not support deny ACEs. If deny ACEs are included in an NFSv4 ACL, you cannot convert the ACL into a POSIX ACL.

7.10.3. Use POSIX ACLs to control access

This topic describes how to configure Portable Operating System Interface (POSIX) access control lists (ACLs). You can use POSIX ACLs to control access to files and directories that reside in an NFSv3 file system.

Prerequisites

An NFSv3 file system is mounted. For more information, see [Mount an NFS file system](#).

Commands

Before you configure POSIX ACLs, we recommend that you familiarize yourself with the related commands.

Command	Description
<code>getfacl <filename></code>	Shows the ACL that applies to the specified file.
<code>setfacl -m g:w <filename></code>	Grants the owning group the write access.
<code>setfacl -m u:player:w <filename></code>	Grants the player user the write access.
<code>setfacl -m g:players:rwx <filename></code>	Grants the players group the read, write, and execute access.
<code>setfacl -x g:players <filename></code>	Removes permissions from the players group
<code>getfacl file1 setfacl --set-file=- file2</code>	Copies the ACL for the <i>file1</i> file to the <i>file2</i> file.
<code>setfacl -b file1</code>	Removes all extended ACEs from the <i>file1</i> file. The base ACEs of the owner, group, and others are retained.
<code>setfacl -k file1</code>	Removes all default ACEs from the <i>file1</i> file.
<code>nfs4_setfacl -R -m g:players:rw dir</code>	Grants the players group the read and write access to files and subdirectories in the <i>dir</i> directory.

Command	Description
<code>setfacl -d -m g:players:rw dir1</code>	Grants the players group the read and write access to the newly created files and subdirectories in the <i>dir1</i> directory.

Procedure

To control access to files and directories by configuring NFS ACLs, follow these steps.

1. Create users and groups.

In this example, the following users are created: `player`, `admini`, and `anonym`. The following groups are created: `players` and `adminis`. The `player` user is added to the `players` group and the `admini` user is added to the `adminis` group.

```
sudo useradd player
sudo groupadd players
sudo usermod -g players player
sudo useradd admini
sudo groupadd adminis
sudo usermod -g adminis admini
sudo useradd anonym
```

2. Configure POSIX ACLs to control access to files and directories.

Use the following commands to complete the operations: create a directory named *dir0* and grant the `players` group the read-only access, the `adminis` group the read, write, and execute permissions, and the others class no access to all the files in the *dir0* directory.

```
sudo umask 777
sudo mkdir dir0
sudo setfacl -m g:players:r-x dir0
sudo setfacl -m g:adminis:rwX dir0
sudo setfacl -m u:--- dir0
sudo setfacl -m g:--x dir0
sudo setfacl -m o:--- dir0
sudo setfacl -d -m g:players:r-x dir0
sudo setfacl -d -m g:adminis:rwX dir0
sudo setfacl -d -m u:--- dir0
sudo setfacl -d -m g:--x dir0
sudo setfacl -d -m o:--- dir0
```

Use the `sudo getfacl dir0` command to verify the result after the configuration is complete.

```
# file: dir0
# owner: root
# group: root
user::---
group::--x
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
default:user::---
default:group::--x
default:group:players:r-x
default:group:adminis:rwx
default:mask::rwx
default:other::---
```

3. Verify the ACL configuration.

- i. Verify that the admini user has read and write access to the dir0/file file.

```
[root@vbox test] sudo su admini -c 'touch dir0/file'
[root@vbox test] sudo su admini -c 'echo 123 > dir0/file'
```

- ii. Use the following command to verify the read-only access of the player user.

```
[root@vbox test] sudo su player -c 'touch dir0/file'
touch: cannot touch 'dir0/file': Permission denied
[root@vbox test] sudo su player -c 'cat dir0/file'
123
[root@vbox test] sudo su player -c 'echo 456 >> dir0/file'
bash: dir0/file: Permission denied
[root@vbox test] sudo su player -c 'getfacl dir0/file'
# file: dir0/file
# owner: admini
# group: adminis
user::---
group::---
group:players:r-x
group:adminis:rwx
mask::rwx
other::---
```

- iii. Verify that the anonym user does not have access to the dir0/file file.

```
[root@vbox test] sudo su anonym -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su anonym -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su anonym -c 'getfacl dir0/file'
getfacl: dir0/file: Permission denied
```

Related operations

If you want to remove user permissions, use the following method.

When you use NFSv4 ACLs, we recommend that you sort each user into different groups. This allows you to configure permissions for a group rather than a separate user. To disable access to an object from a user, you can remove the user from a group that has access to the object. For example, the following commands remove the admini user from the adminis group and add the user to the adminis2 group.

```
[root@vbox test] sudo groupadd adminis2
[root@vbox test] sudo usermod -g adminis2 admini
[root@vbox test] id admini
uid=1057(admini) gid=1057(admini) groups=1061(adminis2)
[root@vbox test] sudo su admini -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su admini -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su admini -c 'getfacl dir0/file'
getfacl: dir0/file: Permission denied
```

7.10.4. Use NFSv4 ACLs to control access

This topic describes how to configure NFSv4 access control lists (ACLs) and apply these ACLs to NFSv4 file systems to control access to files and directories.

Prerequisites

An NFSv4 file system is mounted. For more information, see [Mount an NFS file system](#).

Context

You can mount an NFSv4 file system on an Elastic Compute Service (ECS) instance that runs Linux and install the Linux-specific `nfs4-acl-tools` tool on the instance. You can use the standard `nfs4_getfacl` and `nfs4_setfacl` tools to configure NFSv4 ACLs after the installation is completed.

Commands

Before you configure NFSv4 ACLs, we recommend that you familiarize yourself with the related commands.

Command	Description
<code>nfs4_getfacl <filename></code>	Views the access permissions for the specified file.
<code>nfs4_setfacl -a A::GROUP@:W <filename></code>	Adds an access control entry (ACE) that grants the GROUP@ principal the write access to the specified file.
<code>nfs4_setfacl -a A::1000:W <filename></code>	Adds an ACE that grants a user principal named 1000 the write access to the specified file.
<code>nfs4_setfacl -a A:g:10001:W <filename></code>	Adds an ACE that grants a group principal named 10001 the write access to the specified file.
<code>nfs4_setfacl -e <filename></code>	Edits an ACL in an interactive mode.
<code>nfs4_getfacl <filename> > saved_acl.txt</code>	Saves a list of permissions for the specified file as a TXT file.

Command	Description
<code>nfs4_setfacl -S saved_acl.txt <filename></code>	Configures permissions for the specified file by using a TXT file that includes a list of ready-made permissions.
<code>nfs4_setfacl -m A::1001:rwaxTNCy A::1001:rxtcy file1</code>	Modifies the permission of an ACE that applies to the <i>file1</i> file.
<code>nfs4_getfacl file1 nfs4_setfacl -S - file2</code>	Copies the permissions for the <i>file1</i> file to the <i>file2</i> file.
<code>nfs4_getfacl file1 grep @ nfs4_setfacl -S - file1</code>	Deletes all ACEs that apply to the <i>file1</i> file except for ACEs that include the following principals: OWNER@, GROUP@, and EVERYONE@.
<code>nfs4_setfacl -R -a A:g:10001:rW dir</code>	Adds an ACE that grants a group principal named 10001 the read and write access to files and subdirectories in the <i>dir</i> directory.
<code>find dir -type f -exec sh -c 'for ace in \$(nfs4_getfacl \{} grep "^A.*\:1005\:"); do nfs4_setfacl -x \$ace \{}; done' \;</code>	Deletes ACEs that grant a user principal named 1005 access to files in the <i>dir</i> directory.
<code>nfs4_setfacl -a A:fdg:10001:rW dir1</code>	Adds an ACE that grants a group principal named 10001 the read and write access to new files and subdirectories in the <i>dir1</i> directory.
<code>nfs4_setfacl -a A:fg:10001:rx dir1</code>	Adds an ACE that grants a group principal named 10001 the read, write, and execute access to all newly created files in the <i>dir1</i> directory.

Procedure

To control access to files and directories by configuring NFSv4 ACLs, follow these steps.

1. Create users and groups.

In this example, the following users are created: `player`, `admini`, and `anonym`. The following groups are created: `players` and `adminis`. The `player` user is added to the `players` group and the `admini` user is added to the `adminis` group.

```
sudo useradd player
sudo groupadd players
sudo usermod -g players player
sudo useradd admini
sudo groupadd adminis
sudo usermod -g adminis admini
sudo useradd anonym
```

2. Install the related tools to configure NFSv4 ACLs.

If you have installed these tools, skip this step.

```
sudo yum -y install nfs4-acl-tools
```

3. Obtain the group IDs of the `players` and `adminis` groups. Open the `/etc/group` file. The group IDs of the `players` and `adminis` groups are displayed as follows:

```
players:x:19064:player
adminis:x:19065:admini
```

4. Configure NFSv4 ACLs for files and directories. Use the following commands to complete the operations: create a directory named *dir0* and add ACEs that grant the players group the read-only access, the adminis group the read, write, and execute access, and other users no access to all the files in the *dir0* directory.

```
sudo umask 777
sudo mkdir dir0
sudo nfs4_setfacl -a A:fdg:19064:RX dir0
sudo nfs4_setfacl -a A:fdg:19065:RWX dir0
sudo nfs4_setfacl -a A:fdg:OWNER@: dir0
sudo nfs4_setfacl -a A:fdg:GROUP@: dir0
sudo nfs4_setfacl -a A:fdg:EVERYONE@: dir0
```

Use the `sudo nfs4_getfacl dir0` command to verify the configuration.

```
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:fdi:EVERYONE@:tncy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tncy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

5. Verify the configuration of the ACL.

- i. Use the following commands to verify the read and write access of the admini user.

```
[root@vbox test] sudo su admini -c 'touch dir0/file'
[root@vbox test] sudo su admini -c 'echo 123 > dir0/file'
```

- ii. Use the following command to verify the read-only access of the player user.

```
[root@vbox test] sudo su player -c 'touch dir0/file'
touch: cannot touch 'dir0/file': Permission denied
[root@vbox test] sudo su player -c 'echo 456 >> dir0/file'
bash: dir0/file: Permission denied
[root@vbox test] sudo su player -c 'cat dir0/file'
123
[root@vbox test] sudo su player -c 'nfs4_getfacl dir0/file'
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:g:19064:rxtncy
A:g:19065:rwaxtTnNcCy
```

- iii. Use the following command to verify that the anonym user does not have access to the /dir0/file file.

```
[root@vbox test] sudo su anonym -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su anonym -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su anonym -c 'nfs4_getfacl dir0/file'
Invalid filename: di
```

Related operations

If you want to remove user permissions, use the following method.

When you use NFSv4 ACLs, we recommend that you sort each user into different groups. This allows you to configure permissions only for a group rather than a separate user. To disable access to an object from a user, you can remove the user from a group that has access to the object. For example, use the following commands to remove the admini user from the adminis group and add the user to the adminis2 group:

```
[root@vbox test] sudo groupadd adminis2
[root@vbox test] sudo usermod -g adminis2 admini
[root@vbox test] id admini
uid=1057(admini) gid=1057(admini) groups=1054(adminis2)
[root@vbox test] sudo su admini -c 'ls dir0'
ls: cannot open directory dir0: Permission denied
[root@vbox test] sudo su admini -c 'cat dir0/file'
cat: dir0/file: Permission denied
[root@vbox test] sudo su admini -c 'nfs4_getfacl dir0/file'
Invalid filename: dir0/file
```

8. Tablestore

8.1. What is Tablestore?

Tablestore is a NoSQL database service independently developed by Alibaba Cloud. Tablestore is a proprietary software program that is certified by the relevant authorities in China. Tablestore is built on the Apsara system of Alibaba Cloud, and can store large amounts of structured data and allow real-time access to these data.

Tablestore provides the following features:

- Offers schema-free data storage. You do not need to define attribute columns before you use them. Table-level changes are not required to add or delete attribute columns. You can configure the time to live (TTL) parameter for a table to manage the lifecycle of data. The expired data is deleted from the table.
- Adopts the triplicate technology to keep three copies of data on three servers across three different racks. A cluster can support single storage type instances (SSD only) or mixed storage type instances (SSD and HDD) to meet different budget and performance requirements.
- Adopts a fully redundant architecture that prevents single points of failure (SPOFs). Tablestore supports smooth online upgrades, hot cluster upgrades, and automatic data migration, which enable you to dynamically add or remove nodes for maintenance without incurring service interruptions. The concurrent read and write throughput and storage capacity can be linearly scaled. Each cluster can have at least 500 hosts.
- Supports highly concurrent read and write operations. Concurrent read and write capabilities can be scaled out as the number of hosts increases. The read and write performance is indirectly related to the amount of data in a single table.
- Supports identity authentication and multi-tenancy. Comprehensive access control and isolation mechanisms are provided to safeguard your data. VPC and access over HTTPS are supported. Provides multiple authentication and authorization mechanisms so that you can define access permissions on individual tables and operations.

8.2. Precautions

Before you use Tablestore, you need to take note of the following precautions and limits.

The following table describes the limits for Tablestore. A part of the limits indicate the maximum allowable values rather than the suggested values. To ensure better performance, set the table scheme and data size in a single row based on actual conditions, and adjust the following configurations.

Item	Limit	Description
The number of instances under an Apsara Stack tenant account	1024	To raise the limit, contact the technical support personnel.
The number of tables in an instance	1024	To raise the limit, contact the technical support personnel.
The length of an instance name	3 to 16 bytes	The instance name can contain uppercase and lowercase letters, digits, and hyphens (-). It must start with a letter and cannot end with a hyphen (-).
The length of a table name	1 to 255 bytes	The table name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_).
The length of a column name	1 to 255 bytes	The column name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_).

Item	Limit	Description
The number of columns in a primary key	1 to 4	A primary key can contain one to four primary key columns.
The size of the value in a string type primary key column	1 KB	The size of the value in a STRING primary key column cannot exceed 1 KB.
The size of the value in a STRING attribute column	2 MB	The size of the value in a STRING attribute column cannot exceed 2 MB.
The size of the value in a BINARY primary key column	1 KB	The size of the value in a BINARY primary key column cannot exceed 1 KB.
The size of the value in a BINARY attribute column	2 MB	The size of the value in a BINARY attribute column cannot exceed 2 MB.
The number of attribute columns in a single row	Unlimited	A single row can contain an unlimited number of attribute columns.
The number of attribute columns written by one request	1,024	During a PutRow, UpdateRow, or BatchWriteRow operation, the number of attribute columns written in a row cannot exceed 1,024.
The data size of a row	Unlimited	The total size of all column names and column values for a row is unlimited.

8.3. Quick start

8.3.1. Log on to the Tablestore console

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

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `http://IP address or domain name of the ASCM console/manage`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.

- In the top navigation bar, choose **Products > TableStore**.

8.3.2. Create instances

An instance is a logical entity in Tablestore and is used to manage tables. An instance is the basic unit of the resource management system of Tablestore. Tablestore implements application access control and resource measurement at the instance level. This topic describes how to create an instance.

Procedure

- Log on to the [Tablestore console](#).
- On the **Overview** tab, click **Create Instance**.

 **Note** You can create different instances to manage the tables for different businesses, or create different instances for development, testing, and production environments of the same business. By default, Tablestore allows you to create up to 1,024 instances and up to 1,024 tables in each instance under an Apsara Stack tenant account.

- On the **Create Tablestore Instance** page, configure the following parameters.

Parameter	Description
Region	Select a region from the drop-down list for the instance.
Organization	Select an organization from the drop-down list for the instance.
Resource Set	Select a resource set from the drop-down list for the instance.
Instance Name	Enter a name for the instance. Instance naming conventions: The name must be 3 to 16 characters in length and can contain only letters, digits, and hyphens (-). It must start with a letter and cannot start with case insensitive string <code>ali</code> or <code>ots</code> .
Description	Enter a description for the instance.
Instance Type	Select an instance type from the drop-down list for the instance. Tablestore provides high-performance instances and capacity instances. The instance types vary based on the type of cluster you deploy.

- Click **Submit**.
- In the **Submitted** dialog box, click **Back to Console**. On the **Overview** tab, you can view the created instance.

After the instance is created, you can perform the following operations on the instance:

- Click the instance name or click **Manage Instance** in the **Actions** column. On the **Instance Management** page, click each tab to perform various operations.
 - On the **Instance Details** tab, you can view the Instance Access URL, Basic Information, and Tables sections.
 - On the **Network Management** tab, you can bind or unbind VPCs and view the list of VPCs.
- Click **Release** in the **Actions** column to release an instance.

 **Notice** To create an instance when you release an existing instance, ensure that the name of the instance to create is different from that of the existing instance to avoid conflicts.

8.3.3. Create tables

This topic describes how to create a table in the Tablestore console.

Procedure

1. **Log on to the Tablestore console.**
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the **Actions** column.
3. On the **Instance Details** tab, click **Create Table**.

 **Note** You can create a maximum of 1,024 tables in each instance.

4. In the **Create Table** dialog box, set **Table Name** and **Primary Key**. The following table describes the parameters you can configure.

Parameter	Description
Table Name	<p>The name of the table. This name is used to uniquely identify a table in an instance.</p> <p>The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).</p>
Primary Key	<p>One or more primary key columns in the table that uniquely identify each record in the table.</p> <p>Enter a primary key name and select a data type. Click Add a Primary Key to add a primary key column.</p> <p>You can add one to four primary key columns. By default, the first primary key column is the partition key. The configurations and order of primary key columns cannot be modified after the table is created.</p> <div style="background-color: #e1f5fe; padding: 5px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ◦ In Tablestore, only a primary key column can be used as an auto-increment primary key column. Partition keys cannot be used as auto-increment primary key columns. ◦ After a primary key column is set to an auto-increment primary key column, Tablestore automatically generates a value for the auto-increment primary key column when you write a row of data. You do not need to specify a value for the auto-increment primary key column. The values of auto-increment primary key columns are incremental and unique within the rows that share the same partition key. </div> <ul style="list-style-type: none"> ◦ Naming conventions of primary key columns: The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_) ◦ Data types supported by primary key columns are String, Integer, and Binary.

5. **Optional. Configure advanced parameters.** If you need to configure parameters such as **Time to Live** and **Max Versions**, perform this operation.
 - i. **Turn on Advanced Settings.**

- ii. Configure advanced parameters. The following table describes the advanced parameters you can configure.

Parameter	Description
Time to Live	<p>The period for which data in the table can be retained. When the retention period exceeds the Time to Live (TTL) value, the system deletes the expired data.</p> <p>The minimum TTL value is 86,400 seconds (one day). A value of -1 indicates that data never expires.</p>
Max Versions	<p>The maximum number of versions of data that can be retained for an attribute column. When the versions of data in an attribute column exceed the Max Versions value, the system deletes the earliest versions of data to keep the maximum number of versions equal to the Max Versions value.</p> <p>Valid values: 1 to 10.</p>
Max Version Offset	<p>The difference between the version number and the data written time must be within the value of Max Version Offset. Otherwise, an error occurs when the data is written. Unit: seconds.</p> <p>The valid version range for attribute columns is calculated based on the following formula: Valid version range = [Data written time - Max version offset value, Data written time + Max version offset value).</p>
Reserved Read Throughput	<p>You can set this parameter only for high-performance instances.</p> <p>The read and write throughput that is allocated and reserved for the table.</p>
Reserved Write Throughput	<p>Valid values: integers from 0 to 5000.</p> <p>When the specified reserved read and write throughput is 0, Tablestore does not reserve related resources for the table.</p>

6. Optional. Create secondary indexes. If you need to create secondary indexes, perform this operation.

- i. Turn on **Create Secondary Index**.
- ii. Click the **+ Add** button in the **Pre-defined Column** section. Enter the name of the pre-defined column and select a data type from the drop-down list.
 - This operation is performed to create a predefined column for the base table. Tablestore uses a schema-free model. You can write an unlimited number of columns to a row and do not need to specify a fixed number of predefined columns in a schema. When you create a table, you can also predefine columns and specify their data types.
 - You can add up to 14 predefined columns. To delete the predefined column you add, click the  icon on the left of the corresponding predefined column.
 - The name of a predefined column must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_).
 - The data types of predefined columns include **STRING**, **INTEGER**, **BINARY**, **FLOAT**, and **BOOLEAN**.
- iii. Click **Add Secondary Index**. Enter **Index Name** and set **Primary Key** and **Pre-defined Column** for the index table.
 - The name of an index table must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_).
 - You can set the primary key of the index table to the primary key or predefined columns of the base table.
 - **Pre-defined Column** is optional. You can set the predefined columns of the index table to only the predefined columns of the base table.

7. Click **OK**. After a table is created, you can view the table in the **Table List** section. If the created table is not displayed in the list of tables, click the  icon to refresh the list of tables.

After a table is created, you can perform the following operations on the table:

- Click the name of the table or click **Details** in the **Actions** column. On the **Manage Table** page, you can perform the following operations:
 - On the **Details** tab, you can view the description of the table and the primary key columns list, and modify the attributes of the table.
 - On the **Data Editor** tab, you can insert or update data, query data, view data details, and delete multiple data at a time.
- Click the  icon in the **Actions** column corresponding to a table and choose **Delete** from the shortcut menu. Click **OK** in the **Delete Table** dialog box. The table is deleted.



Notice If you delete a table, the table and the data in the table are permanently deleted from Tablestore and cannot be recovered. Exercise caution when you perform this operation.

8.3.4. Read and write data in the console

After a table is created, you can read data from and write data to the table in the console.

Add data

1. [Log on to the Tablestore console.](#)
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the **Actions** column.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the **Actions** column.
4. On the **Data Editor** tab, click **Insert**.
5. In the **Insert** dialog box that appears, set **Primary Key Value**. Click **Add Column**. Set **Name**, **Type**, **Value**, and **Version**. By default, **System Time** is selected, indicating that the current system time is used as the version number of the data. You can also clear **System Time** and enter the version number of the data.
6. Click **OK**. Rows that contain the written data are displayed on the **Data Editor** tab.

Delete data

You can delete data you no longer need.

1. [Log on to the Tablestore console.](#)
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the **Actions** column.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the **Actions** column.
4. On the **Data Editor** tab, select the row of data you want to delete. Click **Delete**.
5. In the **Delete** message that appears, click **OK**.

Update data

You can update data in the attribute columns of a row.

1. [Log on to the Tablestore console.](#)
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the **Actions** column.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data**

Editor tab. You can also click **Data Editor** in the Actions column.

4. On the **Data Editor** tab, select the row of data to update. Click **Update**.
5. In the **Update** dialog box that appears, modify the type and value for the primary key, add or remove attribute columns, and update or delete data in attribute columns.
 - You can click **+Add Column** to add an attribute column. You can also click the  icon to delete an attribute column.
 - If you select **Update**, you can modify data in attribute columns. If you select **Delete**, select the number of version to delete. If you select **Delete All**, all versions of the data are deleted.
6. Click **OK**.

Query data

In the Tablestore console, you can query data in a single row (**GetRow**) or query data within a specified range (**RangeQuery**).

To query data in a single row, perform the following operations:

1. [Log on to the Tablestore console](#).
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the Actions column.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, click **Search**.
5. Set filter conditions.
 - i. In the **Search** dialog box, Set **Modes** to **GetRow**.
 - ii. By default, the system returns all columns. To return specified attribute columns, turn off **All Columns**. Enter the names of the attribute columns to return. Separate the names of the attribute columns with commas (,).
 - iii. Set **Primary Key Value**.The integrity and accuracy of the primary key value affect the query results.
 - iv. Set **Count of Versions** to specify the maximum number of versions to return.
6. Click **OK**.Data that meets the filter conditions is displayed on the **Data Editor** tab.

To perform range query, perform the following steps:

1. [Log on to the Tablestore console](#).
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the Actions column.
3. In the **Table List** section of the **Instance Details** tab, click the name of the target table and click the **Data Editor** tab. You can also click **Data Editor** in the Actions column.
4. On the **Data Editor** tab, click **Search**.
5. Set filter conditions.
 - i. In the **Search** dialog box, Set **Modes** to **Range Search**.
 - ii. By default, the system returns all columns. To return specified attribute columns, turn off **All Columns**. Enter the names of the attribute columns to return. Separate the names of the attribute columns with commas (,).

- iii. Set **Start Primary Key Column** and **End Primary Key Column**. You can set **Start Primary Key Column** to **Min Value** or **Custom** and **End Primary Key Column** to **Max Value** or **Custom**. If you select **Custom**, enter a custom value.

 **Note**

- The value in the first primary key column takes priority when the range query mode is used. When the minimum and maximum values for the first primary key column are the same, the system uses the value in the second primary key column to perform the query. The query rules for the subsequent primary keys are the same as those for the first two primary keys.
- The Custom range is a left-open and right-closed interval.

- iv. Set **Count of Versions** to specify the maximum number of versions to return.
 - v. Set **Sequence** to **Forward Search** or **Backward Search**.
6. Click **OK**. Data that meets the filter conditions is displayed based on the specified order on the **Data Editor** tab.

8.3.5. Bind a VPC to a Tablestore instance

After you bind a VPC to a Tablestore instance, you can access the Tablestore instance from the ECS instances in the VPC in the same region.

Prerequisites

- A VPC that is within the same region as the Tablestore instance is created.
- After the VPC is created, create an ECS instance in the VPC.

Procedure

1. [Log on to the Table Store console](#).
2. On the **Overview** page, click the name of the target instance or click **Manage Instance** in the **Actions** column.
3. Click the **Network Management** tab.
4. On the **Network Management** tab, click **Bind VPC**.
5. In the **Bind VPC** dialog box, select a VPC and switch, enter **Instance VPC Name**. The name of a VPC can contain only letters and digits and must start with a letter. The name must be 3 to 16 bytes in length.
6. Click **OK**.

After the VPC is bound to the instance, you can view the information of the VPC in the **VPC List** on the **Network Management** tab. You can use the VPC address to access the Tablestore instance from the ECS instances in the VPC.

After you bind a VPC, you can perform the following operations:

- Click **VPC Instance List** in the **Actions** column to view the VPC instances list, which contains the instance name, instance VPC name, and VPC domain name.
- Click **Unbind** in the **Actions** column to unbind the VPC from the instance. After the VPC is unbound, the ECS instance in the VPC can no longer access the Tablestore instance by using the VPC address. To access the Tablestore instance from the ECS instance, you must bind the VPC to the Tablestore instance again.

9. ApsaraDB RDS for MySQL

9.1. What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage of Alibaba Cloud, ApsaraDB for RDS allows you to easily perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

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 for RDS handles a small number of write requests but a large number of read requests, you can create read-only RDS instances that run MySQL 5.6 to scale up the reading capability and increase the application throughput.
- **Read/write splitting:** The read/write splitting feature provides an extra 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 automatically distributed to the primary instance while read requests are distributed to read-only instances based on their weights. To scale up the reading capacity of the system, you can add more read-only instances.

9.2. Log on to the ApsaraDB for RDS console

This topic describes how to log on to the ApsaraDB for RDS console.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Database Services > ApsaraDB for RDS**.

9.3. Quick start

9.3.1. Limits

To ensure instance stability and security, ApsaraDB RDS for MySQL has some service limits, as listed in the following table.

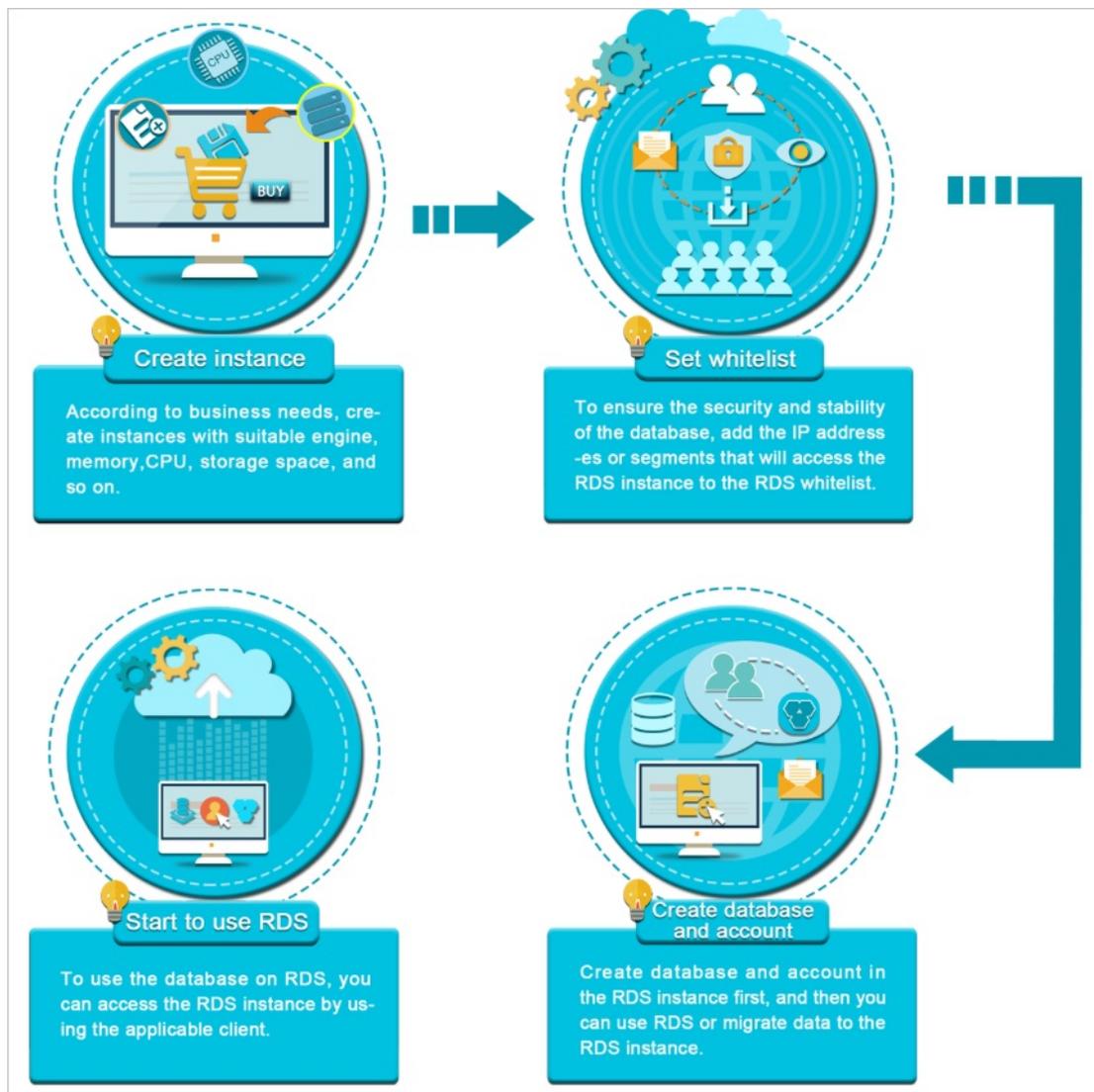
Operation	Description
Instance parameters	Instance parameters can be modified through the RDS console or API operations. Due to security and stability considerations, only specific parameters can be modified.
Root permissions of databases	The root or system administrator permissions are not provided.
Database backup	<ul style="list-style-type: none"> Logical backup can be performed through the command line interface (CLI) or graphical user interface (GUI). Physical backup can only be performed through the RDS console or API operations.
Database restoration	<ul style="list-style-type: none"> Logical restoration can be performed through the CLI or GUI. Physical restoration can only be performed through the RDS console or API operations.
ApsaraDB RDS for MySQL storage engine	<p>Only InnoDB is supported.</p> <ul style="list-style-type: none"> For safety performance and security considerations, we recommend that you use the InnoDB storage engine. The TokuDB engine is not supported. Percona no longer provides support for TokuDB, leading to bugs that cannot be fixed and can cause business losses in extreme cases. The MyISAM engine is not supported. Due to the inherent shortcomings of the MyISAM engine, some data may be lost. Only some existing instances use the MyISAM engine. MyISAM engine tables in newly created instances will be automatically converted to InnoDB engine tables. The Memory engine is not supported. Newly created Memory tables will be automatically converted into InnoDB tables.
Database replication	ApsaraDB RDS for MySQL provides dual-node clusters based on a primary/secondary replication architecture. The secondary instances in this replication architecture are hidden and cannot be accessed directly.
RDS instance restart	Instances must be restarted through the RDS console or API operations.
Account and database management	ApsaraDB RDS for MySQL uses the RDS console to manage accounts and databases. ApsaraDB RDS for MySQL also allows you to create a privileged account to manage users, passwords, and databases.
Standard account	<ul style="list-style-type: none"> Authorization is not allowed. The RDS console allows you to manage accounts and databases. Instances that support standard accounts also support privileged accounts.
Privileged account	<ul style="list-style-type: none"> Authorization is allowed to standard accounts. The RDS console does not provide interfaces to manage accounts or databases. These operations can only be performed through code or DMS. The privileged account cannot be reverted back to a standard account.

9.3.2. Procedure

ApsaraDB for RDS quick start covers the following topics: creating an RDS instance, configuring a whitelist, creating a database, creating an account, and connecting to the instance. This topic uses ApsaraDB RDS for MySQL as an example to describe how to use RDS. It provides all the necessary information to create an RDS instance.

Typically, you must complete several operations after instance creation to make it ready for use, as shown in [Quick start flowchart](#).

Quick start flowchart



- **Create an instance**
An instance is a virtualized database server on which you can create and manage multiple databases.
- **Configure a whitelist**
After creating an RDS instance, you must configure its whitelist to allow access from external devices. The whitelist improves the access security of your RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the RDS instance.
- **Create a database and Create an account**
Before using a database, you must first create the database and an account in the RDS instance.
- **Connect to an ApsaraDB RDS for MySQL instance**
After creating an RDS instance, configuring a whitelist, and creating a database and an account, you can connect to the instance from a database client.

9.3.3. Create an instance

This topic describes how to create an instance in the ApsaraDB for RDS console.

Prerequisites

An Apsara Stack tenant account is obtained.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides. Services in different regions cannot communicate over the internal network. After a region is selected, it cannot be changed.
	Zone	The zone where the instance resides.
Specifications	Instance Name	The name of the instance. <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain digits and special characters. Special characters include _ - : ◦ The name cannot start with http:// or https://.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page.
	Engine Version	The version of the database engine. Set the value to 5.6 or 5.7.
	Edition	The edition of ApsaraDB for RDS. The actual values are displayed in the console.
	Storage Type	The storage type of the instance. The actual values are displayed in the console.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. Use the following path to navigate the guide: Product Introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files. The actual values are displayed in the console.

Section	Parameter	Description
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. <p> Note After you select VPC as the network type, you must also select the corresponding VPC and VSwitch.</p>
	IP Whitelist	The IP addresses that are allowed to connect to the ApsaraDB for RDS instance.
Access Mode	Access Mode	<p>The access mode of the instance, which is automatically set to Standard.</p> <p> Note Standard: ApsaraDB for RDS uses SLB to eliminate the impact of instance high-availability switchover on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.</p>

4. After you configure the preceding parameters, click **Submit**.

9.3.4. Initialization settings

9.3.4.1. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB for RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB for RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB for RDS instance.

To configure a whitelist, perform the following operations:

- **Configure a whitelist**: Add IP addresses to allow them to connect to the RDS instance.
- **Configure an ECS security group**: Add an ECS security group for the RDS instance to allow ECS instances in the group to connect to the RDS instance.

Precautions

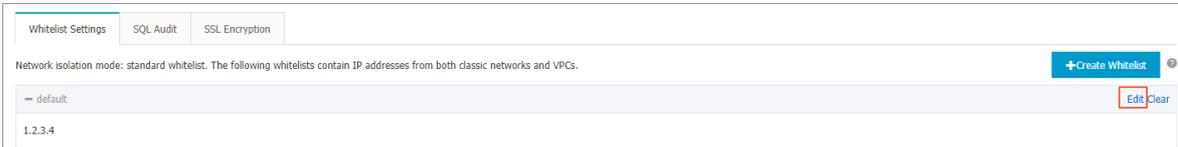
- The default whitelist can be modified or cleared, but cannot be deleted.
- You can add up to 1,000 IP addresses or CIDR blocks to a whitelist. If you want to add a large number of IP addresses, we recommend that you merge them into CIDR blocks, such as 192.168.1.0/24.

Configure a standard IP address whitelist

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

- In the left-side navigation pane, click **Data Security**.
- On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

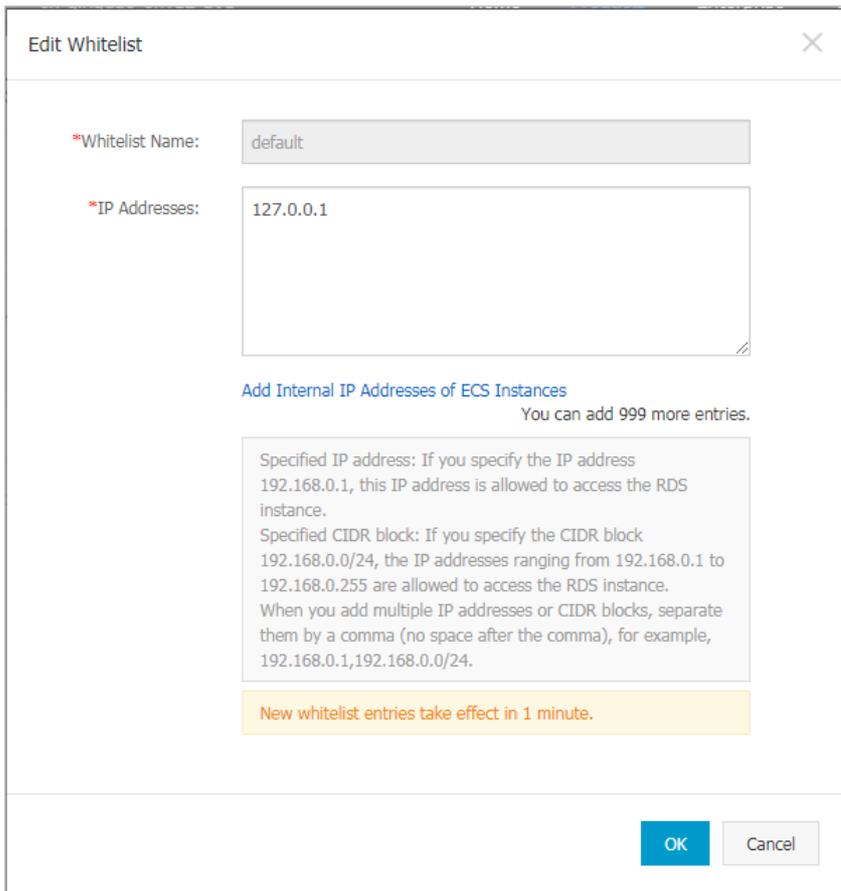


Note

- If you want to connect an ECS instance to an ApsaraDB for RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
- You can click **Create Whitelist** to create a new whitelist.

- In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**.
 - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - After you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances under your Apsara Stack account are displayed. You can select the required IP addresses and add them to the whitelist.

Note If you add a new IP address or CIDR block to the default whitelist, the default address 127.0.0.1 is automatically deleted.



9.3.4.2. Create an account

After you create an ApsaraDB for RDS instance and configure its whitelist, you must create a database and an account in the instance. This topic describes how to create privileged and standard accounts.

Context

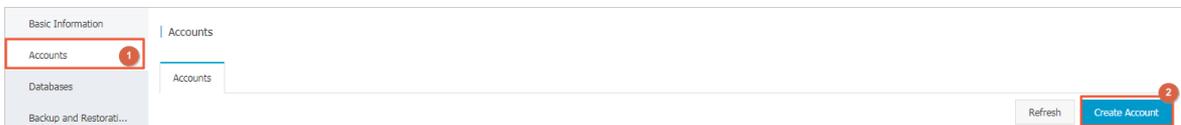
ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB for RDS console. For more information about permissions that can be granted to each type of account, see [Account permissions](#).

Account type	Description
Privileged account	<ul style="list-style-type: none"> You can create and manage privileged accounts by using the ApsaraDB for RDS console or API operations. You can create only one privileged account on each RDS instance. The privileged account can be used to manage all standard accounts and databases on the instance. A privileged account allows you to manage permissions to a fine level. For example, you can grant each standard account the permissions to query specific tables. A privileged account has permissions to disconnect all standard accounts on the instance.
Standard account	<ul style="list-style-type: none"> You can create and manage standard accounts by using the ApsaraDB for RDS console, API operations, or SQL statements. You can create up to 500 standard accounts for an instance. You must manually grant standard accounts the specific database permissions. You cannot use a standard account to create, manage, or disconnect other accounts from databases.

Account type	Number of databases	Number of tables	Number of accounts
Privileged account	Unlimited	< 200,000	Varies based on the kernel parameter settings of the instance.
Standard account	500	< 200,000	Varies based on the kernel parameter settings of the instance.

Create a privileged account

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: <ul style="list-style-type: none"> ◦ The name must be 1 to 16 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Privileged Account.
Password	Enter the password of the standard account. The password must meet the following requirements: <ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the privileged account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Reset the permissions of a privileged account

If an issue occurs on the privileged account, for example, permissions are unexpectedly revoked, you can enter the password of the privileged account to reset permissions.

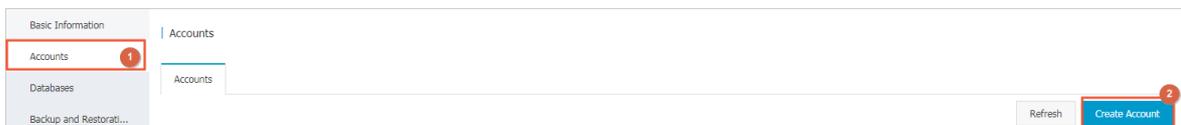
1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

4. In the left-side navigation pane, click **Accounts**.
5. Find the privileged account, and click **Reset Permissions** in the **Actions** column.
6. Enter the password of the privileged account and click **OK**.

Create a standard account

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	<p>Enter the account name. The account name must meet the following requirements:</p> <ul style="list-style-type: none"> ◦ The name must be 1 to 16 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Standard Account.
Authorized Databases	<p>Select one or more databases on which you want to grant permissions to the account. You do not have to configure this parameter at this time. You can authorize databases after the account is created.</p> <ol style="list-style-type: none"> i. Select one or more databases from the Unauthorized Databases section and click Add to add them to the Authorized Databases box. ii. In the Authorized Databases section, select the Read/Write, Read-only, DDL Only, or DML Only permissions on each authorized database. <p>If you want to grant the same permissions on multiple databases to the account, click the button in the upper-right corner of the section. The button may appear as Set All to Read/Write.</p>
Password	<p>Enter the password of the standard account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the standard account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Account permissions

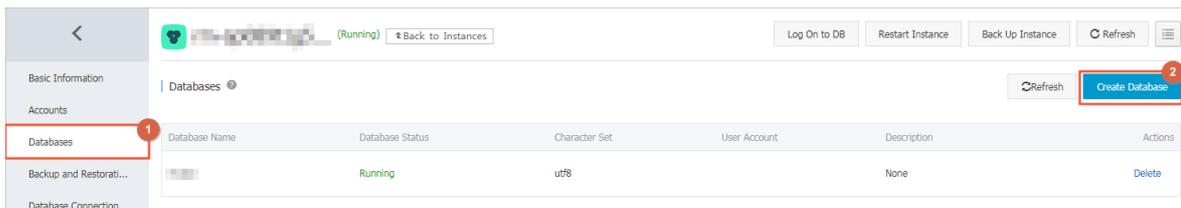
Account type	Authorization type	Permission				
Privileged account	-	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	RELOAD	PROCESS	REFERENCES	INDEX
		ALTER	CREATE TEMPORARY TABLES	LOCK TABLES	EXECUTE	REPLICATION SLAVE
		REPLICATION CLIENT	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		CREATE USER	EVENT	TRIGGER	-	-
Standard account	Read-only	SELECT	LOCK TABLES	SHOW VIEW	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	Read/write	SELECT	INSERT	UPDATE	DELETE	CREATE
		DROP	REFERENCES	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	CREATE VIEW	SHOW VIEW	CREATE ROUTINE
		ALTER ROUTINE	EVENT	TRIGGER	PROCESS	REPLICATION SLAVE
		REPLICATION CLIENT	-	-	-	-
	DDL-only	CREATE	DROP	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-
	DML-only	SELECT	INSERT	UPDATE	DELETE	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	SHOW VIEW	EVENT	TRIGGER
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-

9.3.4.3. Create a database

After you create an ApsaraDB for RDS instance and configure its whitelist, you must create a database and an account in the instance.

Procedure

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Click **Create Database**.



6. Configure the following parameters.

Parameter	Description
Database Name	<ul style="list-style-type: none"> ◦ The name must be 1 to 64 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain lowercase letters, digits, underscores (_), and hyphens (-). ◦ Each database name must be unique in an instance.
Supported Character Set	Select utf8, gbk, latin1, utf8mb4, or all. If you want to use other character sets, select all, and then select the required character set from the list.
Description	Optional. Enter information about the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

9.3.5. Connect to an ApsaraDB RDS for MySQL instance

After you complete the initial configuration of your ApsaraDB RDS for MySQL instance, you can connect to it from an Elastic Compute Service (ECS) instance or an on-premises client.

Context

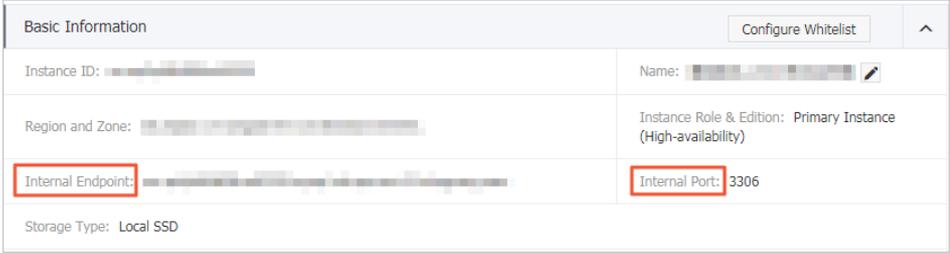
After you perform operations such as [Create an instance](#), [Configure a whitelist](#), and [Create an account](#), you can use a general database client or configure the endpoint, port number, and account information in an application to connect to the MySQL instance.

If you need to connect an ECS instance to an ApsaraDB for RDS instance, you must make sure that both instances are in classic networks or in the same VPC, and the IP address of the ECS instance is correctly configured in the RDS whitelist.

Connect to an instance from a client

ApsaraDB RDS for MySQL is fully compatible with open source MySQL. You can connect to an ApsaraDB RDS for MySQL instance from a database client by using a similar method that you use to connect to an open source MySQL database. In the following example, the [HeidiSQL](#) client is used.

1. Start the HeidiSQL client.
2. In the lower-left corner of the Session manager dialog box, click **New**.
3. Enter information about the RDS instance that you want to connect. The following table describes the required parameters.

Parameter	Description
Network type	Select the network type of the RDS instance that you want to connect. For this example, select MariaDB or MySQL (TCP/IP).
Hostname / IP	<p>Enter the internal or public endpoint of the RDS instance.</p> <ul style="list-style-type: none"> ◦ If your client is deployed on an ECS instance that is in the same region and has the same network type as the RDS instance, use the internal endpoint. For example, if your ECS and RDS instances are both in a VPC located in the China (Hangzhou) region, you can use the internal endpoint of the RDS instance to create a secure connection. ◦ In other scenarios, use the public endpoint. <p>To view the internal and public endpoints and port numbers of the RDS instance, follow these steps:</p> <ol style="list-style-type: none"> Log on to the ApsaraDB for RDS console. Find the target RDS instance and click its ID. In the Basic Information section, view the internal endpoint and internal port number of the instance. 
User	Enter the username of the account that you use to connect to the RDS instance.
Password	Enter the password of the account.
Port	If you connect to the instance over an internal network, enter the internal port number of the instance. If you connect to the instance over the Internet, enter the public port number of the instance.

4. Click **Open**. If the connection information is correct, you can connect to the instance.

9.4. Instances

9.4.1. Create an instance

This topic describes how to create an instance in the ApsaraDB for RDS console.

Prerequisites

An Apsara Stack tenant account is obtained.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, click **Create Instance** in the upper-right corner.

3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides. Services in different regions cannot communicate over the internal network. After a region is selected, it cannot be changed.
	Zone	The zone where the instance resides.
Specifications	Instance Name	The name of the instance. <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain digits and special characters. Special characters include _ - : ◦ The name cannot start with http:// or https://.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page.
	Engine Version	The version of the database engine. Set the value to 5.6 or 5.7.
	Edition	The edition of ApsaraDB for RDS. The actual values are displayed in the console.
	Storage Type	The storage type of the instance. The actual values are displayed in the console.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. Use the following path to navigate the guide: Product Introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files. The actual values are displayed in the console.
	Network Type	The network type of the instance. RDS instances support the following network types: <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note After you select VPC as the network type, you must also select the corresponding VPC and VSwitch. </div>
IP Whitelist	The IP addresses that are allowed to connect to the ApsaraDB for RDS instance.	

Section	Parameter	Description
Access Mode	Access Mode	<p>The access mode of the instance, which is automatically set to Standard.</p> <p> Note Standard: ApsaraDB for RDS uses SLB to eliminate the impact of instance high-availability switchover on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.</p>

4. After you configure the preceding parameters, click **Submit**.

9.4.2. Create an ApsaraDB RDS for MySQL instance with standard SSDs or enhanced SSDs

Disks are block-level data storage products provided by Alibaba Cloud for ECS. They feature low latency and high performance, durability, and reliability. This topic describes how to create an instance with standard or enhanced SSDs in the ApsaraDB for RDS console.

Prerequisites

The instance runs MySQL 5.7 on RDS High-availability Edition.

Context

An RDS instance with standard or enhanced SSDs uses a distributed triplicate mechanism to ensure 99.9999999% data reliability. If service disruptions occur (for example, due to hardware failure) within a zone, data in that zone is copied to an unaffected disk in another zone to ensure data availability.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides. Services in different regions cannot communicate over the internal network. After a region is selected, it cannot be changed.
	Zone	The zone where the instance resides.
	Instance Name	<p>The name of the instance.</p> <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain digits and special characters. Special characters include <code>_</code> and <code>:</code>. ◦ The name cannot start with <code>http://</code> or <code>https://</code>.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page.

Section Specification	Parameter	Description
5	Engine Version	The version of the database engine. Select 5.7.
	Edition	The edition of the database. Select High-availability Edition.
	Storage Type	The storage type of the database. Select cloud ssd.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed on the console. For more information, see Product Introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage space of the instance, including the space for data, system files, binary log files, and transaction files.
Network Type	Network Type	<p>The network type of the instance, which is automatically set to VPC.</p> <p>Note VPC: A Virtual Private Cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways within a VPC. We recommend that you select VPC for improved security.</p> <p>After you select VPC as the network type, you must also select the corresponding VPC and VSwitch.</p>
	IP Whitelist	The IP addresses that are allowed to connect to the ApsaraDB for RDS instance.
Access Mode	Access Mode	<p>The access mode of the instance, which is automatically set to Standard.</p> <p>Note Standard: RDS uses Server Load Balancer (SLB) to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.</p>

4. Click Submit.

9.4.3. View basic information of an instance

This topic describes how to view the details of an ApsaraDB for RDS instance, such as basic information, internal network connection information, status, and configurations.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. Use one of the following methods to go to the **Basic Information** page of an instance:
 - On the **Instances** page, find the target instance and click the instance ID. The **Basic Information** page appears.
 - On the **Instances** page, find the target instance and click **Manage** in the corresponding **Actions** column. The **Basic Information** page appears.

9.4.4. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS for MySQL instance. This applies if the number of connections exceeds the specified threshold or if an instance has any performance issues.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Restart Instance** in the upper-right corner.

 **Note** A restart will disconnect the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

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

9.4.5. Change the specifications of an instance

This topic describes how to change specifications of your instance, such as the instance type and storage space, if the specifications do not meet the requirements of your application.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. Find your RDS instance and click its ID.
3. In the **Configure Information** section of the **Basic Information** page, click **Change Specifications**.

Configuration Information			Change Specifications	^
Instance Family: Dedicated Instance	Database Engine: MySQL 5.6	CPU: 2 Cores		
Memory: 16384MB	Maximum IOPS: 4500	Maximum Connections: 2500		
Maintenance Window: 02:00-06:00 Configure	Instance Type: mysql.x8.medium.2			

4. In the **Upgrade** dialog box that appears, click **Next**.
5. On the **Change Specifications** page, set **Edition**, **Instance Type**, and **Storage**.
6. After you configure the preceding parameters, click **Submit**.

9.4.6. Set a maintenance window

You can set a maintenance window for an ApsaraDB for RDS instance as needed.

Context

To ensure the stability of ApsaraDB for RDS instances, the backend system performs maintenance of the instances at irregular intervals. The default maintenance window is from 02:00 to 06:00. You can set the maintenance window to the off-peak period of your business to avoid impact on business.

Precautions

- To ensure the stability of the maintenance process, the instance will enter the **Maintaining Instance** state before the maintenance time. When the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, apart from account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart are temporarily unavailable.
- During the maintenance window, the instance is disconnected once or twice. Make sure that you configure automatic reconnection policies for your applications to avoid service disruptions.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. Click the ID of an instance or click **Manage** in the Actions column corresponding to the instance.
3. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.

The screenshot displays the configuration page for an ApsaraDB for RDS instance. It is divided into several sections:

- Basic Information:** Shows Instance ID, Name, Region and Zone, Internal Endpoint, Internal Port (3306), Storage Type (Local SSD), and Read/Write Splitting Endpoint. A note indicates that the preceding endpoint should be used for connection, with a requirement to change the VIP in the endpoint to the one used in the environment.
- Status:** Shows the instance is **Running** and was created on Dec 31, 2019, at 10:22:37.
- Configuration Information:** Shows Instance Family (Dedicated Instance), Database Engine (MySQL 5.6), CPU (2 Cores), Memory (16384MB), Maximum IOPS (4500), Maximum Connections (2500), and Instance Type (mysql.x8.medium.2). The **Maintenance Window** is set to 02:00-06:00, and a **Configure** link is visible next to it.
- Usage Statistics:** Shows Storage Capacity (Used 3.16G, Capacity 50.00G) and Space Used for Backup (Data Size: 15.72M, Log Size: 43.42M).

4. Select a maintenance window and click **Save**.

Note The maintenance window is in UTC+8.

9.4.7. Change the data replication mode

You can set the data replication mode between primary and secondary ApsaraDB for RDS instances to improve database availability.

Context

- **Semi-synchronous**

After an application-initiated update is completed on the primary instance, logs are synchronized to all secondary instances. This transaction is considered committed after at least one secondary instance has received the logs, no matter whether the secondary instance finishes executing the updates specified in the logs.

If the secondary instances are unavailable or a network exception occurs between the primary and secondary instances, semi-synchronous replication will degrade to the Asynchronous mode.

- **Asynchronous**

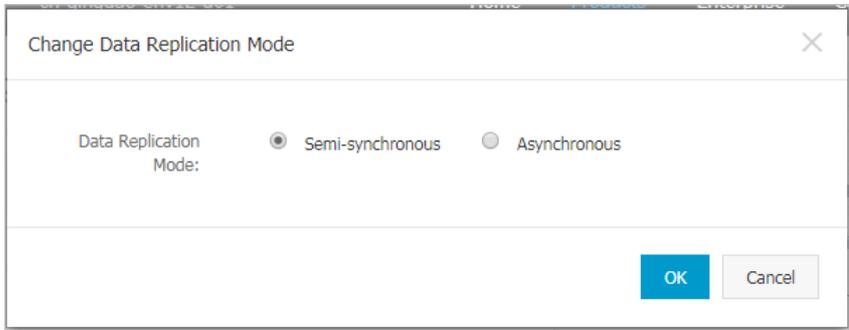
When your application initiates a request to add, delete, or modify data, the primary instance responds to your application immediately after it completes the operation. At the same time, the primary instance starts to asynchronously replicate data to its secondary instances. During asynchronous data replication, the unavailability of secondary instances does not affect the operations on the primary instance. Data remains consistent even if the primary instance is unavailable.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. Find your RDS instance and click its ID.
3. In the left-side navigation pane, click **Service Availability**.
4. Click **Change Data Replication Mode**.



5. In the dialog box that appears, select a data replication mode and click **OK**.



9.4.8. Release an instance

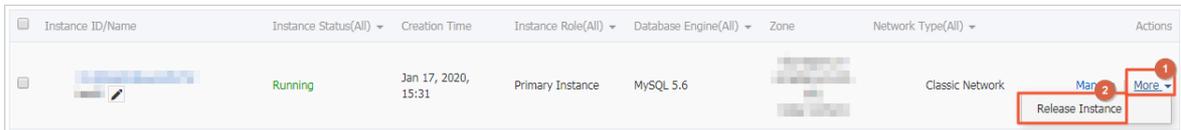
You can manually release instances as needed.

Precautions

- You can manually release only instances that are in the running state.
- After an instance is released, the instance data is immediately cleared. We recommend that you back up your data before you release an instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. In the Actions column corresponding to the instance you want to release, choose **More > Release Instance**.



3. In the Release Instance message that appears, click **Confirm**.

9.4.9. Upgrade the kernel version of an instance

ApsaraDB RDS for MySQL supports automatic and manual updates of the kernel version. These updates increase performance, unveil new features, and fix known issues.

Introduction

ApsaraDB RDS for MySQL automatically upgrades the kernel version by default. You can log on to the ApsaraDB for RDS console, navigate to the **Basic Information** page of your RDS instance, and then view the current **Minor Version Upgrade Mode** in the Configuration Information section.

- **Auto:** When a new kernel version is released, the system automatically upgrades the kernel version of your RDS instance during the specified maintenance window. For more information, see [Set a maintenance window](#).
- **Manual:** You can manually upgrade the kernel version on the **Basic Information** page. For more information, see [Manually upgrade the kernel version](#).

Precautions

- While you upgrade the kernel version of your RDS instance, a 30-second network interruption may occur. We

recommend that you upgrade the kernel version during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

- The kernel version of your RDS instance cannot be downgraded after it is upgraded.
- After you upgrade the specifications of your RDS instance, the kernel version of your RDS instance is upgraded accordingly.

Configure the kernel version upgrade mode

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the **Basic Information** page, click **Configure** to the right of **Minor Version Upgrade Mode**.
5. Select **Auto** or **Manual**, and click **OK**.

Manually upgrade the kernel version

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the page, click **Upgrade Minor Version**.

 **Note** The **Upgrade Minor Version** button is displayed only when a new kernel version is available.

5. In the dialog box that appears, specify the upgrade time and click **OK**.

FAQ

- **Q:** After I upgraded the kernel version of my RDS instance, why does the `SELECT @@version` statement still return the source kernel version that I used before the upgrade?
A: The kernel version that you upgraded is the kernel version of Alibaba Cloud instead of the kernel version of MySQL. You need to execute the `show variables like '%rds_release_date%'` statement to view the kernel version of your RDS instance.
- **Q:** Is each upgrade targeted only for the next kernel version?
A: No, each upgrade is targeted for the latest kernel version.

9.4.10. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query parameter modification records in the console.

Precautions

- To ensure instance stability, you can select specific parameters to modify in the ApsaraDB for RDS console.
- When you modify parameters on the **Editable Parameters** tab, refer to the **Value Range** column corresponding to each parameter.
- After some parameters are modified, you must restart your RDS instance for the changes to take effect. For more information, see the **Force Restart** column on the **Editable Parameters** tab. We recommend that you modify the parameters of an instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.

Modify parameters

1. [Log on to the ApsaraDB for RDS console.](#)

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. You can perform the following operations: Export the parameter settings of the RDS instance to your computer.

On the **Editable Parameters** tab, click **Export Parameters**. The parameter settings of the RDS instance are exported as a TXT file to your computer.

Modify and import the parameter settings.

- i. After you have modified parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.
- ii. Click **OK**.
- iii. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
- Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

Modify a single parameter.

- i. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the  icon in the **Actual Value** column.
- ii. Enter a new value based on the prompted value range.
- iii. Click **Confirm**.
- iv. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
- Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the page that appears, click the **Edit History** tab.
6. Select a time range and then click **Search**.

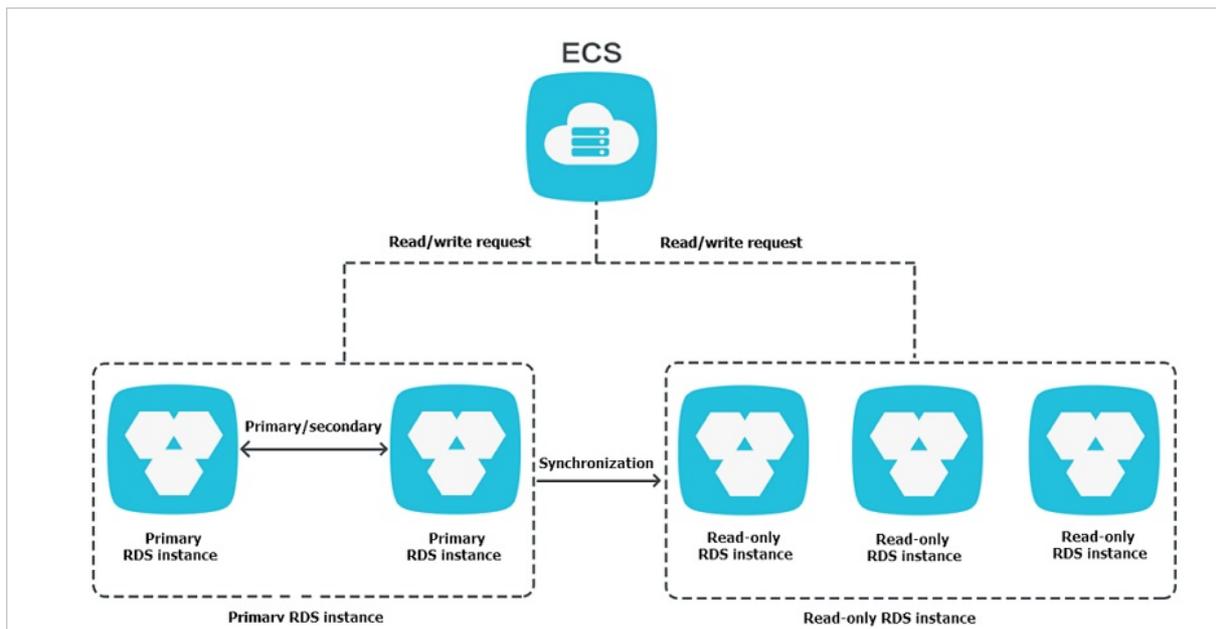
9.4.11. Read-only instances

9.4.11.1. Overview of read-only instances

ApsaraDB RDS for MySQL allows you to create read-only instances. In scenarios where RDS has a small number of write requests but a large number of read requests, you can create read-only instances to distribute database access loads away from the primary instance. This topic describes the features and limits of read-only instances.

To scale the reading capability and distribute database access loads, you can create one or more read-only instances in a region. Read-only instances allow RDS to increase the application throughput when a large amount of data is being read.

A read-only instance with a single physical node and no backup node uses the native replication capability of MySQL to synchronize changes from the primary instance to all its read-only instances. Read-only instances must be in the same region as the primary instance but do not have to be in the same zone as the primary instance. The following figure shows the topology of read-only instances.



Read-only instances have the following features:

- Specifications of a read-only instance can be different from those of the primary instance and can be changed at any time, which facilitates elastic scaling.
- Read-only instances do not require account or database maintenance. Account and database information is synchronized from the primary instance.
- The whitelists of read-only instances can be configured independently.
- System performance monitoring is provided.

RDS provides up to 20 system performance monitoring views, including those for disk capacity, IOPS, connections, CPU utilization, and network traffic. You can view the load of instances.

- RDS provides a variety of optimization recommendations, such as storage engine check, primary key check, large table check, and check for excessive indexes and missing indexes. You can optimize your databases based on the optimization recommendations and specific applications.

9.4.11.2. Create a read-only instance

You can create read-only instances of different specifications based on your business requirements.

Prerequisites

The RDS instance runs High-availability Edition (with local SSDs) or Enterprise Edition.

Precautions

- A maximum of five read-only instances can be created for a primary instance.

- Backup settings and temporary backup are not supported.
- Instance restoration is not supported.
- Data migration to read-only instances is not supported.
- Database creation and deletion are not supported.
- Account creation, deletion, authorization, and password changes are not supported.
- After a read-only instance is created, you cannot restore data by directly overwriting the primary instance with a backup set.

Procedure

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, click **Create Read-only Instance** on the right side of the page.
5. On the **Create Read-only RDS Instance** page, configure the read-only instance parameters.

Section	Parameter	Description
Region	Region	The region where the ApsaraDB for RDS instance resides.
Specifications	Database Engine	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	Engine Version	The engine version of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	Edition	Set the value to Read-only Instance .
	Instance Type	The instance type of the read-only instance. The type of the read-only instance can be different from that of the primary instance, and can be modified at any time to facilitate flexible upgrade and downgrade.
	Storage	The storage space of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type and storage space as the primary instance for the read-only instance.
Network Type	Network Type	The network type of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	VPC	Select a VPC if the network type is set to VPC.
	VSwitch	Select a VSwitch if the network type is set to VPC.

6. After you configure the preceding parameters, click **Submit**.

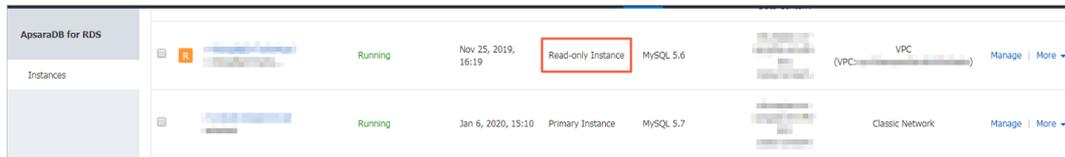
9.4.11.3. View details of read-only instances

This topic describes how to view details of read-only instances. You can go to the **Basic Information** page of a read-only instance from the **Instances** page or the read-only instance list of the primary instance. Read-only instances are managed in the same way as primary instances. The read-only instance management page shows the management operations that can be performed.

View instance details by using a read-only instance

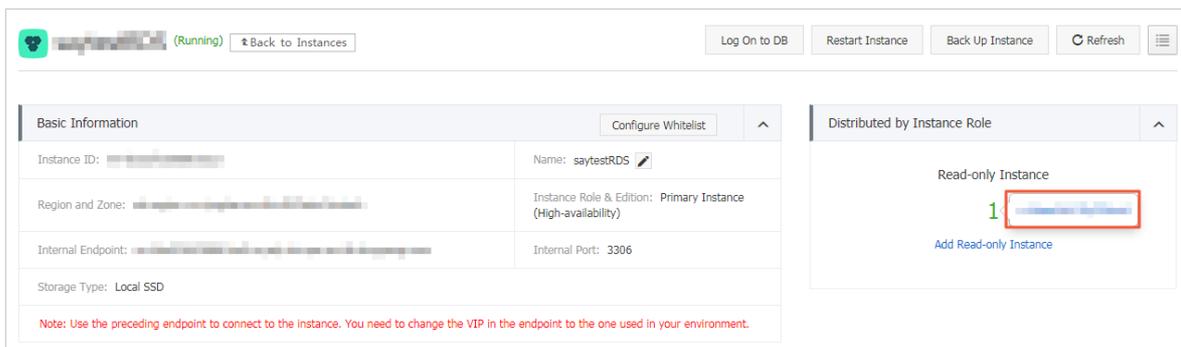
1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, click the ID of a read-only instance. The Basic Information page appears. In the instance list, Instance Role of read-only instances is displayed as Read-only Instance, as shown in [View a read-only instance](#).

View a read-only instance



View instance details by using the primary instance

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click Manage in the Actions column corresponding to the instance to go to the Basic Information page.
4. On the Basic Information page, move the pointer over the number below Read-only Instance in the Distributed by Instance Role section. The ID of the read-only instance is displayed.



5. Click the ID of the read-only instance to go to the read-only instance management page.

9.5. Accounts

9.5.1. Create an account

After you create an ApsaraDB for RDS instance and configure its whitelist, you must create a database and an account in the instance. This topic describes how to create privileged and standard accounts.

Context

ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB for RDS console. For more information about permissions that can be granted to each type of account, see [Account permissions](#).

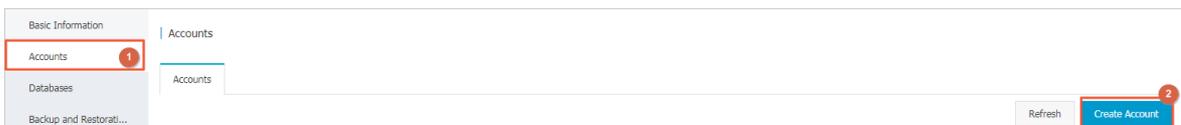
Account type	Description
--------------	-------------

Account type	Description
Privileged account	<ul style="list-style-type: none"> You can create and manage privileged accounts by using the ApsaraDB for RDS console or API operations. You can create only one privileged account on each RDS instance. The privileged account can be used to manage all standard accounts and databases on the instance. A privileged account allows you to manage permissions to a fine level. For example, you can grant each standard account the permissions to query specific tables. A privileged account has permissions to disconnect all standard accounts on the instance.
Standard account	<ul style="list-style-type: none"> You can create and manage standard accounts by using the ApsaraDB for RDS console, API operations, or SQL statements. You can create up to 500 standard accounts for an instance. You must manually grant standard accounts the specific database permissions. You cannot use a standard account to create, manage, or disconnect other accounts from databases.

Account type	Number of databases	Number of tables	Number of accounts
Privileged account	Unlimited	< 200,000	Varies based on the kernel parameter settings of the instance.
Standard account	500	< 200,000	Varies based on the kernel parameter settings of the instance.

Create a privileged account

- Log on to the ApsaraDB for RDS console.
- On the Instances page, find the target instance.
- Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
- In the left-side navigation pane, click **Accounts**.
- Click **Create Account**.



- Configure the following parameters.

Parameter	Description
Database Account	Enter the account name. The account name must meet the following requirements: <ul style="list-style-type: none"> The name must be 1 to 16 characters in length. The name must start with a letter and end with a letter or digit. The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Privileged Account.

Parameter	Description
Password	<p>Enter the password of the standard account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the privileged account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

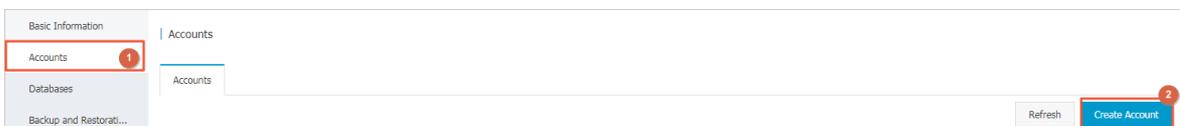
Reset the permissions of a privileged account

If an issue occurs on the privileged account, for example, permissions are unexpectedly revoked, you can enter the password of the privileged account to reset permissions.

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the privileged account, and click **Reset Permissions** in the **Actions** column.
6. Enter the password of the privileged account and click **OK**.

Create a standard account

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.



6. Configure the following parameters.

Parameter	Description
Database Account	<p>Enter the account name. The account name must meet the following requirements:</p> <ul style="list-style-type: none"> ○ The name must be 1 to 16 characters in length. ○ The name must start with a letter and end with a letter or digit. ○ The name can contain lowercase letters, digits, and underscores (_).
Account Type	Select Standard Account.

Parameter	Description
Authorized Databases	<p>Select one or more databases on which you want to grant permissions to the account. You do not have to configure this parameter at this time. You can authorize databases after the account is created.</p> <ol style="list-style-type: none"> i. Select one or more databases from the Unauthorized Databases section and click Add to add them to the Authorized Databases box. ii. In the Authorized Databases section, select the Read/Write, Read-only, DDL Only, or DML Only permissions on each authorized database. <p>If you want to grant the same permissions on multiple databases to the account, click the button in the upper-right corner of the section. The button may appear as Set All to Read/Write.</p>
Password	<p>Enter the password of the standard account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the standard account again.
Description	Optional. Enter information about the account to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

Account permissions

Account type	Authorization type	Permission				
		SELECT	INSERT	UPDATE	DELETE	CREATE
Privileged account	-	DROP	RELOAD	PROCESS	REFERENCES	INDEX
		ALTER	CREATE TEMPORARY TABLES	LOCK TABLES	EXECUTE	REPLICATION SLAVE
		REPLICATION CLIENT	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		CREATE USER	EVENT	TRIGGER	-	-
		SELECT	LOCK TABLES	SHOW VIEW	PROCESS	REPLICATION SLAVE
Read-only	-	REPLICATION CLIENT	-	-	-	-
		SELECT	INSERT	UPDATE	DELETE	CREATE

Account type	Authorization type	Permission				
	Standard account	Read/write	DROP	REFERENCES	INDEX	ALTER
LOCK TABLES			EXECUTE	CREATE VIEW	SHOW VIEW	CREATE ROUTINE
ALTER ROUTINE			EVENT	TRIGGER	PROCESS	REPLICATION SLAVE
REPLICATION CLIENT			-	-	-	-
DDL-only		CREATE	DROP	INDEX	ALTER	CREATE TEMPORARY TABLES
		LOCK TABLES	CREATE VIEW	SHOW VIEW	CREATE ROUTINE	ALTER ROUTINE
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-
DML-only		SELECT	INSERT	UPDATE	DELETE	CREATE TEMPORARY TABLES
		LOCK TABLES	EXECUTE	SHOW VIEW	EVENT	TRIGGER
		PROCESS	REPLICATION SLAVE	REPLICATION CLIENT	-	-

9.5.2. Reset the password

You can use the ApsaraDB for RDS console to reset the password of your database account.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the target account and click **Reset Password** in the **Actions** column.
6. In the dialog box that appears, enter and confirm the new password, and then click **OK**.

Note The password must meet the following requirements:

- The password must be 8 to 32 characters in length.
- The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- Special characters include ! @ # \$ % ^ & * () _ + - =

9.5.3. Modify account permissions

You can modify the account permissions of your ApsaraDB for RDS instance at any time.

Prerequisites

You can modify the permissions of a standard account. The permissions of privileged accounts can only be reset to the default settings and cannot be changed to a specific set of permissions.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the target account and click **Edit Permissions** in the **Actions** column.
6. Configure the following parameters.

Parameter	Description
Authorized Databases	In the Unauthorized Databases section, select a database and click Add to authorize the database. In the Authorized Databases section, select a database and click Remove to remove permissions from the database.
Authorized Databases	You can set permissions on each database in the Authorized Database section. You can also click the button such as Set All to Read/Write in the upper-right corner to set the permissions of the account on all authorized databases. <ul style="list-style-type: none"> ○ Read-only: grants the database read-only permissions to the account. ○ Read/Write: grants the database read/write permissions to the account. ○ DDL Only: grants the database permissions of DDL operations to the account. ○ DML Only: grants the database permissions of DML operations to the account.

7. Click **OK**.

9.5.4. Delete an account

You can delete a database account in the ApsaraDB for RDS console.

Prerequisites

You can use the console to delete privileged and standard accounts that are no longer used.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

4. In the left-side navigation pane, click **Accounts**.
5. Find the account you want to delete and click **Delete** in the **Actions** column.
6. In the message that appears, click **Confirm**.

? **Note** Accounts in the Processing state cannot be deleted.

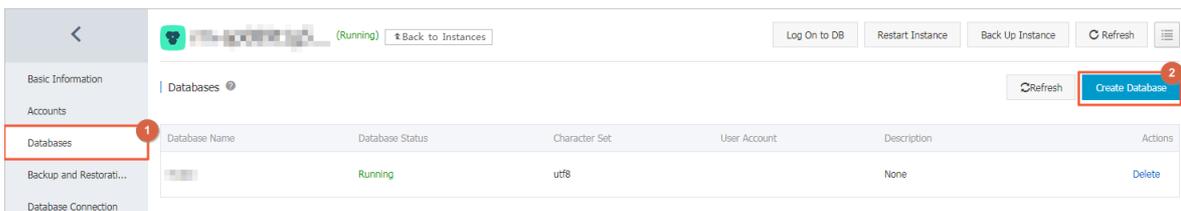
9.6. Databases

9.6.1. Create a database

After you create an ApsaraDB for RDS instance and configure its whitelist, you must create a database and an account in the instance.

Procedure

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Click **Create Database**.



6. Configure the following parameters.

Parameter	Description
Database Name	<ul style="list-style-type: none"> ◦ The name must be 1 to 64 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain lowercase letters, digits, underscores (_), and hyphens (-). ◦ Each database name must be unique in an instance.
Supported Character Set	Select utf8, gbk, latin1, utf8mb4, or all. If you want to use other character sets, select all, and then select the required character set from the list.
Description	Optional. Enter information about the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

9.6.2. Delete a database

You can delete databases that are no longer used in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Find the database you want to delete and click **Delete** in the **Actions** column.
6. In the message that appears, click **Confirm**.

9.7. Database connection

9.7.1. Change the endpoint and port number of an instance

This topic describes how to view and change the endpoint and port number of an instance.

View the endpoint and port number

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
You can view the **Internal Endpoint** and **Internal Port** of the instance in the **Database Connection** section.

Change the endpoint and port number

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Change Endpoint**.
6. In the dialog box that appears, set **Connection Type**, **Endpoint**, and **Port**, and click **OK**.

Note

- The prefix of the endpoint must be 8 to 64 characters in length and can contain letters, digits, and hyphens (-). It must start with a lowercase letter.
- The port number must be in the range of 1000 to 65534.

9.7.2. Log on to an ApsaraDB for RDS instance by using DMS

This topic describes how to log on to an ApsaraDB for RDS instance by using Data Management (DMS).

Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see [Configure a whitelist](#).

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the upper-right corner of the page, click **Log On to DB** to go to the Database Logon page.
5. On the logon page, set the following parameters:

- ①: The endpoint and port number that are used to connect to your RDS instance. The endpoint and port number are in the `<Internal endpoint>:<Internal port number>` format. Example: `rm-bpxxxxxxx.rds.aliyuncs.com:3433`. For more information, see [Change the endpoint of an instance](#).
 - ②: The account that is used to access the RDS database.
 - ③: The password of the account that is used to access the RDS database.
6. Click **Login**.

Note If you want the browser to remember the password, select **Remember your password** and click **Login**.

9.7.3. Switch the access mode

ApsaraDB for RDS supports two access modes: Standard Mode and Database Proxy (Safe Mode). This topic describes the differences between the two access modes and their configuration methods.

Prerequisites

The RDS instance runs MySQL 5.6.

Context

The Standard Mode and Database Proxy (Safe Mode) have the following differences:

- **Standard Mode:** RDS uses SLB to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.
- **Database Proxy (Safe Mode):** This mode prevents 90% of network interruptions and intercepts SQL injection attacks based on semantic analysis. However, it increases the response time by over 20%.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click Database Connection.
5. On the right side of the page, click Switch Access Mode.
6. In the message that appears, click OK.

Note When the access mode change is in progress, Status of the instance changes to Switching Links. When Status changes to Running, the access mode is changed.

9.7.4. Hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB for RDS to change the network type of an instance from classic network to Virtual Private Network (VPC) without network interruptions.

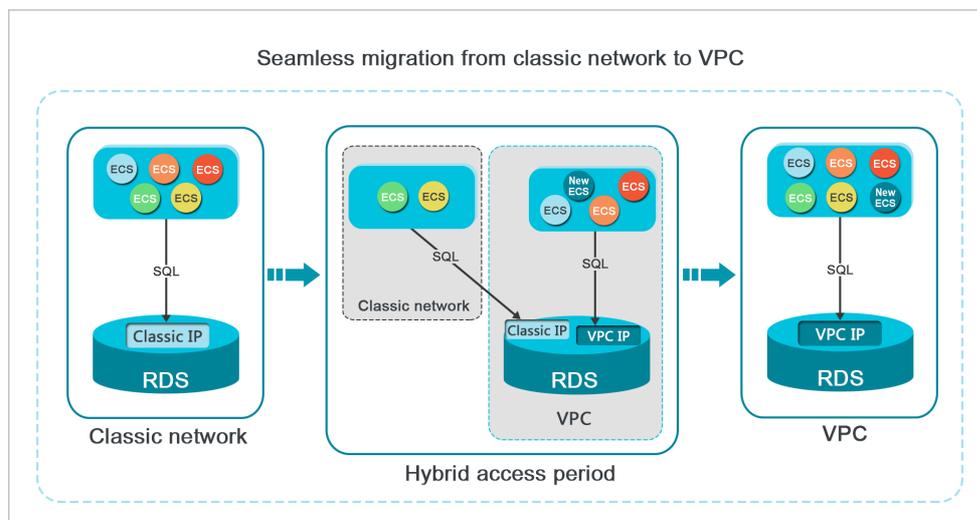
Background

In the past, when you changed the network type of an RDS instance from classic network to VPC, the internal endpoint of the RDS instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the RDS instance through the internal endpoint within this period. To smoothly change the network type, ApsaraDB for RDS provides the hybrid access solution.

Hybrid access refers to the ability of an RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimize the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database through the internal endpoint of VPCs, and the other applications can access the database through the original internal endpoint of the classic network. When all the applications access the database through the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- Changing to the classic network is not supported.
- Migrating the RDS instance to another zone is not supported.

Prerequisites

- The network type of the instance is classic network.
- Available VPCs and VSwitches exist in the zone where the instance resides.

Change the network type from classic network to VPC

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. Click **Switch to VPC**.
6. In the dialog box that appears, select a VPC and a VSwitch and specify whether to retain the endpoint used in the classic network.

Determine whether to select the **Reserve Original Classic Network Endpoint** option. The following table describes the details.

Operation	Description
Clear the Reserve Original Classic Network Endpoint option	The endpoint used in the classic network is replaced with an endpoint in the VPC. When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the RDS instance are interrupted.
Select the Reserve Original Classic Network Endpoint option	The endpoint used in the classic network is retained, and a new endpoint to be used in the VPC is generated. In such cases, the RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the RDS instance over the internal network. When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the RDS instance will be available until the endpoint used in the classic network expires. Specify the expiration time of the classic network endpoint. You must add the new VPC endpoint to the ECS instance before the endpoint in the classic network expires. This ensures smooth network switchover.

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the RDS instance over the internal network. If no VPC whitelists are available, create a whitelist. For more information, see [Configure a whitelist](#).

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be connected over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

Follow these steps to change the expiration time:

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. Click **Change Expiration Time**.
6. In the **Change Expiration Time** dialog box, select an expiration time and click **OK**.

9.7.5. Change the network type of an instance

This topic describes how to change the network type of an ApsaraDB for RDS instance between classic network and VPC.

Context

- **Classic network:** RDS instances in the classic network are not isolated. Unauthorized access to these instances can be blocked only by whitelists.
- **VPC:** Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you select the VPC network type because it is more secure than the classic network.

You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can use leased lines or VPNs to connect on-premises data center to a VPC to create a virtual data center.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the internal endpoint of the RDS instance remains unchanged, but the IP address bound to the internal endpoint changes.
 - After you change the network type from VPC to classic network, ECS instances in the same VPC as the RDS instance can no longer connect to the RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
 - When you change the network type, a 30-second network interruption may occur. To avoid business interruption, change the network type during off-peak hours or make sure that your applications are configured with automatic reconnection policies.
1. [Log on to the ApsaraDB for RDS console](#).
 2. On the **Instances** page, find the target instance.
 3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
 4. In the left-side navigation pane, click **Database Connection**.
 5. In the upper-right corner of the **Database Connection** section, click **Switch to Classic Network**.
 6. In the message that appears, click **OK**.

Change the network type from classic network to VPC

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Switch to VPC**.
6. In the **Switch to VPC** dialog box, select a VPC and VSwitch and specify whether to **Reserve Original Classic Network Endpoint**. Click **OK**. For more information about **Reserve Original Classic Network Endpoint**, see [Hybrid access from both the classic network and VPCs](#).

9.8. Database proxy

9.8.1. Database proxy

The database proxy is a high-performance database middleware provided by ApsaraDB for RDS. It is secure, stable, fully compatible with database protocols, and transparent to clients.

Prerequisites

The RDS instance runs MySQL 5.6 High-availability Edition.

Principle

The database proxy sits between a client and the database engine, and is automatically maintained by ApsaraDB for RDS. All database requests and responses are processed by the proxy.

Features

The database proxy supports the read/write splitting function. You do not need to modify code of the client. Queries are distributed to read-only RDS instances to reduce the load on the primary instance. For more information, see [Enable read/write splitting](#).

Enable the database proxy feature

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Database Proxy** tab, turn on the **Disabled** switch to the right of **Database Proxy Status**.
6. In the message that appears, click **Confirm**.

9.8.2. Dedicated proxy

This topic describes the dedicated proxy feature of ApsaraDB RDS for MySQL. This feature provides advanced functions such as read/write splitting, connection pool, and transaction splitting.

Prerequisites

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.7 on RDS Enterprise Edition
- MySQL 5.7 on RDS High-availability Edition (with local SSDs)

Context

The dedicated proxy feature uses dedicated computing resources. It has the following benefits:

- A unified proxy endpoint is provided to connect to all of the dedicated proxies that are enabled on the primary RDS instance. This reduces maintenance costs by relieving you from updating the endpoints on your application. The proxy endpoint remains valid unless you release the dedicated proxy instance. For example, you may enable read/write splitting during peak hours, but then release read-only instances and disable read/write splitting after peak hours. In such cases, you do not need to update the endpoints on your application because the proxy endpoint is still connected.
- Dedicated proxies exclusively serve the primary RDS instance and its read-only RDS instances. This ensures service stability by relieving you from competing with other users for resources.
- Dedicated proxies support scaling. You can add dedicated proxies based on your business requirements to handle more workloads.

Limits

- Dedicated proxies do not support SSL encryption.

- Dedicated proxies do not support compression protocols.

Precautions

- While you change the specifications of the primary RDS instance or its read-only RDS instances, a network interruption may occur.
- If you connect your application to the proxy endpoint without enabling the transaction splitting function, all requests encapsulated in transactions are routed to the primary RDS instance.
- If you use the proxy endpoint to implement read/write splitting, the read consistency of the requests that are not encapsulated in transactions cannot be guaranteed. If you require this read consistency, you must encapsulate these requests in transactions.
- If you connect your application to the proxy endpoint, the `SHOW PROCESSLIST` statement returns a combination of results from the primary RDS instance and all of its read-only RDS instances.
- If you execute **multi-statements** or run stored procedures, the read/write splitting function is disabled and all subsequent requests over the current connection are routed to the primary RDS instance. To enable the read/write splitting function again, you must close the current connection and establish a new one.
- The dedicated proxy feature supports the `/*FORCE_MASTER*/` and `/*FORCE_SLAVE*/` hints. However, requests that contain hints have the highest route priorities, and therefore they are not constrained by consistency or transaction limits. You must check whether these hints are suitable for your workloads before you use them. In addition, these hints cannot contain statements such as `/*FORCE_SLAVE*/ set names utf8;` . Such statements can change environment variables. If you include such statements in these hints, errors may occur when you process your subsequent workloads.
- After you enable the dedicated proxy feature, each connection is replicated to the primary RDS instance and all of its read-only RDS instances in compliance with the 1:N connection model. We recommend that you specify the same connection specifications for these instances. If these instances have different connection specifications, the number of connections allowed is subject to the lowest connection specifications among these instances.
- If you create or reboot a read-only RDS instance after you enable the dedicated proxy feature, only the requests over new connections are routed to the new or rebooted read-only RDS instance.
- The `max_prepared_stmt_count` parameter must be set to the same value for the primary RDS instance and all of its read-only RDS instances.

Enable the dedicated proxy feature

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. Click **Enable now**.

Overview of the Database Proxy page

After the dedicated proxy feature is enabled, you can use the generated proxy endpoint to implement functions such as read/write splitting, connection pool, and transaction splitting.

The screenshot shows the configuration page for a Proxy Service. It has tabs for 'Proxy Service', 'Read/Write Splitting', and 'Monitoring Data'. A 'Disable Proxy Service' button is in the top right. The 'Proxy Endpoint' section shows: Status: Running; Endpoint: [redacted] Copy Address; Port: 3306; Instance ID: [redacted]; Read/Write Splitting: Disabled; Transaction Splitting: Enabled Disable; Endpoint Type: Internal (Classic Network); Instances Associated with Proxy: 1; Short-Lived Connection Optimization: Disabled Enable. The 'Proxy Instance' section shows a table with columns: Proxy Type, CPU and Memory, Instances Associated, Adjusted Instances, and Adjustment Plan. The table contains one row: Dedicated Proxy, 2 Cores, 4 GB, 1, - 1 +, and buttons for Apply and Cancel.

Section	Parameter	Description
Proxy Endpoint	Instance ID	The ID of the proxy instance.
	Instances Associated with Proxy	The number of dedicated proxies that are enabled on the primary RDS instance. You can add dedicated proxies to process more queries. After the public preview phase ends, you must pay for added proxy instances.
	Read/Write Splitting	Specifies whether to enable read/write splitting for the proxy endpoint. For more information, see Read/write splitting .
	Short-Lived Connection Optimization	Specifies to enable or disable connection pool for the proxy endpoint. This function is suitable for scenarios that PHP short-lived connections are established. For more information, see Short-lived connection optimization . Note You can click Enable or Disable to the right of Short-Lived Connection Optimization to enable or disable this function.
	Transaction Splitting	Specifies whether to enable the transaction splitting function for the proxy endpoint. For more information, see Transaction splitting . Note You can click Enable or Disable to the right of Transaction Splitting to enable or disable this feature.
	Endpoint	The proxy endpoint that is generated after the dedicated proxy feature is enabled. This endpoint connects to all of the dedicated proxies that are enabled on the primary RDS instance. The read/write splitting function is also bound to this endpoint. Note You can click Copy Address to the right of Endpoint to copy the endpoint.
	Port	The port that is associated with the proxy endpoint.
	Endpoint Type	The network type of the proxy endpoint.
	Proxy Type	The type of proxy that is enabled on the primary RDS instance. Only the Dedicated Proxy type is supported.

Section	Parameter	Description
Proxy Instance	CPU and Memory	The CPU and memory of proxy instances. Only 2 Cores, 4 GB is supported.
	Instances Associated	<p>The total number of dedicated proxies that are enabled on the primary RDS instance. Up to 60 dedicated proxies are supported.</p> <p> Note We recommend that you specify the number of dedicated proxies as the rounded-up integer of the total number of CPU cores of the primary and read-only RDS instances divided by 8.</p> <p>For example, if the primary RDS instance has eight CPU cores and its read-only RDS instances have four CPU cores, the recommended number of dedicated proxies is 2 based on the following formula: $(8 + 4)/8 = 1.5$ (rounded up to 2).</p>

Adjust the number of dedicated proxies

 **Note** While you adjust the number of dedicated proxies, a network interruption will occur. Make sure that your applications are configured with automatic reconnection policies.

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. In the **Proxy Instance** section of the **Proxy Service** tab, change the number in the **Adjusted Instances** column and click **Apply** in the **Adjustment Plan** column.
6. In the dialog box that appears, you can click **Switch Now** to apply the changes. You can also click **Switch Within Maintenance Window** to set a time point for the change to take effect. Click **OK**.

View the monitoring data of dedicated proxies

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. Click the **Monitoring Data** tab.
6. Select a time range and view the **CPU Utilization (%)** metric within that time range.

Disable the dedicated proxy feature

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. In the upper-right corner of the page, click **Disable Proxy Service**.
6. In the message that appears, click **OK**.

9.8.3. Short-lived connection optimization

This topic introduces the short-lived connection optimization function provided by ApsaraDB RDS for MySQL in its dedicated proxy feature.

Prerequisites

- The primary RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.7 on RDS Enterprise Edition
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- The database proxy is enabled for the instance. For more information, see [Dedicated proxy](#).

Context

The short-lived connection optimization function is used to reduce workloads on the RDS instance caused by frequent short-lived connections. If a client is disconnected from a connection, the system determines whether the connection is idle. If the connection is idle, the proxy retains the connection in the connection pool for a short period. When the client initiates a connection request again, the proxy matches the request with idle connections retained in the connection pool based on the values of the user, clientip, and dbname fields. If an idle connection in the connection pool is matched, the proxy uses this idle connection. If no idle connection is matched, a new connection is established with the database. This reduces the overheads of database connections.

 **Note** The short-lived connection optimization function does not reduce concurrent connections with the database. It reduces the frequency to establish connections between the application and database and workloads of the primary MySQL thread. This improves efficiency to process business requests. However, idle connections in the connection pool still occupy the database threads for a short period of time.

Precautions

You cannot configure different permissions for the same account with different source IP addresses. Otherwise, errors may occur when connections in the connection pool are reused. For example, if the user account has permissions on database_a when its source IP address is 192.168.1.1 but does not have permissions on database_a when its source IP address is 192.168.1.2, the short-lived connection optimization function may encounter permission errors.

Enable short-lived connection optimization

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Proxy Service** tab, click **Enable** on the right of **Short-Lived Connection Optimization**.

9.8.4. Transaction splitting

This topic introduces the transaction splitting function provided by the database proxy of ApsaraDB for RDS. This function identifies and distributes read requests initiated before write requests within a transaction to read-only instances. This reduces workloads on the primary instance.

Prerequisites

- The primary RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.7 on RDS Enterprise Edition
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)

- The database proxy is enabled for the instance. For more information, see [Dedicated proxy](#).

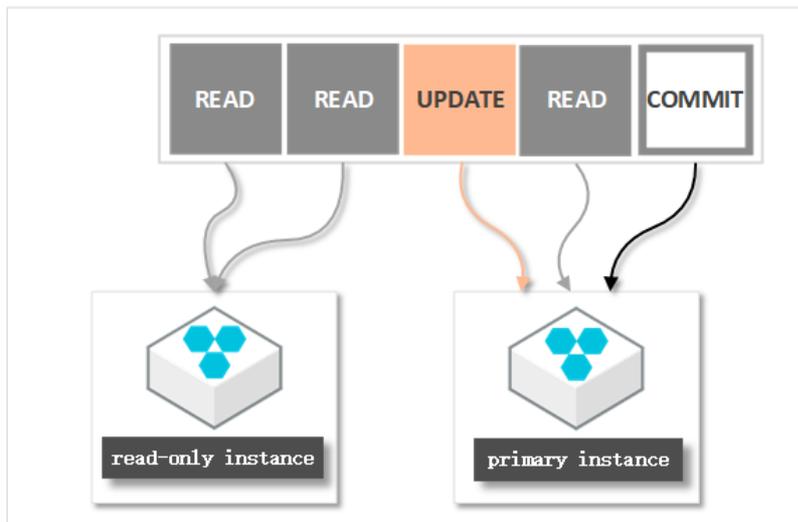
Context

By default, the dedicated proxy sends all requests in transactions to the primary instance to ensure the correctness of the transactions. If the framework used encapsulates all requests in transactions, the primary instance becomes heavily loaded. In this case, you can enable the transaction splitting function.

When transaction splitting is enabled and the default isolation level READ COMMITTED is used, the RDS instance starts a transaction only for write requests when autocommit is disabled (set autocommit=0). Read requests that arrive before the transaction is started are distributed to read-only instances by the load balancer.

Note

- Explicit transactions do not support splitting, such as transactions started by using the BEGIN or START statement.
- After transaction splitting is enabled, global consistency cannot be ensured. If your business requires global consistency, we recommend that you evaluate whether you can enable transaction splitting.



Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Proxy Service** tab, click **Enable** on the right of **Transaction Splitting**.

Note

- When you no longer need transaction splitting, you can click **Disable** on the right of **Transaction Splitting**.
- The operation to enable or disable transaction splitting takes effect only on new connections.

9.8.5. Read/write splitting

9.8.5.1. Enable read/write splitting

This topic describes the read/write splitting function of ApsaraDB RDS for MySQL in its dedicated proxy feature and how to enable this function.

Prerequisites

- The RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.6
 - MySQL 5.7 on RDS Enterprise Edition
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- You have enabled the database proxy or dedicated database proxy. For more information, see [Enable the database proxy feature](#) and [Enable the dedicated proxy feature](#).
- At least one read-only instance is created. For more information about how to create a read-only instance, see [Create a read-only instance](#).

Context

If your primary instance needs to process a large number of read requests but only a small number of write requests, you can create one or more read-only instances to offload read requests from your primary instance. This allows you to ensure service stability. For more information, see [Create a read-only instance](#).

After you create read-only instances, you can enable read/write splitting. In this case, a read/write splitting endpoint is provided. After you add the endpoint to your application, write requests are routed to the primary instance and read requests are routed to the read-only instances.

□

Differences between the read/write splitting endpoint and the internal and public endpoints

After you enable read/write splitting and add the read/write splitting endpoint to your application, all requests are routed to this endpoint, and then to the primary and read-only instances based on the request types and the read weights of these instances.

If the internal or public endpoint of the primary instance is added to your application, all requests are routed to the primary instance. In this case, you must add the endpoints and read weights of the primary and read-only instances to your application to implement read/write splitting.

Logic to route requests

- The following requests are routed only to the primary instance:
 - Data manipulation language (DML) statements, which are INSERT, UPDATE, DELETE, and SELECT FOR UPDATE
 - All data definition language (DDL) statements used to perform operations such as creating databases or tables, deleting databases or tables, and changing schemas or permissions
 - All requests that are encapsulated in transactions
 - Requests for user-defined functions
 - Requests for stored procedures
 - Requests for EXECUTE statements
 - Requests for [multi-statements](#)
 - Requests that involve temporary tables
 - Requests for SELECT last_insert_id() statements
 - All requests to query or modify user environment variables
 - Requests for SHOW PROCESSLIST statements
 - All requests for KILL statements in SQL (Note that these are not KILL commands in Linux.)
- The following requests are routed to the primary instance or its read-only instances:

- Read requests that are not encapsulated in transactions
- Requests for `COM_STMT_EXECUTE` statements
- The following requests are routed to all of the primary and read-only instances:
 - All requests to modify system environment variables
 - Requests for `USE` statements
 - Requests for `COM_STMT_PREPARE` statements
 - Requests for `COM_CHANGE_USER`, `COM_QUIT`, and `COM_SET_OPTION` statements

Benefits

- Easier maintenance with a unified endpoint

If you do not enable read/write splitting, you must add the endpoints of the primary and read-only instances to your application. This makes write requests routed to the primary instance and read requests routed to the read-only instances.

If you enable read/write splitting, the endpoint of the dedicated proxy is used for read/write splitting. After your application is connected to this endpoint, requests are routed to the primary and read-only instances based on the read weights of these instances. This reduces maintenance costs.

In addition, you can scale up the read capability of your database system by creating read-only instances. This relieves you from the need to modify the configuration data on your application.

- Higher performance and lower maintenance cost with a native RDS link

If you build a separate proxy layer on the cloud to implement read/write splitting, statements need to be parsed and forwarded by a number of components before they reach your database system. This increases response latency. The read/write splitting function provided with ApsaraDB for RDS shortens response latency, increases processing speed, and reduces maintenance costs.

- Ideal in various use scenarios with configurable read weights and thresholds

You can specify the read weights of the primary and read-only instances. You can also specify a latency threshold for each read-only instance.

- Highly available with instance-level health check

The read/write splitting module actively performs health checks on the primary and read-only instances. If an instance breaks down or its latency exceeds the specified threshold, the read/write splitting module stops routing requests to the instance and redirects the requests that were destined for the instance to other healthy instances. This allows you to ensure service availability in the event of faults on a single instance. After the instance is recovered, the read/write splitting module resumes routing read requests to it.

 **Note** To avoid single points of failure (SPOFs), we recommend that you create at least two read-only instances.

Precautions

- A network interruption may occur while you change the specifications of the primary instance or its read-only instances.
- After you create a read-only instance, only the requests over new connections can be routed to the read-only instance.
- The endpoint of the dedicated proxy does not support SSL encryption.
- The endpoint of the dedicated proxy does not support compression.
- If the endpoint of the dedicated proxy is used for connection, all of the requests encapsulated in transactions are routed to the primary instance.
- If the endpoint of the dedicated proxy is used for read/write splitting, the read consistency of the requests that are not encapsulated in transactions cannot be ensured. If you require the read consistency, encapsulate the requests in transactions.

- If the endpoint of the dedicated proxy is used for connection, the `SHOW PROCESSLIST` statement combines the results from the primary and read-only instances and returns a result set.
- If short-lived connection optimization is enabled, the `SHOW PROCESSLIST` statement may return idle connections.
- If you execute **multi-statements** or stored procedures, read/write splitting is disabled and all of the subsequent requests over the current connection are routed to the primary instance. To enable read/write splitting again, you must terminate the current connection and establish a new one.
- The `/*FORCE_MASTER*/` and `/*FORCE_SLAVE*/` hints are supported. However, requests that contain hints have higher route priorities. These requests are not constrained by consistency or transaction limits. You must check whether these hints are suitable for your business before you use them. A hint cannot contain statements that change environment variables. An example is `/*FORCE_SLAVE*/ set names utf8;`. Otherwise, an error may occur in the subsequent procedure.

Prerequisites

A read-only instance is created for the primary instance. For more information, see [Create a read-only instance](#).

Enable read/write splitting

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Enable now**.
6. Configure the following parameters.

Parameter	Description
Latency Threshold	<p>The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the replication latency for a read-only instance exceeds the specified threshold, the read/write splitting module stops routing read requests to the read-only instance. This applies even if the read-only instance has a high read weight.</p> <p>Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a certain latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.</p>
Read Weight Distribution	<p>The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200. In this situation, the primary instance only processes write requests, and the three read-only instances process all of the read requests at the 1:2:2 ratio.</p> <ul style="list-style-type: none"> ◦ Automatic Distribution: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. ◦ Customized Distribution: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, its read weight defaults to 0. You must manually specify the read weight of the read-only instance.

7. Click **OK**.

9.8.5.2. Set the read/write splitting parameters

This topic describes how to configure the latency threshold and specify read weights for an ApsaraDB for RDS instance in the RDS console.

Prerequisites

- The RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.6
 - MySQL 5.7 on RDS Enterprise Edition
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- Read/write splitting is enabled. For more information, see [Enable read/write splitting](#).

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Configure Read/Write Splitting**.
6. Configure the following parameters.

Parameter	Description
Latency Threshold	<p>The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the replication latency for a read-only instance exceeds the specified threshold, the read/write splitting module stops routing read requests to the read-only instance. This applies even if the read-only instance has a high read weight.</p> <p>Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a certain latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.</p>
Read Weight Distribution	<p>The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200. In this situation, the primary instance only processes write requests, and the three read-only instances process all of the read requests at the 1:2:2 ratio.</p> <ul style="list-style-type: none"> ◦ Automatic Distribution: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. ◦ Customized Distribution: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, its read weight defaults to 0. You must manually specify the read weight of the read-only instance.

7. Click **OK**.

9.8.5.3. Disable read/write splitting

This topic describes how to disable the read/write splitting feature of an ApsaraDB for RDS instance in the RDS console.

Prerequisites

- The RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.6
 - MySQL 5.7 on RDS Enterprise Edition
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- You have enabled read/write splitting. For more information, see [Enable read/write splitting](#).

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the **Read/Write Splitting** tab, click **Disable Read/Write Splitting**.
6. Click **OK**.

9.9. Monitoring and alerts

9.9.1. View resource and engine monitoring data

The ApsaraDB for RDS console provides a variety of performance metrics to monitor the status of your instances.

Prerequisites

The RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- MySQL 5.7 on RDS Enterprise Edition

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. Find the target instance and click the instance ID.
3. In the left-side navigation pane, click **Monitoring and Alerts**.
4. On the **Monitoring and Alerts** page, select **Resource Monitoring** or **Engine Monitoring**, and select a time range to view the corresponding monitoring data. The following table describes the metrics.

Category	Metric	Description
Resource Monitoring	Disk Space (MB)	<p>The disk space usage of the instance. It consists of the following parts:</p> <ul style="list-style-type: none"> ◦ Instance size ◦ Data usage ◦ Log size ◦ Temporary file size ◦ Other system file size <p>Unit: MB.</p>
	IOPS	The number of I/O requests per second for the instance.
	Total Connections	The total number of current connections to the instance, including the number of active connections and the total number of connections.
	CPU Utilization and Memory Usage (%)	The CPU utilization and memory usage of the RDS instance. These metrics do not include the CPU utilization and memory usage for the operating system.

Category	Metric	Description
	Network Traffic (KB)	The inbound and outbound traffic of the instance per second. Unit: KB.
Engine Monitoring	TPS/QPS	The average number of transactions per second and the average number of SQL statements executed per second.
	InnoDB Buffer Pool Read Hit Ratio, Usage Ratio, and Dirty Block Ratio (%)	The read hit ratio, usage ratio, and dirty block ratio of the InnoDB buffer pool.
	InnoDB Read/Write Volume (KB)	The amount of data that InnoDB reads and writes per second. Unit: KB.
	InnoDB Buffer Pool Read/Write Frequency	The number of read and write operations that InnoDB performs per second.
	InnoDB Log Read/Write/fsync	The average frequency of physical writes to log files per second by InnoDB, the log write request frequency, and the average frequency of fsync writes to log files.
	Temporary Tables Automatically Created on Hard Disk when MySQL Statements Are Executed	The number of temporary tables that are automatically created on the hard disk when the database executes SQL statements.
	MySQL_COMDML	The number of SQL statements that the database executes per second. The following SQL statements are included: <ul style="list-style-type: none"> ◦ Insert ◦ Delete ◦ Insert_Select ◦ Replace ◦ Replace_Select ◦ Select ◦ Update
	MySQL_RowDML	The numbers of operations that InnoDB performs per second. The following items are included: <ul style="list-style-type: none"> ◦ The number of physical writes to log files per second ◦ The number of rows that are read, updated, deleted, and inserted from InnoDB tables per second
	MyISAM Read/Write Frequency	The numbers of operations that MyISAM performs per second. The following items are included: <ul style="list-style-type: none"> ◦ The number of MyISAM reads and writes from the buffer pool per second ◦ The number of MyISAM reads and writes from the hard disk per second
MyISAM Key Buffer Read/Write/Usage Ratio (%)	The read hit ratio, write hit ratio, and usage of the MyISAM key buffer per second.	

9.9.2. Set a monitoring frequency

The ApsaraDB for RDS console provides a variety of performance metrics for which you can set a monitoring frequency.

Prerequisites

The RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition (with local SSDs)
- MySQL 5.7 on RDS Enterprise Edition

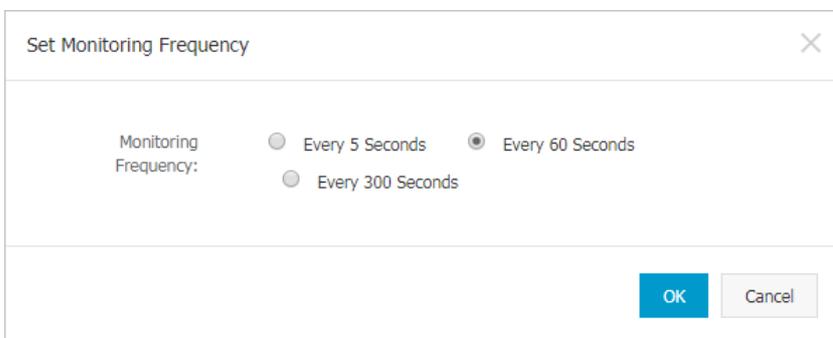
Context

ApsaraDB for RDS provides the following monitoring frequencies:

- Every 5 seconds for the first seven days. After the seventh day, performance metrics are monitored every minute.
- Every 60 seconds.
- Every 300 seconds.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Resource Monitoring** page, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select a new monitoring frequency.



7. Click **OK**.

Note If your instance does not support the selected monitoring frequency, a prompt appears in the Set Monitoring Frequency dialog box. Select a monitoring frequency supported by the instance.

9.10. Data security

9.10.1. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB for RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB for RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB for RDS instance.

To configure a whitelist, perform the following operations:

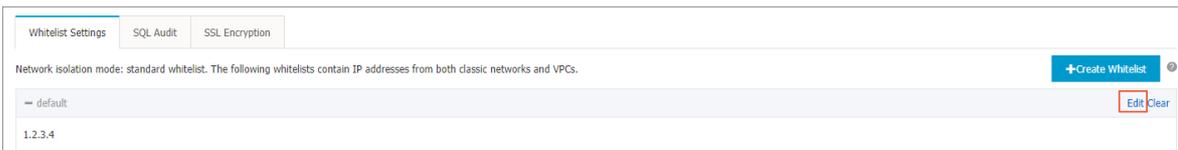
- Configure a whitelist: Add IP addresses to allow them to connect to the RDS instance.
- Configure an ECS security group: Add an ECS security group for the RDS instance to allow ECS instances in the group to connect to the RDS instance.

Precautions

- The default whitelist can be modified or cleared, but cannot be deleted.
- You can add up to 1,000 IP addresses or CIDR blocks to a whitelist. If you want to add a large number of IP addresses, we recommend that you merge them into CIDR blocks, such as 192.168.1.0/24.

Configure a standard IP address whitelist

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.



Note

- If you want to connect an ECS instance to an ApsaraDB for RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
- You can click **Create Whitelist** to create a new whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**.
 - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - After you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances under your Apsara Stack account are displayed. You can select the required IP addresses and add them to the whitelist.

Note If you add a new IP address or CIDR block to the default whitelist, the default address 127.0.0.1 is automatically deleted.

9.10.2. Configure SSL encryption

This topic describes how to enhance endpoint security. You can enable Secure Sockets Layer (SSL) encryption and install SSL certificates issued by certificate authorities (CAs) on the required application services. SSL is used at the transport layer to encrypt network connections and enhance the security and integrity of communication data. SSL also increases the response time.

Prerequisites

The RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition

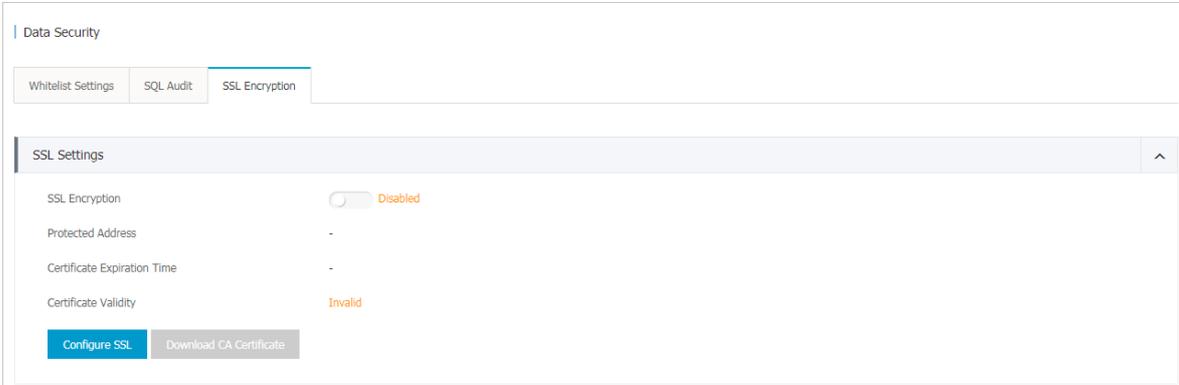
Precautions

- An SSL CA certificate is valid for one year. You must update the validity period of the certificate and then download and configure the certificate again. Otherwise, clients that use encrypted connections cannot connect to the RDS instance.
- SSL encryption may cause a significant increase in CPU utilization. We recommend that you enable SSL encryption only when you want to encrypt connections from the Internet. In most cases, connections that use an internal endpoint do not require SSL encryption.
- Read/write splitting endpoints do not support SSL encryption.
- If you disable SSL encryption, the RDS instance restarts. Proceed with caution.

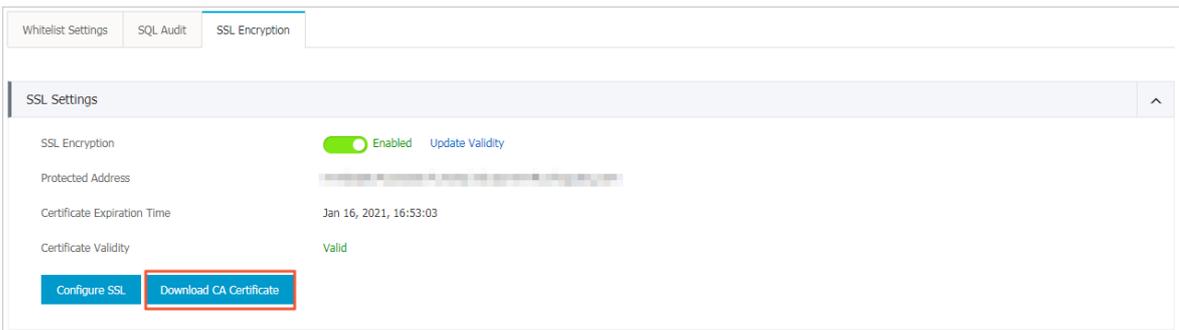
Enable SSL encryption

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.



6. Turn on the switch next to **Disabled**.
7. In the **Configure SSL** dialog box, select the endpoint for which you want to enable SSL encryption and click **OK**.
8. Click **Download CA Certificate** to download the SSL CA certificate files in a compressed package.



The downloaded package contains the following files:

- **P7B file:** used to import CA certificates to the Windows operating system.
- **PEM file:** used to import CA certificates to other operating systems or applications.
- **JKS file:** the Java truststore file. The password is `apsaradb`. It is used to import the CA certificate chain to Java programs.

Note When the JKS file is used in Java, you must modify the default JDK security configuration in JDK 7 and JDK 8. Open the `/jre/lib/security/java.security` file on the host where your application resides, and modify the following configurations:

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
jdk.certpath.disabledAlgorithms=MD2, RSA keySize < 1024
```

If you do not modify the JDK security configuration, the following error is reported. Similar errors are also caused by the Java security configuration.

```
javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
```

Configure an SSL CA certificate

After you enable SSL encryption, configure the SSL CA certificate on your application or client before they can connect to the RDS instance. This section uses MySQL Workbench and Navicat as examples to describe how to configure an SSL CA certificate. For more information, see the instructions for the other applications or clients.

Configure a certificate on MySQL Workbench

1. Start MySQL Workbench.
2. Choose **Database > Manage Connections**.
3. Enable **Use SSL** and import the SSL CA certificate files.

Configure a certificate on Navicat

1. Start Navicat.
2. Right-click the target database and select **Edit Connection**.
3. Click the **SSL** tab. Select the path of the SSL CA certificate file in the .pem format.
4. Click **OK**.

 **Note** If the `connection is being used` error is reported, the previous session is still connected. Restart Navicat.

5. Double-click the target database to test whether the database is connected.

Update the validity period of an SSL CA certificate

 **Note** Update Validity causes the RDS instance to restart. Proceed with caution.

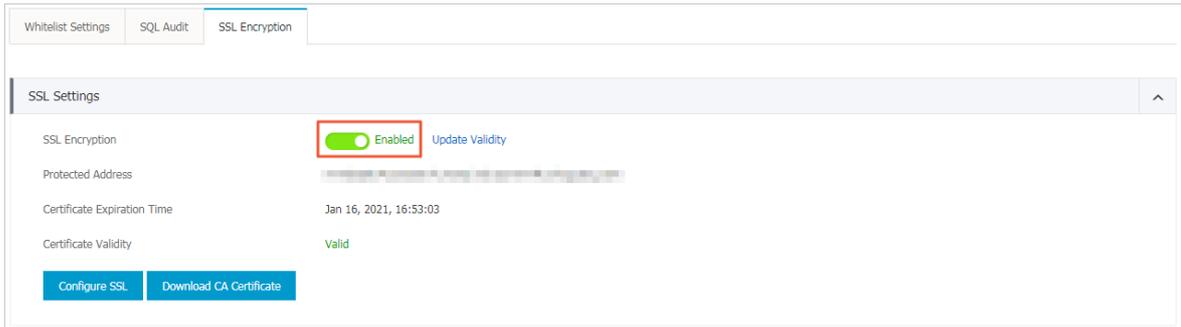
1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. Click **Update Validity**.

Disable SSL encryption

 **Note**

- If you disable SSL encryption, the RDS instance restarts. To reduce the impact on your business, the system triggers a primary/secondary switchover. We recommend that you disable SSL encryption during off-peak hours.
- Database access features higher performance but lower security after SSL encryption is disabled. We recommend that you disable SSL encryption only in secure environments.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. Turn off the switch next to **Enabled**. In the message that appears, click **OK**.



9.10.3. Configure TDE

Transparent data encryption (TDE) encrypts and decrypts data files in real time. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the size of data files. You can use TDE without making changes to applications.

Prerequisites

- Your RDS instance runs one of the following MySQL versions and RDS editions:
 - MySQL 5.7 on RDS High-availability Edition (with local SSDs)
 - MySQL 5.6
- Key Management Service (KMS) is activated. If KMS is not activated, you can activate it as prompted when you enable TDE.

Context

Encryption keys are created and managed by KMS. RDS does not provide the keys and certificates that are required for encryption. For specific zones, you can use the keys that are automatically generated by Alibaba Cloud or use your own key materials to generate data keys, and then authorize your RDS instance to use these keys.

Precautions

- Enabling TDE restarts your RDS instance and terminates all of its connections. Make appropriate service arrangements before you enable TDE. Proceed with caution.
- After TDE is enabled, it cannot be disabled.
- After TDE is enabled, you cannot change the key.
- After TDE is enabled, if you want to restore data to your computer, you must **decrypt data** on your RDS instance.
- After TDE is enabled, CPU utilization significantly increases.
- If you use an existing custom key, note the following points:
 - If you disable a key, set a key deletion plan, or delete the key materials, the key becomes unavailable.
 - After you revoke the key that is authorized for an RDS instance, the RDS instance becomes unavailable after the instance is restarted.
 - You must use an Apsara Stack account or an account that has the AliyunSTSAssumeRoleAccess permission.

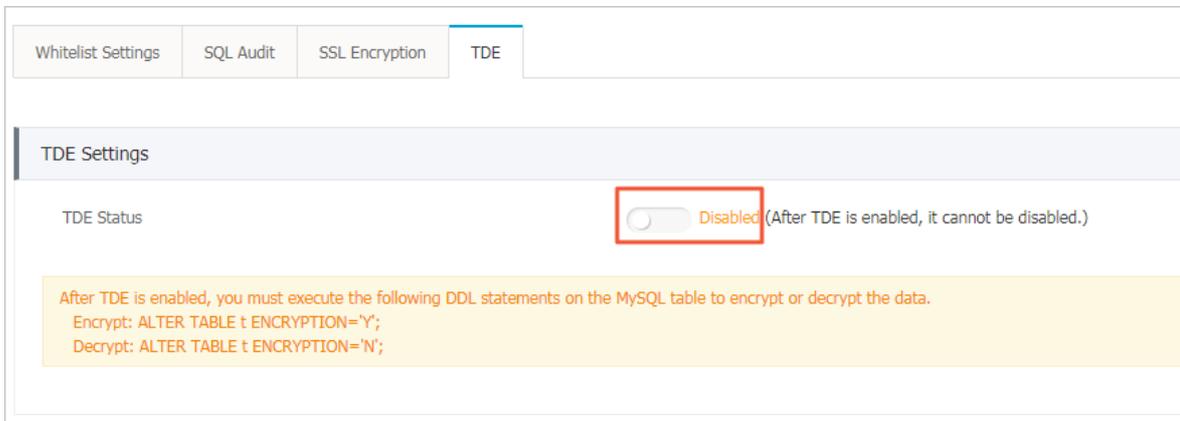
 **Note** Use the following path to navigate the key management guide: *Key Management Service > User Guide*.

Use a key that is automatically generated by Alibaba Cloud

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

- In the left-side navigation pane, click **Data Security**.
- Click the **TDE** tab. Then, turn on **TDE Status**.



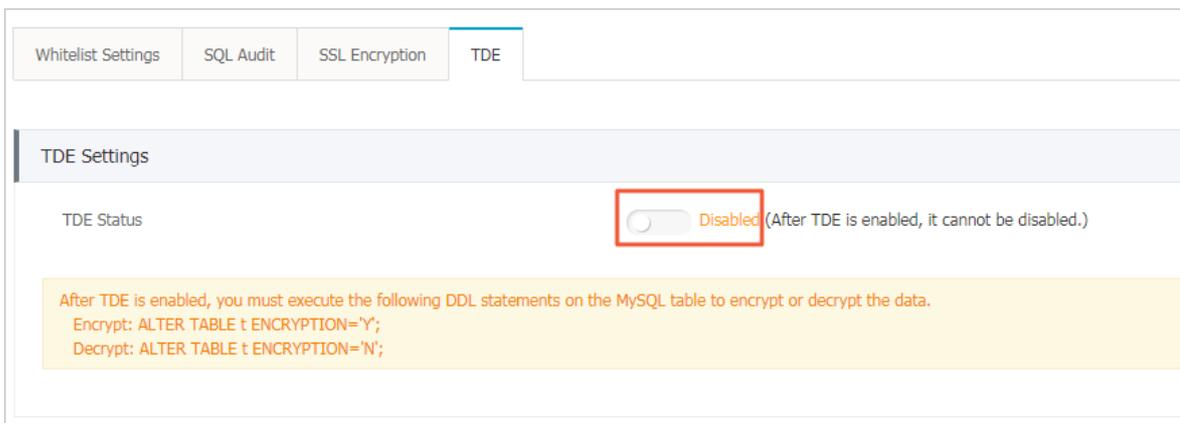
- In the dialog box that appears, select **Use an Automatically Generated Key** and click **OK**.

Note If the instance runs MySQL 5.7 on RDS High-availability Edition, you can select one of the following encryption methods:

- SM4 Encryption
- AES_256_CBC Encryption

Use an existing custom key

- Log on to the **ApsaraDB for RDS console**.
- On the **Instances** page, find the target instance.
- Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
- In the left-side navigation pane, click **Data Security**.
- Click the **TDE** tab. Then, turn on **TDE Status**.



- In the dialog box that appears, select **Use an Existing Custom Key** and click **OK**.

Note If you do not have a custom key, click **create a key** to go to the KMS console and import the key materials. Use the following path to navigate the guide: *Key Management Service > User Guide > Create a key*.

Encrypt a table

Log on to the target database and execute the following statement to encrypt a table:

- MySQL 5.6

```
alter table <tablename> engine=innodb,block_format=encrypted;
```

- MySQL 5.7

```
alter table <tablename> engine=innodb,encryption='Y';
```

Decrypt a table

Execute the following statement to decrypt a table that is encrypted with TDE:

- MySQL 5.6

```
alter table <tablename> engine=innodb,block_format=default;
```

- MySQL 5.7

```
alter table <tablename> engine=innodb,encryption='N';
```

FAQ

- Q: Can common database tools such as Navicat be used after TDE is enabled?

A: Yes, common database tools such as Navicat can be used after TDE is enabled.

- Q: Why is data still in plaintext after it is encrypted?

A: Data is stored in ciphertext. However, when you query it, the data is decrypted and then loaded into memory in plaintext. After TDE is enabled, data is not leaked even if backup files are disclosed. The backup files are encrypted and cannot be used to restore data to your computer. If you want to restore data to your computer, you must first [decrypt data](#).

9.10.4. SQL audit

You can use the SQL audit feature to audit SQL executions and check the details. SQL audit does not affect instance performance.

Context

 **Note** You cannot view the logs that are generated before you enable SQL audit.

You can view the incremental data of your ApsaraDB RDS for MySQL instance in SQL audit logs or binlogs. However, these two methods differ in the following aspects:

- SQL audit logs are similar to audit logs in MySQL and record all DML and DDL operations by using network protocol analysis. SQL audit does not parse the actual parameter values. Therefore, a small amount of information may be lost if a large number of SQL statements are executed to query data. The incremental data obtained using this method may be inaccurate.
- Binlogs record all add, delete, and modify operations and the incremental data used for data restoration. Binlogs are temporarily stored in your ApsaraDB for RDS instance after they are generated. The system transfers full binlog files to OSS on a regular basis. OSS then stores the files for seven days. However, a binlog file cannot be transferred if data is being written to it. Such binlog files will fail to be uploaded to OSS after you click **Upload Binlogs** on the **Backup and Restoration** page. Binlogs are not generated in real time, but you can obtain accurate incremental data from them.

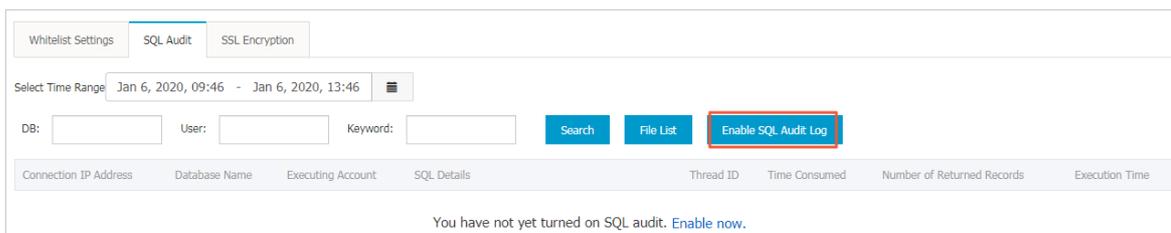
Precautions

- SQL audit is disabled by default. SQL audit does not affect instance performance.
- SQL audit logs are retained for 30 days.

- Log files exported from SQL audit are retained for two days. The system clears files that are retained for more than two days.

Enable SQL audit

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab.



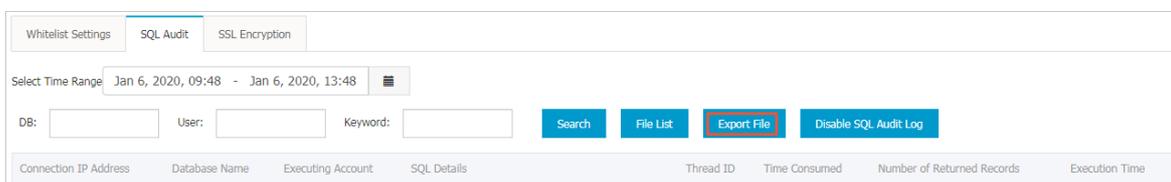
6. Click **Enable SQL Audit**.
7. In the message that appears, click **Confirm**.
After SQL audit is enabled, you can query SQL information based on conditions such as the time range, database, user, and keyword.

Disable SQL audit

Note If SQL audit is disabled, all SQL audit logs are deleted. We recommend that you export and store audit logs locally before you disable SQL audit.

You can disable SQL audit to avoid charges when you do not need it. To disable SQL audit, perform the following operations:

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab.



6. Click **Export File** to export and store the SQL audit content to your computer.
7. After the file is exported, click **Disable SQL Audit**.
8. In the message that appears, click **Confirm**.

9.11. Service availability

9.11.1. Automatically or manually switch over services between primary and secondary instances

This topic describes how to automatically or manually switch over services between primary and secondary instances. After the switchover, the original primary instance becomes a secondary instance.

Context

- **Automatic switchover:** the default switchover mode. If the primary instance experiences a fault, your RDS services are automatically switched over to the secondary instance.

 **Note** You can click **Switch Primary/Secondary Instance** on the **Service Availability** page of an ApsaraDB RDS for MySQL instance with standard SSDs or enhanced SSDs to disable automatic switchover. This ensures you to troubleshoot errors of the primary instance.

- **Manual switchover:** allows you to manually switch over services between primary and secondary instances.

 **Note** Data is synchronized between the primary and secondary instances in real time. You can only connect to the primary instance. The secondary instance serves only as a backup and does not allow external access.

Precautions

- Services may be disconnected during a switchover. Make sure that your applications are configured with automatic reconnection policies to avoid service disruptions.
- If the primary instance is attached with read-only instances, data on the read-only RDS instances shows a latency of a few minutes after a switchover. This is because it takes time to re-establish replication connections and synchronize incremental data.

Manually switch over services between primary and secondary instances

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. Click **Switch Primary/Secondary Instance** on the right side of the page.

 **Note** Services may be disconnected once or twice during the switchover. Make sure that your applications are configured with automatic reconnection policies to avoid service disruptions.

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

FAQ

Q: Can I connect to secondary instances?

A: No, you cannot connect to secondary instances. You can only connect to primary instances. Secondary instances only serve as a backup and do not allow external access.

9.11.2. Change the data replication mode

You can set the data replication mode between primary and secondary ApsaraDB for RDS instances to improve database availability.

Prerequisites

The RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 5.6
- MySQL 5.7 on RDS High-availability Edition (with local SSDs)

Data replication mode

- Semi-synchronous

After an update that is initiated by your application is completed on the primary instance, the log is synchronized to all of the secondary instances. The transaction that is used to perform the update is considered committed after the secondary instances receive the log. Your database system does not need to wait for the log to be replayed.

If the secondary instances are unavailable or a network exception occurs between the primary and secondary instances, semi-synchronous replication will degrade to the Asynchronous mode.

- Asynchronous

When your application initiates a request to add, delete, or modify data, the primary instance responds to your application immediately after it completes the operation. At the same time, the primary instance starts to asynchronously replicate data to its secondary instances. During asynchronous data replication, the unavailability of secondary instances does not affect the operations on the primary instance. Data remains consistent even if the primary instance is unavailable.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. Click **Change Data Replication Mode** on the right side of the page.
6. In the dialog box that appears, select a data replication mode and click **OK**.

FAQ

Q: Which data replication mode is recommended?

A: You can select a data replication mode based on your business requirements. If you require quick responses, we recommend that you select the asynchronous mode. In other scenarios, you can select the semi-synchronous mode.

9.12. Database backup and restoration

9.12.1. Automatic backup

Automatic backup supports full physical backups. ApsaraDB for RDS automatically backs up data based on pre-configured policies. This topic describes how to configure a policy for automatic backup.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. Click the **Backup Settings** tab.
6. Click **Edit**.

Note To ensure data security, the system compares the new backup cycle and time with the original settings, and selects the most recent time point to back up the data. Therefore, the next backup may still be performed based on the original backup cycle and time. For example, if the backup time is set to 19:00-20:00 every Wednesday and you modify the time to 19:00-20:00 every Thursday before 19:00 this Wednesday, the system will still back up data during 19:00-20:00 this Wednesday.

The screenshot shows a 'Backup Settings' dialog box with the following configuration:

- Data Retention Period:** 7 Days
- Backup Cycle:** Tuesday, Thursday, Saturday (checked)
- Backup Time:** 15:00-16:00
- Log Backup:** Enable (selected)
- Log Retention Period:** 7 Days

7. Configure the following parameters.

Parameter	Description
Data Retention Period	The number of days for which data backup files are retained. Valid values: 7 to 730. Default value: 7.
Backup Cycle	The backup cycle. You can select one or multiple days within a week.
Backup Time	Any period of time within a day. Unit: hours. We recommend that you back up data during off-peak hours.
Log Backup	Specifies whether to enable log backup. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p>Notice If you disable log backup, all the log backup files are deleted, and you cannot restore data to a saved point in time.</p> </div>
Log Retention Period	The number of days for which log backup files are retained. Valid values: 7 to 730. Default value: 7.

8. After you configure the preceding parameters, click **OK**.

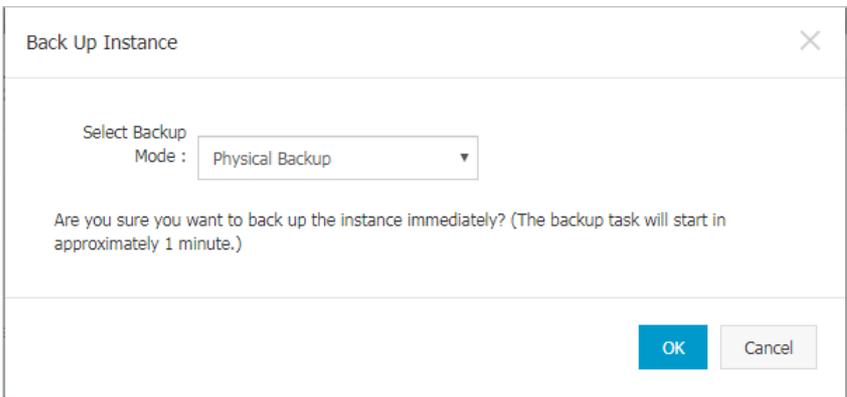
9.12.2. Manual backup

Manual backup supports both full physical backups and full logical backups. This topic describes how to manually back up ApsaraDB for RDS data.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Back Up Instance** in the upper-right corner.



5. Set the backup mode and backup policy, and click **OK**.

Note Two backup methods are available:

- Physical backup: directly backs up all files in all databases.
- Logical backup: extracts data from the databases through SQL and backs up the data in the text format. If you select logical backup, you must select a backup policy:
 - Instance Backup: backs up the entire instance.
 - Single-Database Backup: backs up one of the databases in the instance.

9.12.3. Download data and log backup files

This topic describes how to download unencrypted data and log backup files in the ApsaraDB for RDS console to archive the files and restore data to an on-premises database.

Limits

Database engine	Download of data backup files	Download of binary log files
MySQL	<ul style="list-style-type: none"> • MySQL 5.6 and MySQL 5.7 with local SSDs (in the RDS High-availability or Enterprise Editions) support the download of full physical or logical data backup files. • MySQL 5.7 with standard SSDs or enhanced SSDs does not support snapshot download. 	All MySQL versions support the download of log files.

Procedure

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. Click the **Data Backup** or **Log Backup** tab.
 - To download data backup files, click the **Data Backup** tab.
 - To download log backup files, click the **Log Backup** tab.

6. Select a time range to which you want to restore the instance.
7. Find the data backup or log file you want to download, and click **Download** in the **Actions** column.

Note

- If the **Download** button is unavailable, see the **Limits** section in this topic.
- If you want to use a data backup file to restore data, select the backup file that is the closest to the time for restoration.
- If you want to use a log file to restore data to an on-premises database, note the following items:
 - The Instance ID on the Log Backup tab is the same as the Instance No. on the Data Backup tab.
 - The start time of the file is later than the start time of the specified time range. It must also be earlier than the point in time to which you want to restore data.

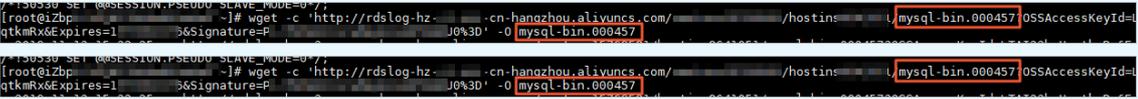
8. In the download message that appears, click **Download**.

Download method	Description
Download	Use a browser to download the backup file.
Copy Internal Endpoint	Copy the internal endpoint to download the file. If your ECS and RDS instances reside within the same region, you can log on to the ECS instance and use the internal endpoint to download the file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint to download the file. If you want to use other tools to download the file, use the public endpoint.

Note If you use a Linux operating system, you can run the following command to download the file:

```
wget -c '<Public endpoint of the backup file, which is the download URL>' -O <File name>
```

- The **-c** option enables resumable download.
- The **-O** option saves the downloaded file by using a specified name. We recommend that you use the file name contained in the download URL.
- If the URL contains more than one parameter, enclose the download URL in a pair of single quotation marks (').



9.12.4. Upload binlogs

Context

This topic describes how to upload binary log files to OSS.

Procedure

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration** to go to the **Backup and Restoration** page.
5. In the upper-right corner of the page, click **Upload Binlogs**.
6. In the message that appears, click **Confirm**.

9.12.5. Restore data to a new instance (formerly known as cloning an instance)

A cloned instance is a new instance with the same content as the primary instance, including data and settings. This feature allows you to restore data of the primary instance or create multiple instances that are the same as the primary instance.

Prerequisites

The following requirements must be met:

- The primary instance is in the running state.
- The primary instance does not have an ongoing migration task.
- Data backup and log backup are enabled.
- The primary instance has at least one completed backup set before you clone the instance by backup set.

Context

You can specify a backup set or any point in time within the backup retention period to clone an instance.

Note

- A cloned instance copies only the content of the primary instance, but not the content of read-only instances. The copied content includes database information, account information, and instance settings such as whitelist settings, backup settings, parameter settings, and alert threshold settings.
- The database engine of a cloned instance must be the same as that of the primary instance. Other settings can be different, such as the instance edition, zone, network type, instance type, and storage space. If you want to clone an instance to restore the data of a primary instance, we recommend that you select an instance type that has higher specifications and more storage space than those of the primary instance to speed up the data restoration process.
- The account type of a cloned instance must be the same as that of the primary instance. The account password of the cloned instance can be changed.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the backup list, find the target backup and click **Restore** in the **Actions** column.
6. In the dialog box that appears, select **Restore Database** and click **OK**.
7. On the **Restore RDS Instance** page, configure the following parameters.

Category	Parameter	Description
Region	Region	The region where the ApsaraDB for RDS instance resides.

Category	Parameter	Description
Database Restoration	Restore Mode	The data restore mode of the primary instance. Valid values: <ul style="list-style-type: none"> By Time By Backup Set
	Time	The point in time to which you want to restore the database. <p>Note When Restore Mode is set to By Time, you must specify this parameter.</p>
	Backup Set	The backup set for restoration. <p>Note When Restore Mode is set to By Backup Set, you must specify this parameter.</p>
Specifications	Instance Name	The name of the cloned instance.
	Database Engine	The engine of the database, which cannot be modified.
	Engine Version	The version of the database engine, which cannot be modified.
	Edition	The edition of the database. The actual values are displayed in the console.
	Storage Type	The storage type of the database. The actual values are displayed in the console.
	Instance Type	The type of the cloned instance. <p>Note We recommend that you select an instance type and storage space that are higher than those of the primary instance to speed up the data restoration process.</p>
	Storage	The storage space of the instance, including the space for data, system files, binary log files, and transaction files. The available storage capacity is displayed in the console. <p>Note RDS instances with local SSDs in the dedicated instance family occupy exclusive resources. The storage capacities are based on instance types.</p>

Category	Parameter	Description
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize route tables, IP address ranges, and gateways within a VPC. We recommend that you select VPC for improved security.
	VPC	<p>Select a VPC.</p> <p> Note When Network Type is set to VPC, you must specify this parameter.</p>
	VSwitch	<p>Select a VSwitch.</p> <p> Note When Network Type is set to VPC, you must specify this parameter.</p>

8. After you configure the preceding parameters, click **Submit**.

9.13. CloudDBA

9.13.1. Introduction to CloudDBA

CloudDBA is a cloud service for database self-detection, self-repair, self-optimization, self-maintenance, and self-security ensuring based on machine learning and expert experience. CloudDBA helps you ensure stable, secure, and efficient databases without worrying about the management complexity and services failures caused by manual operations.

Features

In ApsaraDB RDS for MySQL, CloudDBA provides the following features:

- **Diagnostics**
You can use this feature to diagnose your instance and visualize diagnostic results.
- **Instance sessions**
You can view sessions, collect session statistics, analyze SQL statements, and optimize the execution of SQL statements.
- **Real-time monitoring**
You can view the real-time information of your instance, such as the queries per second (QPS), transactions per second (TPS), number of connections, and network traffic.
- **Storage analysis**
You can view the space utilization, trends, exceptions, tablespaces, and data spaces.
- **Deadlock analysis**
You can view and analyze the last deadlock in a database.
- **Dashboard**

You can view and compare performance trends, customize monitoring dashboards, check exceptions, and view instance topologies.

- **Slow query logs**

You can view the trends and statistics of slow queries.

- **Diagnostic reports**

You can use this feature to generate diagnostics reports or view automatically generated reports about instance health, alerts, and slow query logs.

9.13.2. Diagnostics

In ApsaraDB RDS for MySQL, CloudDBA provides the diagnostics feature. This feature diagnoses your ApsaraDB RDS for MySQL instance and visualizes the results.

Open the Diagnostics page

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > One-click diagnostics**.
5. Click the **Diagnostics** tab.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Diagnostics*.

9.13.3. Instance sessions

In ApsaraDB RDS for MySQL, CloudDBA provides the instance sessions feature. This feature allows you to view and manage sessions of an instance.

Open the Instance Sessions page

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > One-click diagnostics**.
5. Click the **Instance Sessions** tab.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Instance sessions*.

9.13.4. Real-time monitoring

In ApsaraDB RDS for MySQL, CloudDBA provides the real-time monitoring feature. This feature allows you to view the real-time performance of your ApsaraDB RDS for MySQL instance.

Open the Real-time Monitoring page

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, choose **CloudDBA > One-click diagnostics**.
5. Click the **Real-time Monitoring** tab.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Real-time monitoring*.

9.13.5. Storage analysis

In ApsaraDB RDS for MySQL, CloudDBA provides the storage analysis feature. This feature allows you to check and resolve storage exceptions in a timely manner to ensure database stability.

Context

You can use the storage analysis feature of CloudDBA to view the disk space usage of your RDS instance and the number of remaining days when disk space is available. It also provides information about the space usage, fragmentation, and exception diagnostic results of a table.

Open the Storage Analysis page

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > One-click diagnostics**.
5. Click the **Storage Analysis** tab.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Storage analysis*.

9.13.6. Deadlock analysis

In ApsaraDB RDS for MySQL, CloudDBA provides the deadlock analysis feature. This feature allows you to view and analyze the last deadlock in a database.

Open the Deadlock Analysis page

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > One-click diagnostics**.
5. Click the **Deadlock Analysis** tab.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Deadlock analysis*.

9.13.7. Dashboard

In ApsaraDB RDS for MySQL, CloudDBA provides the dashboard feature. This feature allows you to view performance trends in specific ranges, compare performance trends, and customize charts to view performance trends.

Open the Dashboard page

1. [Log on to the ApsaraDB for RDS console](#).

2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Dashboard**.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Dashboard*.

9.13.8. Slow query logs

In ApsaraDB RDS for MySQL, CloudDBA provides the slow query logs feature. This feature allows you to view the trends and execution details of slow queries and obtain optimization suggestions for your ApsaraDB RDS for MySQL instance.

Open the Slow Query Logs page

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Slow Query Logs**.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Slow query logs*.

9.13.9. Diagnostic reports

In ApsaraDB RDS for MySQL, CloudDBA provides the diagnostic reports feature. This feature allows you to create and view diagnostic reports.

Open the Report page

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, choose **CloudDBA > Report**.

 **Note** Use the following path to navigate the guide: *Database Autonomy Service > User Guide > Diagnostic reports*.

9.14. Logs

All ApsaraDB for RDS instances support log management. You can query details about the error logs and slow query logs of an ApsaraDB for RDS instance through the ApsaraDB for RDS console. The logs help you locate faults.

Procedure

1. **Log on to the ApsaraDB for RDS console.**
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

- In the left-side navigation pane, click **Logs**.
- On the **Logs** page that appears, click the **Error Logs**, **Slow Query Logs**, **Slow Query Log Summary**, or **Primary/Secondary Switching Logs** tab, select a time range, and click **Search**.

Log type	Description
Error Logs	Records database running errors that occurred within the last month.
Slow Log Details	Records SQL statements within the last month that took longer than one second to execute. Duplicated SQL statements are removed. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p> Note Slow query logs in the ApsaraDB for RDS console are updated once every minute. However, you can query real-time slow query logs from the <code>mysql.slow_log</code> table.</p> </div>
Slow Query Log Summary	Records and analyzes SQL statements within the last month that took longer than one second to execute. Analysis reports of slow query logs are provided.
Primary/Secondary Switching Logs	Records the primary/secondary instance switching logs. This feature is suitable for ApsaraDB RDS for MySQL High-availability Edition instances.

9.15. Use mysqldump to migrate MySQL data

This topic describes how to use `mysqldump` to migrate local data to RDS for MySQL.

Prerequisites

An ECS instance has been activated.

Context

`mysqldump` is easy to use but has long downtimes. The tool is suitable for scenarios where the amount of data is small or long downtimes are allowed.

RDS for MySQL is fully compatible with the native database service. The procedure to migrate the original database to an RDS for MySQL instance is similar to that of migrating data from one MySQL server to another.

Before you migrate data, create a migration account in the local database, and grant the read and write permissions on the database to the migration account.

Procedure

- Run the following command to create a migration account in the local database: `CREATE USER 'username'@'host' IDENTIFIED BY 'password';`

Parameter description:

- `username`: specifies the name of the account to be created.
- `host`: specifies the database host to which the account logs on. As a local user, you can use `localhost` to log on to the database. To enable the account to log on to any host, you can use wildcard `%`.
- `password`: specifies the password that is used to log on to the account.

The following example creates an account named William with password Changme123, which is allowed to log on to the local database from any host.

```
CREATE USER 'William'@'%' IDENTIFIED BY 'Changme123';
```

- Run the following command to authorize the migration account of the local database: `GRANT SELECT ON database.tablename TO 'username'@'host' WITH GRANT OPTION; GRANT REPLICATION SLAVE ON database.tablename TO 'username'@'host' WITH GRANT OPTION; GRANT REPLICATION SLAVE ON database.tablename TO 'us`

```
username'@'host' WITH GRANT OPTION;
```

Parameter description:

- **privileges:** specifies the operation permissions of the account, such as SELECT, INSERT, and UPDATE. To grant all permissions to the account, use ALL.
- **dbname:** specifies the database name. To grant all database permissions to the account, use wildcard *.
- **tablename:** specifies the table name. To grant all table permissions to the account, use wildcard *.
- **username:** specifies the name of the account to be granted permissions.
- **host:** specifies the host, from which the account is authorized to log on to the database. As a local user, you can use localhost to log on to the database. To log on from any host, you can use wildcard %.
- **WITH GRANT OPTION:** specifies an optional parameter that enables the account to use the GRANT command.

In the following command, the account named William is granted all database and table permissions, and allowed to log on to the local database from any host:

```
GRANT ALL ON *. * TO 'William'@'%';
```

3. Use the data export tool of mysqldump to export data from the database as data files.

 **Notice** Do not update data during the data export. This step exports data only. It does not export stored procedures, triggers, or functions.

```
mysqldump -h localhost -u userName -p --opt --default-character-set=utf8 --hex-blob dbName --skip-triggers > /tmp/dbName.sql
```

Parameter description:

- **localhost:** specifies the IP address of the local database server.
- **userName:** specifies the migration account of the local database.
- **dbName:** specifies the name of the database to be migrated.
- **/tmp/dbName.sql:** specifies the name of the backup file.

4. Use mysqldump to export stored procedures, triggers, and functions.

 **Notice** Skip this step if no stored procedures, triggers, or functions are used in the database. When you export stored procedures, triggers, or functions, you must remove the definer to be compatible with RDS.

```
mysqldump -h localhost -u userName -p --opt --default-character-set=utf8 --hex-blob dbName -R | sed -e 's/DEFINER[ ]*=[ ]*[^\]*\*/\*/' > /tmp/triggerProcedure.sql
```

Parameter description:

- **localhost:** specifies the IP address of the local database server.
- **userName:** specifies the migration account of the local database.
- **dbName:** specifies the name of the database to be migrated.
- **/tmp/triggerProcedure.sql:** specifies the name of the backup file.

5. Upload the data files and stored procedure files to ECS. The example in this topic describes how to upload files to the following path:

```
/tmp/dbName.sql
```

```
/tmp/triggerProcedure.sql
```

6. Log on to ECS and import the data files and stored procedure files to the target RDS for MySQL instance.

```
mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/dbName.sql
```

```
mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/triggerProcedure.sql
```

Parameter description:

- `intranet4example.mysql.rds.aliyuncs.com`: specifies the IP address that is used to connect to the RDS for MySQL instance. In this example, an intranet IP address is used.
- `userName`: specifies the migration account of the RDS for MySQL database.
- `dbName`: specifies the name of the database to be imported.
- `/tmp/dbName.sql`: specifies the name of the data file to be imported.
- `/tmp/triggerProcedure.sql`: specifies the name of the stored procedure file to be imported.

10. ApsaraDB RDS for SQL Server

10.1. What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage of Alibaba Cloud, ApsaraDB for RDS allows you to easily perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS for SQL Server provides strong support for a variety of enterprise applications under the high-availability architecture, is capable of restoring data to any point in time.

ApsaraDB RDS for SQL Server provides basic features such as whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases.

10.2. Log on to the ApsaraDB for RDS console

This topic describes how to log on to the ApsaraDB for RDS console.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Database Services > ApsaraDB for RDS**.

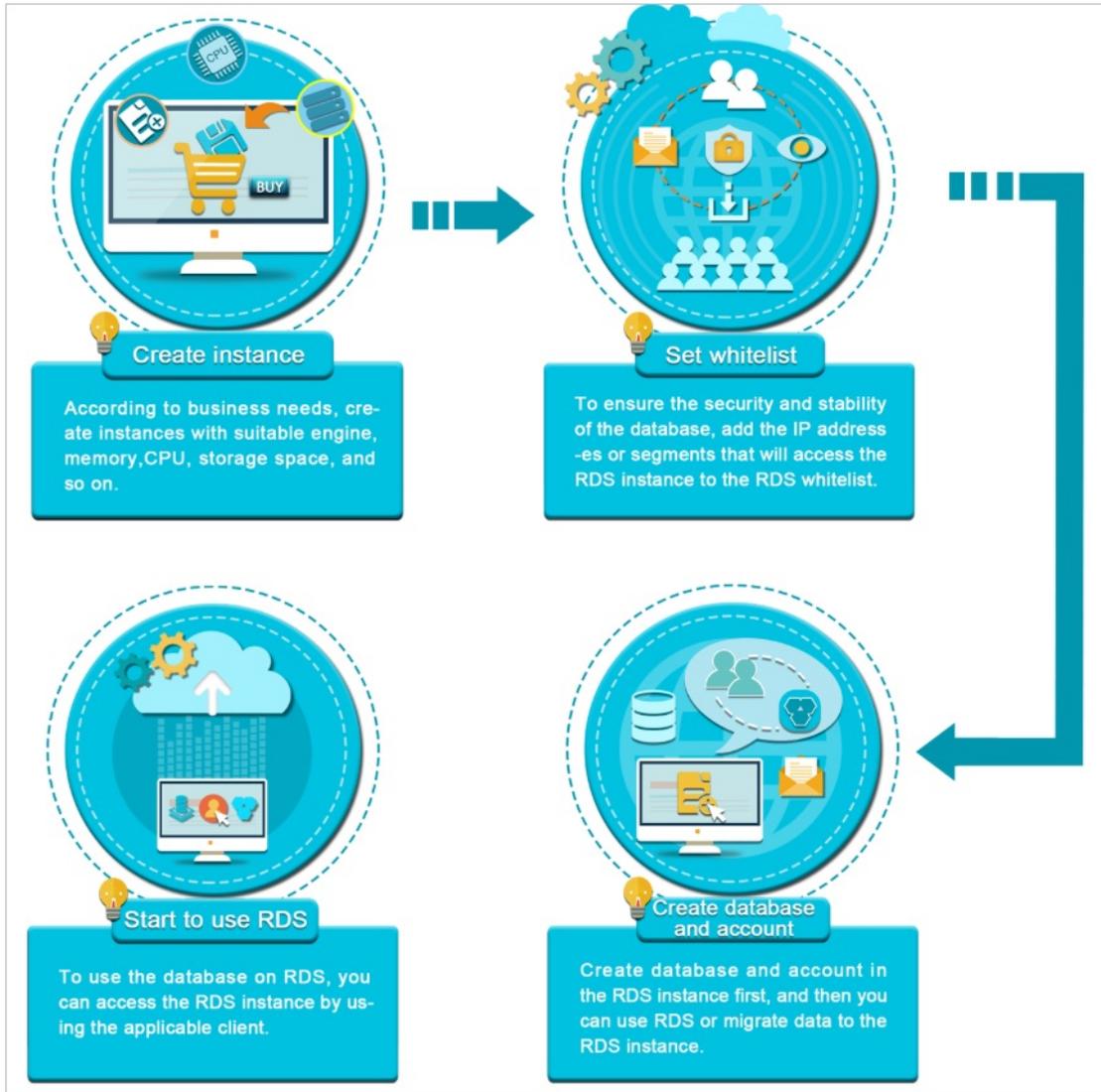
10.3. Quick Start

10.3.1. Procedure

ApsaraDB for RDS quick start covers the following topics: creating an RDS instance, configuring a whitelist, creating a database, creating an account, and connecting to the instance.

You must complete several operations after instance creation to make it ready for use, as shown in the following figure.

Quick start flowchart



10.3.2. Create an instance

This topic describes how to create an instance in the ApsaraDB for RDS console.

Prerequisites

Before you create an RDS instance, you must apply for an Apsara Stack account.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the Instances page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.

Section	Parameter	Description
Region	Region	The region where the instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The ID of the zone where the instance resides.
Specifications	Instance Name	<p>The name of the instance.</p> <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain special characters. Special characters include underscores (_), hyphens (-), and colons (:). ◦ The name cannot start with http:// or https://.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page.
	Engine Version	The version of the database engine.
	Edition	The edition of the instance. Select one from the drop-down list.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Product Introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files.
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note After you select a VPC, you must specify a VPC and a VSwitch.</p> </div>
	IP Whitelist	You can add IP addresses to allow them to connect to the ApsaraDB for RDS instance.
Access Mode	<p>The access mode of the instance, which is automatically set to Standard.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Standard: RDS uses SLB to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.</p> </div>	

4. After you configure the preceding parameters, click **Submit**.

10.3.3. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB for RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB for RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB for RDS instance.

Procedure

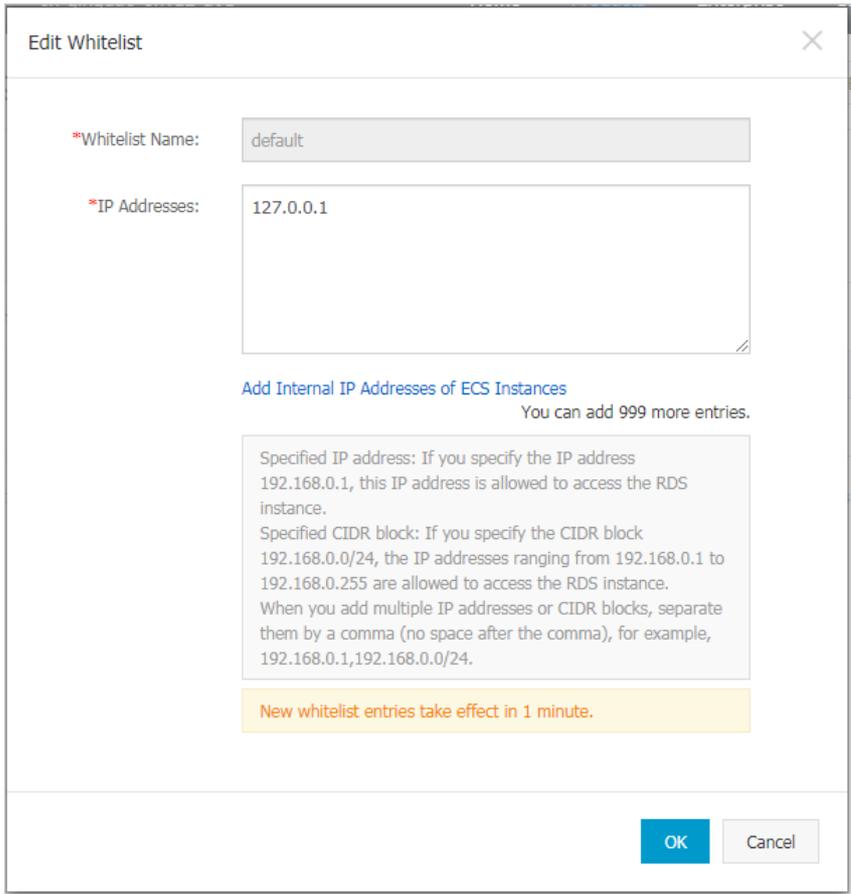
1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

Note

- If you want to connect an ECS instance to an ApsaraDB for RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
- You can click **Create Whitelist** to create a new whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**.
 - If you enter the CIDR block 10.10.10.0/24 in the **IP Addresses** field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - After you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances under your Apsara Stack account are displayed. You can select the required IP addresses and add them to the whitelist.

 **Note** If you add a new IP address or CIDR block to the default whitelist, the default address 127.0.0.1 is automatically deleted.



10.3.4. Connect to an instance

This topic describes how to use Data Management (DMS) to connect to an ApsaraDB for RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the RDS Database Logon page.
5. On the logon page, check the endpoint and port number. If the information is correct, enter the database account and password.

Note For more information about how to create an account, refer to the following topic:

- [Create an account for an ApsaraDB for RDS instance running SQL Server 2016 or 2012](#)

6. Click **Login**.

Note

- If you want the web browser to remember the password, select **Remember Password** before you click **Login**.
- If the system prompts you to add the CIDR block of the DMS server to an IP address whitelist of your RDS instance, click **Configure Whitelist**. For more information about how to manually configure a whitelist, see [Configure a whitelist](#).

10.3.5. Create an account

This topic describes how to create an account.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the right side of the page, click **Create Account**.
6. Enter the information of the account to be created.

Parameter	Description
Database Account	Enter the name of the account. The name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter and end with a letter or digit.
Account Type	<ul style="list-style-type: none"> ◦ Privileged Account: You can select the Privileged Account option only if this is the first time that you create an account for your RDS instance. Each RDS instance can have only one privileged account. The privileged account of an RDS instance cannot be deleted. ◦ Standard Account: You can select the Standard Account option only after a privileged account is created for your RDS instance. Each RDS instance can have more than one standard account. You must manually grant the permissions on databases to each standard account.
Authorized Databases	<p>Select the authorized databases of the account when you have selected the Standard Account type. If no databases are created, you can leave this parameter empty.</p> <p>You can follow these steps to grant the permissions on more than one database to the account:</p> <ol style="list-style-type: none"> i. In the Unauthorized Databases section, select the databases on which you want to grant permissions to the account. ii. Click the > icon to add the selected databases to the Authorized Databases section. iii. In the Authorized Databases section, specify the permissions that the account will gain on each authorized database. The permissions are Read/Write, Read-only, or Owner. You can also click Set All to Read/Write, Set All to Read-only, or Set All to Owner to set the permissions of the account on all authorized databases. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ▪ The account is authorized to create tables, delete tables, and modify schemas in a database only when it has the Owner permission on the database. ▪ The account has permissions on all databases and does not require authorization if you have selected the Privileged Account type. </div>
Password	<p>Enter the password of the account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the account again.
Description	Enter a description that helps identify the account. The description can be up to 256 characters in length.

7. Click **Create**.

10.3.6. Create a database

This topic describes how to create a database in an ApsaraDB RDS for SQL Server instance in the RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.

3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. In the upper-right corner of the page, click **Create Database**.
6. Set the parameters for the database that you want to create.

Parameter	Description
Database Name	Enter the name of the database. The name must be 2 to 64 characters in length. It can contain lowercase letters, digits, underscores (_), and hyphens (-). It must start with a letter and end with a letter or digit.
Supported Character Sets	Select the character set that is supported by the database. You can also select all and then select a character set from the drop-down list that appears.
Description	Enter a description of the database to facilitate subsequent management. The description can be up to 256 characters in length.

7. Click **Create**.

10.4. Instances

10.4.1. Create an instance

This topic describes how to create an instance in the ApsaraDB for RDS console.

Prerequisites

Before you create an RDS instance, you must apply for an Apsara Stack account.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The ID of the zone where the instance resides.
	Instance Name	The name of the instance. <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain special characters. Special characters include underscores (_), hyphens (-), and colons (:). ◦ The name cannot start with http:// or https://.

Section	Parameter	Description
Specifications	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page.
	Engine Version	The version of the database engine.
	Edition	The edition of the instance. Select one from the drop-down list.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Product Introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files.
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ Classic Network: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ VPC: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. <p> Note After you select a VPC, you must specify a VPC and a VSwitch.</p>
	IP Whitelist	You can add IP addresses to allow them to connect to the ApsaraDB for RDS instance.
Access Mode	Access Mode	<p>The access mode of the instance, which is automatically set to Standard.</p> <p> Note Standard: RDS uses SLB to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception.</p>

4. After you configure the preceding parameters, click **Submit**.

10.4.2. View basic information of an instance

This topic describes how to view the details of an ApsaraDB for RDS instance, such as basic information, internal network connection information, status, and configurations.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. Use one of the following methods to go to the **Basic Information** page of an instance:
 - On the **Instances** page, find the target instance and click the instance ID. The **Basic Information** page appears.
 - On the **Instances** page, find the target instance and click **Manage** in the corresponding **Actions** column. The **Basic Information** page appears.

10.4.3. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS for MySQL instance. This applies if the number of connections exceeds the specified threshold or if an instance has any performance issues.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Restart Instance** in the upper-right corner.

 **Note** A restart will disconnect the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

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

10.4.4. Change the specifications of an instance

This topic describes how to change specifications such as the instance type and storage space if they do not meet the requirements of your application. When the specification changes take effect, a 30-second network interruption may occur. Business operations that involve databases, accounts, and networks are interrupted. We recommend that you change the specifications during off-peak hours or make sure that your applications are configured with automatic reconnection policies.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Change Specifications**.
5. On the **Change Specifications** page, specify **Instance Type** and **Storage**.
6. After you configure the preceding parameters, click **Submit**.

10.4.5. Set the maintenance window of an instance

This topic describes how to set the maintenance window of an ApsaraDB RDS for SQL Server instance. The backend system performs maintenance on the RDS instance during the maintenance window. This ensures the stability of the RDS instance. The default maintenance window is from 02:00 (UTC+8) to 06:00 (UTC+8). We recommend that you set the maintenance window to off-peak hours of your business to avoid impacts on your business.

Context

- To ensure the stability of the maintenance process, the instance will enter the **Maintaining Instance** state before the maintenance time. While the instance is in this state, access to data in the database and query operations such as performance monitoring are not affected. However, modification operations such as upgrade, downgrade, and restart (except for account and database management and IP address whitelist configuration) will be unavailable.
- During the maintenance window, one or two network interruptions may occur. Make sure that your

applications are configured with automatic reconnection policies.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

 **Note** The maintenance window is displayed in UTC+8.

10.4.6. Switch over services between a primary RDS instance and its secondary instance

ApsaraDB for RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance.

Prerequisites

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

Context

Each ApsaraDB RDS for SQL Server instance has a secondary instance. Data between the primary and secondary instances is synchronized in real time. You can access only the primary instance. The secondary instance is a backup instance and cannot be accessed. If the primary instance cannot be accessed, your business automatically switches over to the secondary instance. After the switchover, the primary instance becomes the secondary instance.

Notice

- During a switchover, a network interruption may occur. Make sure that your applications are configured with automatic reconnection policies.
- During a switchover, a one-minute data quality protection mechanism is enabled for data synchronization. If the primary and secondary database states are incorrect or if the latency for data synchronization exceeds one minute due to SQL Server errors, the HA system does not automatically perform the primary/secondary switchover. You must determine whether to perform the switchover.
- If an instance is intermittently unavailable due to excessive mirroring event waits, the switchover is not performed. The instance will be automatically available again.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the dialog box that appears, click **OK**.

Result

After the switchover is complete, the original primary instance becomes the secondary instance for the next primary/secondary switchover.

10.4.7. Release an instance

This topic describes how to manually release an ApsaraDB RDS for SQL Server instance.

Context

- Only instances in the running state can be manually released.
- After an instance is released, the instance data is immediately deleted. We recommend that you back up your data before you release an instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. In the Actions column corresponding to the instance you want to release, click **More** and select **Release Instance** from the drop-down list.
3. In the **Release Instance** message, click **Confirm**.

10.5. Accounts

10.5.2. Create an account

This topic describes how to create an account.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the right side of the page, click **Create Account**.
6. Enter the information of the account to be created.

Parameter	Description
Database Account	Enter the name of the account. The name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter and end with a letter or digit.
Account Type	<ul style="list-style-type: none">◦ Privileged Account: You can select the Privileged Account option only if this is the first time that you create an account for your RDS instance. Each RDS instance can have only one privileged account. The privileged account of an RDS instance cannot be deleted.◦ Standard Account: You can select the Standard Account option only after a privileged account is created for your RDS instance. Each RDS instance can have more than one standard account. You must manually grant the permissions on databases to each standard account.

Parameter	Description
Authorized Databases	<p>Select the authorized databases of the account when you have selected the Standard Account type. If no databases are created, you can leave this parameter empty.</p> <p>You can follow these steps to grant the permissions on more than one database to the account:</p> <ol style="list-style-type: none"> i. In the Unauthorized Databases section, select the databases on which you want to grant permissions to the account. ii. Click the > icon to add the selected databases to the Authorized Databases section. iii. In the Authorized Databases section, specify the permissions that the account will gain on each authorized database. The permissions are Read/Write, Read-only, or Owner. You can also click Set All to Read/Write, Set All to Read-only, or Set All to Owner to set the permissions of the account on all authorized databases. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ▪ The account is authorized to create tables, delete tables, and modify schemas in a database only when it has the Owner permission on the database. ▪ The account has permissions on all databases and does not require authorization if you have selected the Privileged Account type. </div>
Password	<p>Enter the password of the account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the password of the account again.
Description	Enter a description that helps identify the account. The description can be up to 256 characters in length.

7. Click **Create**.

10.5.3. Reset the password

You can use the ApsaraDB for RDS console to reset the password of your database account.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Find the target account and click **Reset Password** in the **Actions** column.
6. In the dialog box that appears, enter and confirm the new password, and then click **OK**.

-  **Note** The password must meet the following requirements:
- The password must be 8 to 32 characters in length.
 - The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
 - Special characters include ! @ # \$ % ^ & * () _ + - =

10.6. Databases

10.6.1. Create a database

This topic describes how to create a database for an ApsaraDB RDS for SQL Server instance.

Terms

- **Instance:** a virtualized database server, on which you can create and manage more than one database.
- **Database:** a set of data that is stored together in a certain way and can be shared by multiple users. A database provides the smallest redundancy. It is a data warehouse independent of applications.
- **Character set:** a collection of letters and special characters and their encoding rules used in a database.

Prerequisites

An ApsaraDB RDS for SQL Server instance is created. For more information, see [Create an instance](#).

Procedure

For more information, see [Create a database](#).

What to do next

For more information, see [Connect to an instance](#).

10.6.2. Delete a database

This topic describes how to delete a database from an instance in the console or by using SQL statements based on the instance type.

Use the console to delete a database

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. Find the target database and click **Delete** in the **Actions** column.
6. In the message that appears, click **Confirm**.

Execute an SQL statement to delete a database

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the RDS Database Logon page.
5. On the **Data Management (DMS)** logon page that appears, check the endpoint and port number displayed

on the page. If the information is correct, enter the database account and password in the fields.

-  **Note** For more information about how to create an account, see
- [Create an account for an ApsaraDB for RDS instance running SQL Server 2016 or 2012.](#)

6. Click **Login**.

-  **Note**
- If you want the browser to remember the password, select **Remember your password** and click **Login**.
 - If the system prompts you to add the Classless Inter-Domain Routing (CIDR) block of the DMS server to an IP address whitelist of the RDS instance, click **Configure Whitelist**. For more information about how to manually configure the whitelist, see [Configure a whitelist](#).

7. In the top navigation bar, choose **SQL Operations > SQL Window**.

8. Enter the following statement and click **execute** to delete the database.

```
drop database <database name>;
```

-  **Note** For high-availability instances of ApsaraDB RDS for SQL Server 2012 and later, you can also use the following stored procedure. This stored procedure deletes the specified database, removes the associated image, and kills connections to the database.

```
EXEC sp_rds_drop_database 'database name'
```

10.6.3. Change the character set collation and the time zone of system databases

This topic describes how to change the character set collation and the time zone of system databases. System databases include master, msdb, tempdb, and model.

Prerequisites

- The instance runs ApsaraDB RDS for SQL Server 2012 on Standard Edition.
- No database other than system databases exists in the instance.

-  **Note** If you have just deleted databases from the instance, the deletion task may be pending in the secondary instance. Before you change the character set collation and the time zone, make sure that neither the primary instance nor the secondary instance contains databases.

Precautions

- The default character set collation is Chinese_PRC_CI_AS.
- The default time zone is China Standard Time.
- You can view the available character set collations and time zones in the console.
- The instance is in the unavailable state during the change process. It takes about one minute to change the time zone, and 2 to 10 minutes to change the character set collation.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Databases**.
5. On the Databases page, click **Change Character Set Collation and Time Zone**.

 **Note** If you fail to find this button on the page, make sure that requirements in **Prerequisites** are met.

6. In the dialog box that appears, select a **Time Zone**, **Character Set Collation**, or both of them, and click **OK**.

UTC offsets of time zones

Time zone	UTC offset	Description
Afghanistan Standard Time	(UTC+04:30)	Kabul
Alaskan Standard Time	(UTC-09:00)	Alaska
Arabian Standard Time	(UTC+04:00)	Abu Dhabi, Muscat
Atlantic Standard Time	(UTC-04:00)	Atlantic Time (Canada)
AUS Central Standard Time	(UTC+09:30)	Darwin
AUS Eastern Standard Time	(UTC+10:00)	Canberra, Melbourne, Sydney
Belarus Standard Time	(UTC+03:00)	Minsk
Canada Central Standard Time	(UTC-06:00)	Saskatchewan
Cape Verde Standard Time	(UTC-01:00)	Cabo Verde Is.
Gen. Australia Standard Time	(UTC+09:30)	Adelaide
Central America Standard Time	(UTC-06:00)	Central America
Central Asia Standard Time	(UTC+06:00)	Astana
Central Brazilian Standard Time	(UTC-04:00)	Cuiaba
Central Europe Standard Time	(UTC+01:00)	Belgrade, Bratislava, Budapest, Ljubljana, Prague
Central European Standard Time	(UTC+01:00)	Sarajevo, Skopje, Warsaw, Zagreb
Central Pacific Standard Time	(UTC+11:00)	Solomon Islands, New Caledonia
Central Standard Time	(UTC-06:00)	Central Time (US and Canada)
Central Standard Time (Mexico)	(UTC-06:00)	Guadalajara, Mexico City, Monterrey
China Standard Time	(UTC+08:00)	Beijing, Chongqing, Hong Kong, Urumqi
E. Africa Standard Time	(UTC+03:00)	Nairobi

Time zone	UTC offset	Description
E. Australia Standard Time	(UTC+10:00)	Brisbane
E. Europe Standard Time	(UTC+02:00)	Chisinau
E. South America Standard Time	(UTC-03:00)	Brasilia
Eastern Standard Time	(UTC-05:00)	Eastern Time (US and Canada)
Georgian Standard Time	(UTC+04:00)	Tbilisi
GMT Standard Time	(UTC)	Dublin, Edinburgh, Lisbon, London
Greenland Standard Time	(UTC-03:00)	Greenland
Greenwich Standard Time	(UTC)	Monrovia, Reykjavik
GTB Standard Time	(UTC+02:00)	Athens, Bucharest
Hawaiian Standard Time	(UTC-10:00)	Hawaii
India Standard Time	(UTC+05:30)	Chennai, Kolkata, Mumbai, New Delhi
Jordan Standard Time	(UTC+02:00)	Amman
Korea Standard Time	(UTC+09:00)	Seoul
Middle East Standard Time	(UTC+02:00)	Beirut
Mountain Standard Time	(UTC-07:00)	Mountain Time (US and Canada)
Mountain Standard Time (Mexico)	(UTC-07:00)	Chihuahua, La Paz, Mazatlan
US Mountain Standard Time	(UTC-07:00)	Arizona
New Zealand Standard Time	(UTC+12:00)	Auckland, Wellington
Newfoundland Standard Time	(UTC-03:30)	Newfoundland
Pacific SA Standard Time	(UTC-03:00)	Santiago
Pacific Standard Time	(UTC-08:00)	Pacific Time (US and Canada)
Pacific Standard Time (Mexico)	(UTC-08:00)	Baja California
Russian Standard Time	(UTC+03:00)	Moscow, St. Petersburg, Volgograd
SA Pacific Standard Time	(UTC-05:00)	Bogota, Lima, Quito, Rio Branco
SE Asia Standard Time	(UTC+07:00)	Bangkok, Hanoi, Jakarta
China Standard Time	(UTC+08:00)	Kuala Lumpur, Singapore
Tokyo Standard Time	(UTC+09:00)	Osaka, Sapporo, Tokyo
US Eastern Standard Time	(UTC-05:00)	Indiana (East)
UTC	UTC	Coordinated Universal Time
UTC-02	(UTC-02:00)	Coordinated Universal Time-02

Time zone	UTC offset	Description
UTC-08	(UTC-08:00)	Coordinated Universal Time-08
UTC-09	(UTC-09:00)	Coordinated Universal Time-09
UTC-11	(UTC-11:00)	Coordinated Universal Time-11
UTC+12	(UTC+12:00)	Coordinated Universal Time+12
W. Australia Standard Time	(UTC+08:00)	Perth
W. Central Africa Standard Time	(UTC+01:00)	West Central Africa
W. Europe Standard Time	(UTC+01:00)	Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

10.7. Database connection

10.7.1. Change the endpoint and port number of an instance

This topic describes how to view and change the endpoint and port number of an instance.

View the endpoint and port number

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
You can view the **Internal Endpoint** and **Internal Port** of the instance in the **Database Connection** section.

Change the endpoint and port number

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Change Endpoint**.
6. In the dialog box that appears, set **Connection Type**, **Endpoint**, and **Port**, and click **OK**.

Note

- The prefix of the endpoint must be 8 to 64 characters in length and can contain letters, digits, and hyphens (-). It must start with a lowercase letter.
- The port number must be in the range of 1000 to 65534.

10.7.2. Connect to an instance

This topic describes how to use Data Management (DMS) to connect to an ApsaraDB for RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the RDS Database Logon page.
5. On the logon page, check the endpoint and port number. If the information is correct, enter the database account and password.

 **Note** For more information about how to create an account, refer to the following topic:

- [Create an account for an ApsaraDB for RDS instance running SQL Server 2016 or 2012](#)

6. Click **Login**.

 **Note**

- If you want the web browser to remember the password, select **Remember Password** before you click **Login**.
- If the system prompts you to add the CIDR block of the DMS server to an IP address whitelist of your RDS instance, click **Configure Whitelist**. For more information about how to manually configure a whitelist, see [Configure a whitelist](#).

10.8. Monitoring and alerting

10.8.1. Set a monitoring frequency

The ApsaraDB for RDS console provides a variety of performance metrics for which you can set a monitoring frequency.

Context

ApsaraDB for RDS provides the following monitoring frequencies:

- Every 60 seconds
- Every 300 seconds

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Resource Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select the monitoring frequency you want.
7. Click **OK**.

10.8.2. View resource and engine monitoring data

The ApsaraDB for RDS console provides a variety of performance metrics to monitor the status of your instances.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring and Alerts** page, select **Resource Monitoring** or **Engine Monitoring**, and select a time range to view the corresponding monitoring data. The following table describes the metrics.

Monitoring type	Metric	Description
Resource Monitoring	Disk Space (unit: MB)	The disk usage of the RDS instance, including: <ul style="list-style-type: none"> ◦ Instance Size ◦ Data Usage ◦ Log Size ◦ Temporary File Size ◦ Other System File Size
	IOPS (unit: times/second)	The number of input/output operations per second (IOPS) for the RDS instance.
	Total Connections	The total number of current connections of the instance.
	MSSQL Instance CPU Utilization (percentage in the operating system: %)	The CPU utilization of the RDS instance. This includes the CPU utilization for the operating system. Unit: %.
	SQLServer Average Input/Output Traffic	The inbound and outbound traffic of the instance per second. Unit: KB.
Engine Monitoring	Average Transaction Frequency	The number of transactions processed per second.
	Average QPS	The number of SQL statements executed per second.
	Buffer Hit Ratio (%)	The read hit ratio of the buffer pool.
	Page Write Frequency at Check Point	The number of checkpoints written to pages per second.
	Login Frequency	The number of logons to the RDS instance per second.
	Average Frequency of Whole Table Scans	The number of full table scans per second.
	SQL Compilations per Second	The number of SQL statements compiled per second.
	Lock Timeout Times	The number of lock timeouts on the RDS instance per second.
	Deadlock Frequency	The number of deadlocks on the RDS instance per second.
	Lock Wait Frequency	The number of lock waits on the RDS instance per second.

10.9. Data security

10.9.1. Configure a whitelist

To ensure database security and reliability, you must modify the whitelist of an ApsaraDB for RDS instance before you enable the instance. You must add the IP addresses or CIDR blocks that are used for database access to the whitelist.

Context

The whitelist improves the access security of your ApsaraDB for RDS instance. We recommend that you maintain the whitelist on a regular basis. The whitelist configuration process does not affect the normal operations of the ApsaraDB for RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

Note

- If you want to connect an ECS instance to an ApsaraDB for RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
- You can click **Create Whitelist** to create a new whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**.
 - If you enter the CIDR block 10.10.10.0/24 in the **IP Addresses** field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - After you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances under your Apsara Stack account are displayed. You can select the required IP addresses and add them to the whitelist.

 **Note** If you add a new IP address or CIDR block to the default whitelist, the default address 127.0.0.1 is automatically deleted.

Whitelist Name: default

IP Addresses: 127.0.0.1

[Add Internal IP Addresses of ECS Instances](#)
You can add 999 more entries.

Specified IP address: If you specify the IP address 192.168.0.1, this IP address is allowed to access the RDS instance.
Specified CIDR block: If you specify the CIDR block 192.168.0.0/24, the IP addresses ranging from 192.168.0.1 to 192.168.0.255 are allowed to access the RDS instance.
When you add multiple IP addresses or CIDR blocks, separate them by a comma (no space after the comma), for example, 192.168.0.1,192.168.0.0/24.

New whitelist entries take effect in 1 minute.

OK Cancel

10.9.2. Configure SSL encryption for an instance

This topic describes how to enhance endpoint security. You can enable Secure Sockets Layer (SSL) encryption and install SSL certificates issued by certificate authorities (CAs) on the required application services. SSL is used at the transport layer to encrypt network connections and enhance the security and integrity of communication data. SSL also increases the response time.

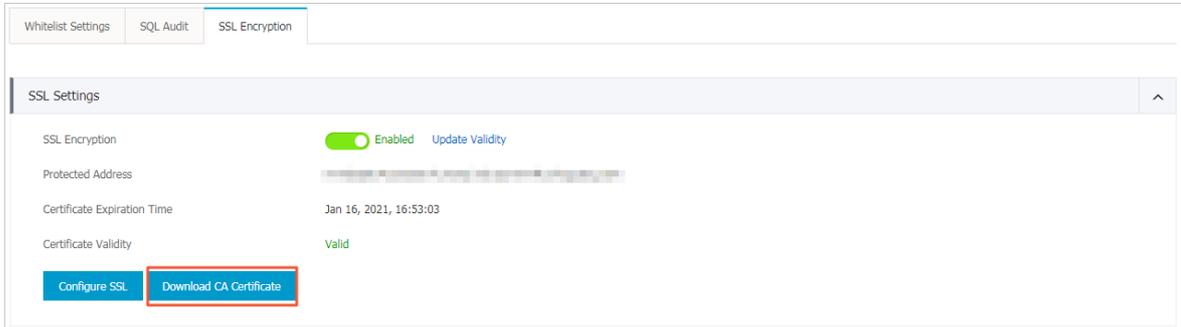
Precautions

- An SSL CA certificate is valid for one year. You must update the validity period of the SSL CA certificate in your application or client within one year. Otherwise, your application or client that uses an encrypted network connections cannot connect the RDS instance.
- SSL encryption may cause a significant increase in CPU utilization. We recommend that you enable SSL encryption only when you want to encrypt connections from the Internet. In most cases, connections that use an internal endpoint do not require SSL encryption.
- Read/write splitting endpoints do not support SSL encryption.
- If you enable SSL encryption, you cannot disable it. Proceed with caution.

Enable SSL encryption

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. In the **SSL Settings** section, turn on **SSL Encryption**.
7. In the **Configure SSL** dialog box, select the endpoint for which you want to enable SSL encryption and click **OK**.
8. Click **Download CA Certificate** to download the SSL CA certificate files in a compressed package.



The downloaded package contains three files:

- **P7B file:** used to import CA certificates to the Windows operating system.
- **PEM file:** used to import CA certificates to other operating systems or applications.
- **JKS file:** the Java truststore file. The password is `apsaradb`. It is used to import the CA certificate chain to Java programs.

Note When the JKS file is used in Java, you must modify the default JDK security configuration in JDK 7 and JDK 8. Open the `/jre/lib/security/java.security` file on the host where your application resides, and modify the following configurations:

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
jdk.certpath.dis
```

```
abledAlgorithms=MD2, RSA keySize < 1024
```

If you do not modify the JDK security configuration, the following error is reported. Similar errors are also caused by the Java security configuration.

```
javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
```

Configure an SSL CA certificate

After you enable SSL encryption, configure the SSL CA certificate on your application or client before they can connect to the RDS instance. This section describes how to configure an SSL CA certificate. MySQL Workbench and Navicat are used in the example. For more information, see the instructions for the other applications or clients.

Configure a certificate on MySQL Workbench

1. Start MySQL Workbench.
2. Choose **Database > Manage Connections**.
3. Enable **Use SSL** and import the SSL CA certificate files.

Configure a certificate on Navicat

1. Start Navicat.

2. Right-click the target database and select **Edit Connection**.
3. Click the **SSL** tab. Select the path of the PEM-formatted CA certificate.
4. Click **OK**.

Note If the connection is being used error is reported, the previous session is still connected. Restart Navicat.

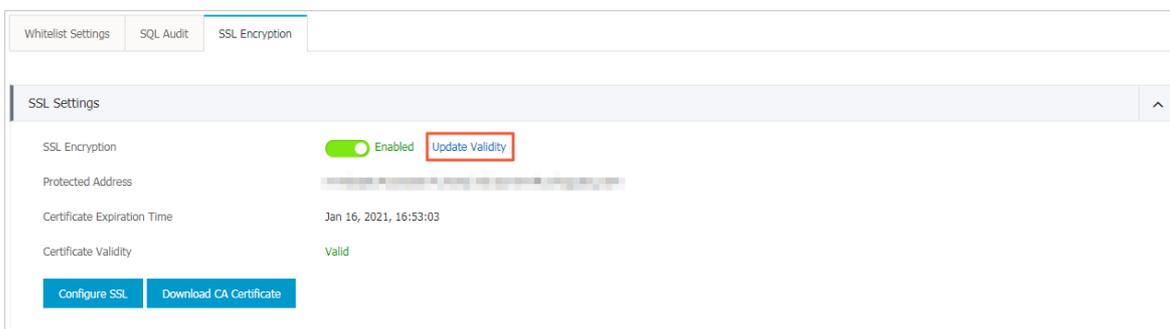
5. Double-click the target database to test whether the database is connected.

Update the validity period of an SSL CA certificate

Note

- **Update Validity** causes the RDS instance to restart. Proceed with caution.
- After you update the validity period, you must download and configure the SSL CA certificate again.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SSL Encryption** tab.
6. Click **Update Validity**.



10.9.3. Configure TDE

Transparent data encryption (TDE) encrypts and decrypts data files in real time. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the sizes of data files. When you use TDE, you do not need to modify the application that uses the ApsaraDB for RDS instance.

Precautions

- Instance-level TDE can be enabled but cannot be disabled. Database-level TDE can be enabled or disabled.
- The keys used for data encryption are generated and managed by Key Management Service (KMS). ApsaraDB for RDS does not provide the keys or certificates used for data encryption. If you want to restore data to your computer after TDE is enabled, you must decrypt the data on your RDS instance. For more information, see [Decrypt data](#).
- TDE increases CPU utilization.

Prerequisites

- Your RDS instance runs SQL Server EE.
- You have logged on to the ApsaraDB for RDS console by using your Apsara Stack account.

- KMS is activated. If KMS is not activated, you can activate it as prompted when you enable TDE.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **TDE** tab.
6. Click the switch next to **Disabled** to turn on TDE.
7. In the dialog box that appears, click **Confirm**.

 **Note** If you have not enabled KMS, you will be prompted to do so when you enable TDE. After you enable KMS, you can click **Not activated** to enable TDE.

8. Click **Configure TDE**. In the Database TDE Settings dialog box, select the databases you want to encrypt from the **Unselected Databases** list, click the  icon to add them to the **Selected Databases** list, and click **OK**.

Decrypt data

If you want to decrypt a database that is encrypted by TDE, you only need to remove the database from the **Selected Databases** list in the **Database TDE Settings** dialog box.

10.11. Database backup and restoration

10.11.1. Configure an automatic backup policy

Automatic backup supports full physical backups. ApsaraDB for RDS automatically backs up data based on pre-configured policies. This topic describes how to configure a policy for automatic backup.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Setting** tab.
6. Click **Edit**.
7. In the dialog box that appears, configure the automatic backup policy.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Default value: 7. Unit: days.
Backup Cycle	The backup cycle. You can select multiple days within a week. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> ? Note To ensure data security, we recommend that you back up your RDS instance at least twice a week. </div>
Backup Time	The hour at which you want to create a backup.
Backup Frequency	<ul style="list-style-type: none"> ◦ Same as Data Backup ◦ Every 30 Minutes <p>The total size of log backup files remains the same no matter which backup frequency you select.</p>

8. Click **OK**.

10.11.2. Manually back up an instance

This topic describes how to manually back up an ApsaraDB for RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. On the **Basic Information** page, click **Back Up Instance** in the upper-right corner.
5. In the **Back Up Instance** dialog box, select **Automatic Backup** or **Full Backup** from the **Select Backup Mode** drop-down list.

-  **Note** ApsaraDB for RDS supports the following backup methods:
- **Automatic Backup:** After you select **Automatic Backup**, the system immediately performs an incremental backup or full backup based on the instance.
 - **Full Backup:** After you select **Full Backup**, the system immediately performs a full backup.

6. Click **OK**.

Result

After the backup is complete, you can view the backup task on the **Data Backup** tab of the **Backup and Restoration** page.

10.11.3. Shrink transaction logs

ApsaraDB RDS for SQL Server allows you to shrink transaction logs to reduce the log file size.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Shrink Transaction Log**.

-  **Note** The shrinkage takes about 20 minutes to complete. ApsaraDB RDS for SQL Server shrinks transaction logs during each backup.

10.12. Migrate full backup data to ApsaraDB RDS for SQL Server

This topic describes how to migrate full backup files of an on-premises database from Object Storage Service (OSS) to ApsaraDB RDS for SQL Server.

Prerequisites

- Your ApsaraDB for RDS instance has sufficient storage space. If the space is insufficient, you must increase it before you migrate data to the instance.
- The destination database on your ApsaraDB for RDS instance has a different name from the on-premises database.
- A privileged account is created on your ApsaraDB for RDS instance. For more information, see [Create an account](#).
- An Object Storage Service (OSS) bucket is created in the region where your ApsaraDB for RDS instance is created. For more information, see [Create buckets in the OSS User Guide](#).
- The DBCC CHECKDB statement is executed, and the execution result indicates that no allocation or consistency errors occur.

 **Note** If no allocation or consistency errors occur, the following execution result is returned:

```
...  
CHECKDB found 0 allocation errors and 0 consistency errors in database 'xxx'.  
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

Precautions

- Full backup files cannot be migrated to an ApsaraDB for RDS instance of an earlier SQL Server version. For example, if the on-premises database runs SQL Server 2016 and your ApsaraDB for RDS instance runs SQL Server 2012, you cannot migrate full backup files of the on-premises database to your ApsaraDB for RDS instance.
- Differential or log backup files are not supported.
- The names of full backup files cannot contain special characters, such as `@` and vertical bars (`|`). If the file names contain special characters, the migration fails.
- After the service account of your ApsaraDB for RDS instance is granted the access permission on the OSS bucket, the system creates a role named **AliyunRDSImportRole** in RAM. Do not modify or delete this role. Otherwise, you cannot download full backup files when you migrate data to your ApsaraDB for RDS instance. In this case, you must re-authorize the service account of your ApsaraDB for RDS instance.
- Before the migration is complete, do not delete the backup files from the OSS bucket. Otherwise, the migration fails.
- The names of backup files can be suffixed only with bak, diff, tm, or log. If you do not use the script in this topic to generate a backup file, you must name the backup file by using one of the following suffixes:
 - bak: indicates a full backup file.

- diff: indicates a differential backup file.
- trn or log: indicates a log backup file.

Back up the on-premises database

 **Note** Before you perform a full backup, stop writing data to the on-premises database. The data written during the backup process is not backed up.

1. Download the [backup script](#). Double-click the backup script to open it by using the Microsoft SQL Server Management Studio (SSMS) client.
2. Configure the following parameters.

Parameter	Description
@backup_databases_list	The databases that you want to back up. Separate them with semicolons (;) or commas (,).
@backup_type	The backup type. Valid values: <ul style="list-style-type: none"> ○ FULL: full backup ○ DIFF: differential backup ○ LOG: log backup
@backup_folder	The directory in which you want to store the backup files on your computer. If the specified directory does not exist, the system creates a directory.
@is_run	Specifies whether to perform a backup. Valid values: <ul style="list-style-type: none"> ○ 1: performs a backup. ○ 0: performs no backup but a check.

3. Run the backup script.

Upload full backup files to the OSS bucket

After the on-premises database is backed up, you must upload full backup files to the OSS bucket. You can use one of the following methods:

- Use the OSS console

If the size of backup files is smaller than 5 GB, you can upload the files in the OSS console. For more information, see [Upload objects](#) in the *OSS User Guide*.

- Call an OSS API operation

You can call an OSS API operation to upload the full backup files in resumable mode. For more information, see [Multipart upload-relevant operations](#) in the *OSS Developer Guide*.

Create a migration task

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to

the **Basic Information** page.

4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Migrate OSS Backup Data to RDS**.
6. Click **Next** twice until the **Import Data** step appears.
7. Configure the following parameters.

Parameter	Description
Database Name	Enter the name of the destination database on your ApsaraDB for RDS instance.  Note The name of the database must meet the requirements of SQL Server.
OSS Bucket	Select the OSS bucket that stores the backup files.
OSS Subfolder Name	Enter the name of the OSS subfolder that stores the backup files.
OSS File	Click the search icon to search for backup files by using the prefix-based fuzzy match. The system displays the name, size, and update time of each backup file. Select the backup file that you want to migrate to your ApsaraDB for RDS instance.
Cloud Migration Method	One-time Full Backup File Migration: uploads full backup data to your ApsaraDB for RDS instance. Select this option if you want to migrate only a single full backup file.

8. Click **OK**.

Wait for the migration task to complete. You can click **Refresh** to view the latest status of the migration task. If the migration fails, fix the error based on the message displayed in the **Task Description** column. For more information, see [Common errors](#).

View the migration task

In the left-side navigation pane, click **Backup and Restoration**. Click the **Backup Data Upload History** tab. The system displays the migration tasks in the last week.

Common errors

Each record of a migration task contains a task description, which helps you identify the error cause and fix the error. The following list describes common errors:

- A database with the same name as the on-premises database exists on your ApsaraDB for RDS instance.
 - Error message: The database (xxx) is already exist on RDS, please backup and drop it, then try again.
 - Cause: The on-premises database is named the same as an existing database on your ApsaraDB for RDS instance. For data security purposes, ApsaraDB RDS for SQL Server does not allow such a database to be migrated.

- Solution: If you need to overwrite the database in your ApsaraDB for RDS instance with the on-premises database, you must back up the database, delete it from your ApsaraDB for RDS instance, and then migrate the on-premises database to your ApsaraDB for RDS instance.
- A differential backup file is used.
 - Error message: Backup set (xxx.bak) is a Database Differential backup, we only accept a FULL Backup.
 - Cause: The file that you uploaded is a differential backup file, but not a full backup file. The migration solution for full backup data supports only full backup files.
- A log backup file is used.
 - Error message: Backup set (xxx.trn) is a Transaction Log backup, we only accept a FULL Backup.
 - Cause: The file that you uploaded is a log backup file, but not a full backup file. The migration solution for full backup data supports only full backup files.
- The backup file fails the verification.
 - Error message: Failed to verify xxx.bak, backup file was corrupted or newer edition than RDS.
 - Cause: The backup file is damaged, or the on-premises database runs an SQL Server version later than your ApsaraDB for RDS instance. For example, if the on-premises database runs SQL Server 2016 and your ApsaraDB for RDS instance runs SQL Server 2012, the error message is returned.
 - Solution: If the backup file is damaged, perform a full backup on the on-premises database again. If the database engine version does not meet the requirements, select an ApsaraDB for RDS instance that runs the same version as or a later version than the on-premises database.
- DBCC CHECKDB fails to be executed.
 - Error message: DBCC checkdb failed.
 - Cause: Allocation or consistency errors occurred in the on-premises database.
 - Solution: Execute the following statement in the on-premises database.

 **Note** Data loss may occur when you use this statement to fix errors.

```
DBCC CHECKDB (DBName, REPAIR_ALLOW_DATA_LOSS) WITH NO_INFOMSGS, ALL_ERRORMSGS
```

- The remaining storage space of your ApsaraDB for RDS instance is insufficient. (1)
 - Error message: Not Enough Disk Space for restoring, space left (xxx MB) < needed (xxx MB).
 - Cause: The remaining storage space of your ApsaraDB for RDS instance does not meet the migration requirements.
 - Solution: Increase the storage space of your ApsaraDB for RDS instance.
- The remaining storage space of your ApsaraDB for RDS instance is insufficient. (2)
 - Error message: Not Enough Disk Space, space left xxx MB < bak file xxx MB.
 - Cause: The remaining storage space of your ApsaraDB for RDS instance is smaller than the size of the backup file.
 - Solution: Increase the storage space of your ApsaraDB for RDS instance.
- No privileged account exists.
 - Error message: Your RDS doesn't have any init account yet, please create one and grant permissions on RDS console to this migrated database (XXX).

- Cause: No privileged account is created on your ApsaraDB for RDS instance, and the database permissions are not granted to accounts. However, when this error message is returned, the backup file has been restored to your ApsaraDB for RDS instance, and the migration task is successful.
- Solution: Create a privileged account. For more information, see [Create an account](#).

11.PolarDB

11.1. What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB for RDS allows you to easily perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

PolarDB

PolarDB is a stable, secure, and scalable enterprise-grade relational database that provides one of the database engines that ApsaraDB for RDS runs. Based on PostgreSQL, the most advanced open source database in the world, PolarDB enhances performance, application solutions, and compatibility. It also provides the capability of directly running Oracle applications. You can run enterprise-grade applications on PolarDB to implement stable and cost-effective services.

11.2. Limits on PolarDB

Before you use PolarDB, you must understand its limits and take necessary precautions.

The following table describes the limits on PolarDB.

Operation	Limit
Database parameter modification	Not supported.
Root privilege of databases	Superuser permissions are not provided.
Database replication	<ul style="list-style-type: none"> The system automatically builds HA databases based on PolarDB streaming replication without user input. PolarDB standby nodes are hidden and cannot be accessed directly.
ApsaraDB for RDS instance restart	PolarDB instances must be restarted using the ApsaraDB for RDS console or API operations.

11.3. Log on to the ApsaraDB for RDS console

This topic describes how to log on to the ApsaraDB for RDS console.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

- In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
- Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Database Services > ApsaraDB for RDS**.

11.4. Quick Start

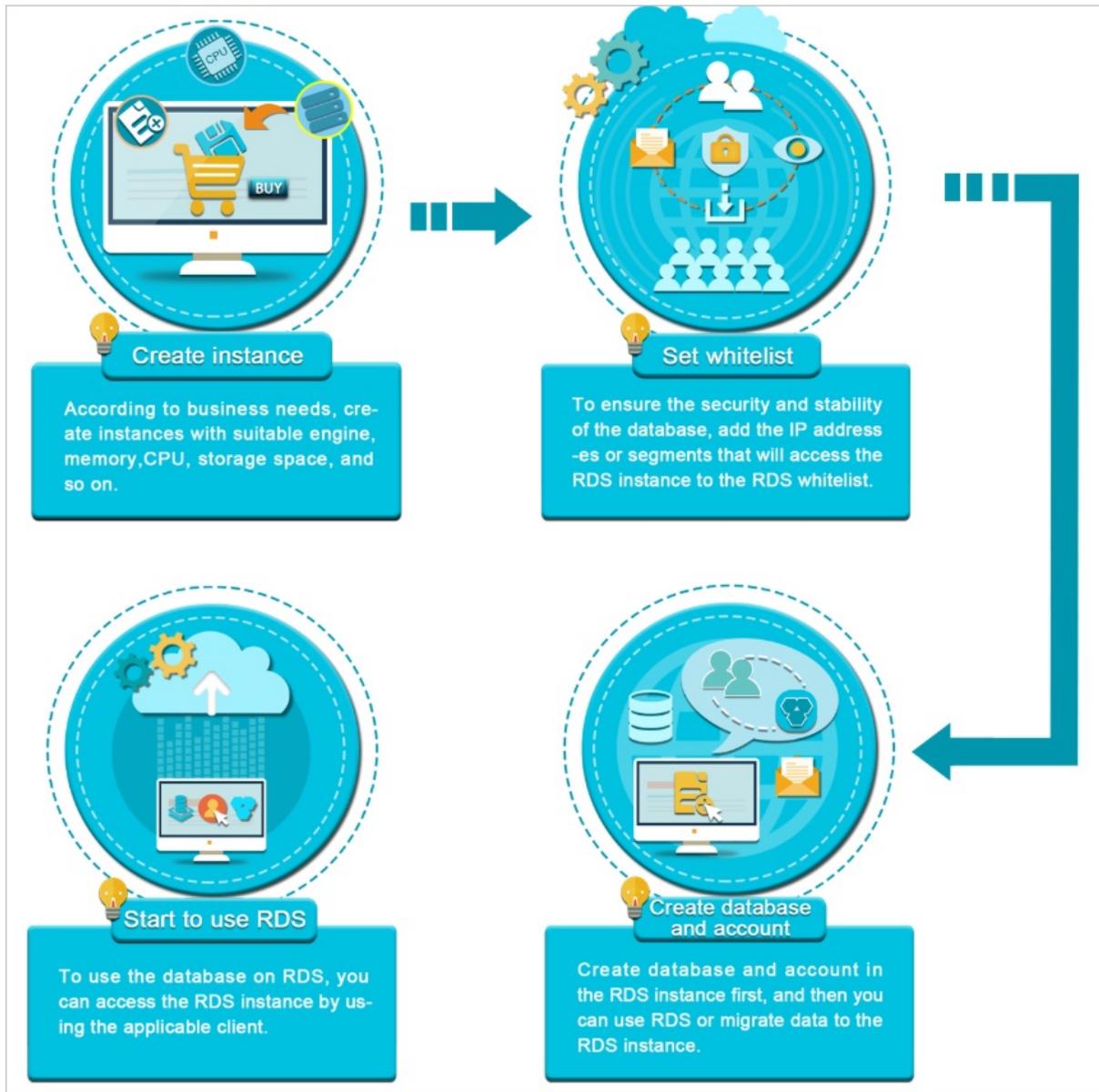
11.4.1. Procedure

This topic describes how to create a PolarDB instance, specify basic configurations, and connect to the instance. It allows you to familiarize yourself with the entire process about how to purchase and use a PolarDB instance.

Quick start flowchart

If you are using ApsaraDB for RDS for the first time, you can start with [Limits](#).

The following figure shows the operations that you must perform before you use a PolarDB instance.



11.4.2. Create an instance

This topic describes how to create a PolarDB instance in the console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides.
	Zone	The zone ID of the instance.

Region Section	Parameter	Description
Specifications	Instance Name	<p>The name of the instance.</p> <ul style="list-style-type: none"> The name must be 2 to 64 characters in length. The name must start with a letter. The name can contain digits and special characters. Special characters include underscores (_), hyphens (-), and colons (:). The name cannot start with http:// or https://.
	Database Engine	The database engine of the instance. Set the value to PolarDB.
	Engine Version	The version of the database engine. Set the value to 11.
	Edition	The edition of the instance. The drop-down list displays supported instance editions.
	Instance Type	The type of the instance. The memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Product introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the capacity to store data, system files, binary log files, and transaction files. The available storage capacity is displayed in the console.
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> Classic Network: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. VPC: A Virtual Private Cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. <p> Note After you select VPC, you must select a VPC and a VSwitch.</p>
	IP Whitelist	An IP address whitelist contains the IP addresses of devices that require access to your RDS instance. For more information, see Configure an IP whitelist .
Access Mode	Access Mode	<p>RDS instances support two access modes: Standard and Database Proxy.</p> <ul style="list-style-type: none"> Standard: RDS uses Server Load Balancer (SLB) to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception. Database Proxy: This mode prevents 90% of network interruptions and intercepts SQL injection attacks based on semantic analysis. However, it increases the response time by over 20%. <p> Note PolarDB supports the Standard mode.</p>

4. After you configure the preceding parameters, click **Submit**.

11.4.3. Configure an IP whitelist

This topic describes how to configure a whitelist for an ApsaraDB for RDS instance. Only devices that are listed in a whitelist can access your RDS instance.

Context

Whitelists make your RDS instance more secure and do not interrupt the operations of your RDS instance during configuration. We recommend that you update the IP address whitelists and security groups configured for your RDS instance on a regular basis.

You can use one of the following methods to configure a whitelist.

- **Configure a whitelist:** Add IP addresses to grant them access to the RDS instance.

 **Note** The default IP address whitelist contains only the IP address 127.0.0.1. This indicates that no devices are allowed to access the RDS instance.

- **Configure an ECS security group:** Add an ECS security group for the RDS instance to allow ECS instances in the group to access the RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**. On the page that appears, click the **Whitelist Settings** tab.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

 **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**. The following section describes the rules:
 - If you enter the CIDR block 10.10.10.0/24 in the **IP Addresses** field, all IP addresses in the 10.10.10.X format can access your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all created ECS instances within your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

11.4.4. Create a database and an account

Before you start to use ApsaraDB for RDS, you must create a database and an account for an ApsaraDB for RDS instance. This topic describes how to create a database and an account for a PolarDB instance.

Create an account

You can create initial and standard accounts for a PolarDB instance. The following section describes how to create an initial account.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

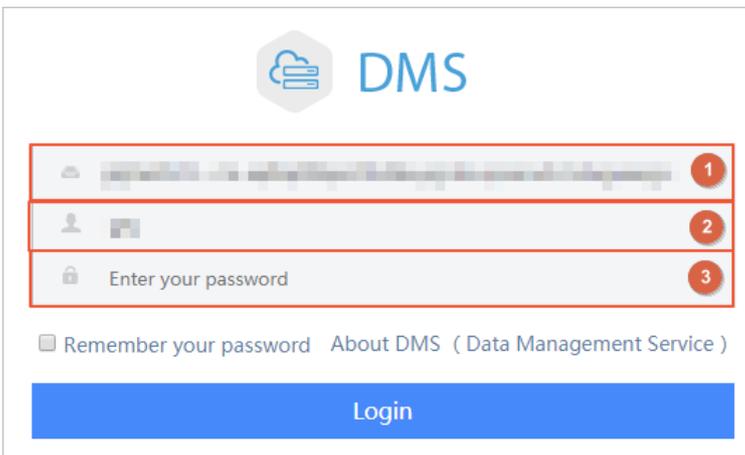
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Initial Account** and configure the following parameters.

Parameter	Description
Database Account	<ul style="list-style-type: none"> ◦ The name of the account must be 2 to 16 characters in length. ◦ The name of the account can contain lowercase letters, digits, and underscores (_). ◦ The name of the account must start with a letter and end with a letter or digit.
Password	<ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the same password again.

6. Click **Create**.

Create a database and a standard account

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS)** logon page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.



Parameter	Description
IP Address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the name of the account used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account used to access the database.

6. Click **Login**.

 **Note** If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP whitelist](#).

7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.

8. In the SQL window, execute the following statement to create a database:

```
CREATE DATABASE name
[ [ WITH ] [ OWNER [=] user_name ]
  [ TEMPLATE [=] template ]
  [ ENCODING [=] encoding ]
  [ LC_COLLATE [=] lc_collate ]
  [ LC_CTYPE [=] lc_ctype ]
  [ TABLESPACE [=] tablespace_name ]
  [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

9. Click **execute**.

10. In the SQL window, execute the following statement to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
  SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

11. Click **execute**.

11.4.5. Connect to a PolarDB instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB for RDS instance.

Context

You can log on to DMS from the ApsaraDB for RDS console and then connect to an RDS instance.

Data Management (DMS) is an integrated database for data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational databases and non-relational databases, such as MySQL, SQL Server, and PostgreSQL. It can also be used to manage Linux servers.

You can also use a database client to connect to an RDS instance. ApsaraDB for RDS is fully compatible with native database engines. You can connect to RDS instances in the similar manner as you would connect to an on-premises database. This topic describes how to use the pgAdmin 4 client to connect to an RDS instance.

Use DMS to connect to an RDS instance

For more information about how to connect an RDS instance through DMS, see [Log on to an ApsaraDB for RDS instance by using DMS](#).

Use the pgAdmin 4 client to connect to an RDS instance

1. Add the IP address that requires access to the RDS instance to a whitelist of the RDS instance. For more information about how to configure a whitelist, see [Configure an IP whitelist](#).
2. Start the pgAdmin 4 client.

 **Note** For more information about how to download the pgAdmin 4 client, visit [pgAdmin 4 \(Windows\)](#).

3. Right-click **Servers** and choose **Create > Server**.
4. On the **General** tab of the **Create - Server** dialog box that appears, enter the name of the server.
5. Click the **Connection** tab and enter the information of the destination instance.

Parameter	Description
Host name/address	The internal endpoint of the RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number .
Port	The internal port number that is used to connect to the RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number .
Username	The name of the initial account for the RDS instance. For more information about how to create an initial account, see Create a database and an account .
Password	The password of the initial account.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**.

 **Notice** The postgres database is the default system database of the RDS instance. Do not perform operations on this database.

11.5. Instances

11.5.1. Create an instance

This topic describes how to create a PolarDB instance in the console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides.
	Zone	The zone ID of the instance.
Specifications	Instance Name	The name of the instance. <ul style="list-style-type: none"> ◦ The name must be 2 to 64 characters in length. ◦ The name must start with a letter. ◦ The name can contain digits and special characters. Special characters include underscores (_), hyphens (-), and colons (:). ◦ The name cannot start with http:// or https://.
	Database Engine	The database engine of the instance. Set the value to PolarDB.
	Engine Version	The version of the database engine. Set the value to 11.
	Edition	The edition of the instance. The drop-down list displays supported instance editions.
	Instance Type	The type of the instance. The memory size determines the maximum number of connections and IOPS. The actual values are displayed in the console. For more information, see Product introduction > Instance type in the <i>ApsaraDB for RDS documentation</i> .
	Storage	The storage capacity of the instance, including the capacity to store data, system files, binary log files, and transaction files. The available storage capacity is displayed in the console.

Section	Parameter	Description
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A Virtual Private Cloud (VPC) helps you build an isolated network environment on Alibaba Cloud. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for improved security. <p> Note After you select VPC, you must select a VPC and a VSwitch.</p>
	IP Whitelist	<p>An IP address whitelist contains the IP addresses of devices that require access to your RDS instance. For more information, see Configure an IP whitelist.</p>
Access Mode	Access Mode	<p>RDS instances support two access modes: Standard and Database Proxy.</p> <ul style="list-style-type: none"> ◦ <i>Standard</i>: RDS uses Server Load Balancer (SLB) to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception. ◦ <i>Database Proxy</i>: This mode prevents 90% of network interruptions and intercepts SQL injection attacks based on semantic analysis. However, it increases the response time by over 20%. <p> Note PolarDB supports the Standard mode.</p>

4. After you configure the preceding parameters, click **Submit**.

11.5.2. Restart an instance

This topic describes how to manually restart an ApsaraDB for RDS instance. This applies if the number of connections exceeds the specified threshold or a performance issue occurs.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Restart Instance**.

 **Notice** When you restart an instance, a network interruption will occur. We recommend that you make appropriate arrangements for your workloads and make sure that your applications are configured with automatic reconnection policies.

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

11.5.3. Set the maintenance window

This topic describes how to set the maintenance window of an ApsaraDB for RDS instance. The backend system performs maintenance on the RDS instance during the maintenance window. This ensures the stability of the RDS instance. The default maintenance window is from 02:00 (UTC+8) to 06:00 (UTC+8). We recommend that you set the maintenance window to the off-peak hours of your business to avoid impacts on your business.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

Note

- The maintenance window is displayed in UTC+8.
- Before the maintenance starts, ApsaraDB for RDS sends text messages and emails to the contacts that are associated with your Alibaba Cloud account.
- To ensure the stability of the maintenance process, the instance will enter the **Maintaining Instance** state before the maintenance time. Access to data in the database and query operations such as performance monitoring are not affected while the instance is in this state. However, apart from account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart will be temporarily unavailable.
- During a maintenance window, one or two network interruptions may occur. Make sure that your applications are configured with automatic reconnection policies.

11.5.4. Configure primary/secondary switchover

ApsaraDB for RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance. This topic describes how to manually switch over services between a primary instance and its secondary instance.

Context

Data is synchronized in real time between the primary and secondary instances. You can access only the primary instance. The secondary instance serves only as a backup instance and does not allow external access. After the switchover, the original primary instance becomes the secondary instance.

 **Note** Network interruptions may occur during a switchover. Make sure that your applications are configured with automatic reconnection policies.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the **Switch Primary/Secondary Instance** message, click **OK**.

Note

- During the switchover, operations such as managing databases and accounts and changing network types cannot be performed. Therefore, we recommend that you select Switch Within Maintenance Window.
- For more information about how to set a maintenance window, see [Set the maintenance window](#).

11.5.5. Change the network type

This topic describes how to change the network type of an ApsaraDB for RDS instance between classic network and VPC.

Context

- **Classic network:** RDS instances in the classic network are not isolated. Unauthorized access to these instances can be blocked only by whitelists.
- **VPC:** Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you select the VPC network type because it is more secure than the classic network.

You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can connect your own data center to a VPC by using leased lines or VPNs. This allows you to build a virtual data center on the cloud.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the internal endpoint of the RDS instance remains unchanged, but the IP address bound to the internal endpoint changes.
- After the network type is changed, ECS instances in the same VPC as the RDS instance can no longer be able to connect to the RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- Changing the network type may result in a network interruption of 30 seconds. To avoid interruptions to your business, we recommend that you change the network type during off-peak hours or configure automatic reconnection policies for your applications.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to Classic Network**.
6. In the dialog box that appears, click **OK**.

Note After the network type is changed to classic network, only ECS instances in the classic network can access your RDS instance over an internal network. Configure the internal endpoint for these ECS instances.

7. Configure a whitelist of your RDS instance to allow access from the ECS instance over the internal network.

Note

- If the network isolation mode of the RDS instance is standard whitelist, add the internal IP addresses of the ECS instances to a whitelist of your RDS instance.
- If the network isolation mode of the RDS instance is **enhanced whitelist**, add the internal IP addresses of the ECS instances to a classic network whitelist. If no classic network whitelists are available, create a whitelist.

Change the network type from classic network to VPC

1. Log on to the **ApsaraDB for RDS console**.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Switch to VPC**.
6. In the **Switch to VPC** dialog box, select a VPC and VSwitch and specify whether to retain the endpoint used in the classic network.

Note

- Select a VPC. We recommend that you select the VPC where your ECS instances reside. Otherwise, the ECS instances cannot communicate with the RDS instance over the internal network.
- Select a VSwitch. If no VSwitches are available in the selected VPC, create one in the same zone where the RDS instance resides. Use the following path to navigate the guide: *Virtual Private Cloud > User Guide > Quick Start > Create a VSwitch*.
- Determine whether to select the **Reserve Original Classic Endpoint** option. The following table describes the details.

Operation	Description
Not selected	<p>The endpoint used in the classic network is replaced with an endpoint in the VPC.</p> <p>When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the RDS instance are interrupted.</p>
Selected	<p>The endpoint used in the classic network is retained, and a new endpoint to be used in the VPC is generated. In such cases, the RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the RDS instance over the internal network.</p> <p>When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the RDS instance will be available until the endpoint used in the classic network expires.</p> <p>To migrate your business to the VPC without interruption, you must add the new endpoint used in the VPC to access the ECS instances before the endpoint used in the classic network expires. Seven days before the endpoint used in the classic network expires, the system will send a text message to the phone number bound to your Alibaba Cloud account every day.</p> <p>For more information, see Hybrid access from both the classic network and VPCs.</p>

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the RDS instance over the internal network. If no VPC whitelists are available, create a

whitelist.



Note

- If you have retained the classic network endpoint, add the VPC endpoint to the ECS instances before the classic network endpoint expires.
- If you have not retained the classic network endpoint, connections between ECS instances in the classic network and the RDS instance over the internal network are interrupted. You must add the new endpoint to ECS instances in the VPC immediately after the network type is changed.

11.5.6. Release an instance

This topic describes how to manually release an ApsaraDB for RDS instance to meet your business needs.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. In the **Actions** column corresponding to the target instance, click **More** and select **Release Instance** from the drop-down list.



Note After an instance is released, the instance data is immediately deleted. We recommend that you back up the data and download the backup file before you release an instance. For more information, see [Back up data](#) and [Download backup files](#).

4. In the message that appears, click **Confirm**.

11.5.7. Change the specifications of an instance

This topic describes how to change the instance type and storage space of an ApsaraDB for RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section of the **Basic Information** page, click **Change Specifications**.



Notice While you change specifications of an instance, a network interruption of about 30 seconds may occur, and most of the operations related to databases, accounts, and network operations cannot be performed. We recommend that you make appropriate arrangements for you workloads before you change specifications.

5. Change specifications based on your business requirement and click **Submit**.

11.5.8. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query the parameter modification records in the console.

Modify parameters

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic**

Information page.

4. In the left-side navigation pane, click **Parameters**.
5. You can perform the following operations: Export the parameter settings of the RDS instance to your computer.

On the **Editable Parameters** tab, click **Export Parameters**. The parameter settings of the RDS instance are exported as a TXT file to your computer.

Modify and import the parameter settings.

- i. After you have modified parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.
- ii. Click **OK**.
- iii. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
- Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

Modify a single parameter.

- i. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the  icon in the **Actual Value** column.
- ii. Enter a new value based on the prompted value range.
- iii. Click **Confirm**.
- iv. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your applications are configured with automatic reconnection policies.
- Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the page that appears, click the **Edit History** tab.
6. Select a time range and then click **Search**.

11.6. Database connection

11.6.1. Connect to a PolarDB instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB for RDS instance.

Context

You can log on to DMS from the ApsaraDB for RDS console and then connect to an RDS instance.

Data Management (DMS) is an integrated database for data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational databases and non-relational databases, such as MySQL, SQL Server, and PostgreSQL. It can also be used to manage Linux servers.

You can also use a database client to connect to an RDS instance. ApsaraDB for RDS is fully compatible with native database engines. You can connect to RDS instances in the similar manner as you would connect to an on-premises database. This topic describes how to use the pgAdmin 4 client to connect to an RDS instance.

Use DMS to connect to an RDS instance

For more information about how to connect an RDS instance through DMS, see [Log on to an ApsaraDB for RDS instance by using DMS](#).

Use the pgAdmin 4 client to connect to an RDS instance

1. Add the IP address that requires access to the RDS instance to a whitelist of the RDS instance. For more information about how to configure a whitelist, see [Configure an IP whitelist](#).
2. Start the pgAdmin 4 client.

 **Note** For more information about how to download the pgAdmin 4 client, visit [pgAdmin 4 \(Windows\)](#).

3. Right-click **Servers** and choose **Create > Server**.
4. On the **General** tab of the **Create - Server** dialog box that appears, enter the name of the server.
5. Click the **Connection** tab and enter the information of the destination instance.

Parameter	Description
Host name/address	The internal endpoint of the RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number .
Port	The internal port number that is used to connect to the RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number .
Username	The name of the initial account for the RDS instance. For more information about how to create an initial account, see Create a database and an account .
Password	The password of the initial account.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**.

 **Notice** The postgres database is the default system database of the RDS instance. Do not perform operations on this database.

11.6.2. Hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB for RDS to change the network type of an instance from classic network to Virtual Private Network (VPC) without network interruptions.

Prerequisites

- The network type of the instance is classic network.
- Available VPCs and VSwitches exist in the zone where the instance resides.

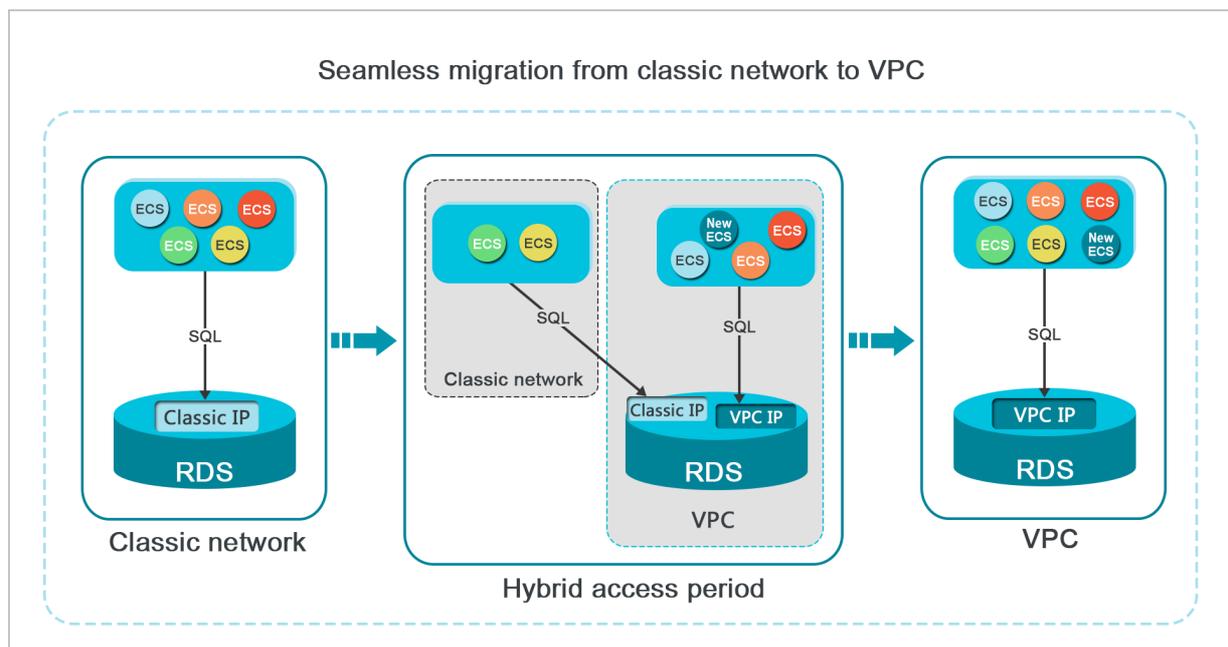
Context

In the past, when you changed the network type of an RDS instance from classic network to VPC, the internal endpoint of the RDS instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the RDS instance through the internal endpoint within this period. To smoothly change the network type, ApsaraDB for RDS provides the hybrid access solution.

Hybrid access refers to the ability of an RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimize the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database through the internal endpoint of VPCs, and the other applications can access the database through the original internal endpoint of the classic network. When all the applications access the database through the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- Changing to the classic network is not supported.
- Migrating the RDS instance to another zone is not supported.

Change the network type from classic network to VPC

For more information, see [Change the network type from classic network to VPC](#).

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be connected over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

Follow these steps to change the expiration time:

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the Instance Connection tab, click **Change Expiration Time**.
6. On the **Change Expiration Time** dialog box that appears, select an expiration time and click **OK**.

11.6.3. Log on to an ApsaraDB for RDS instance by using

DMS

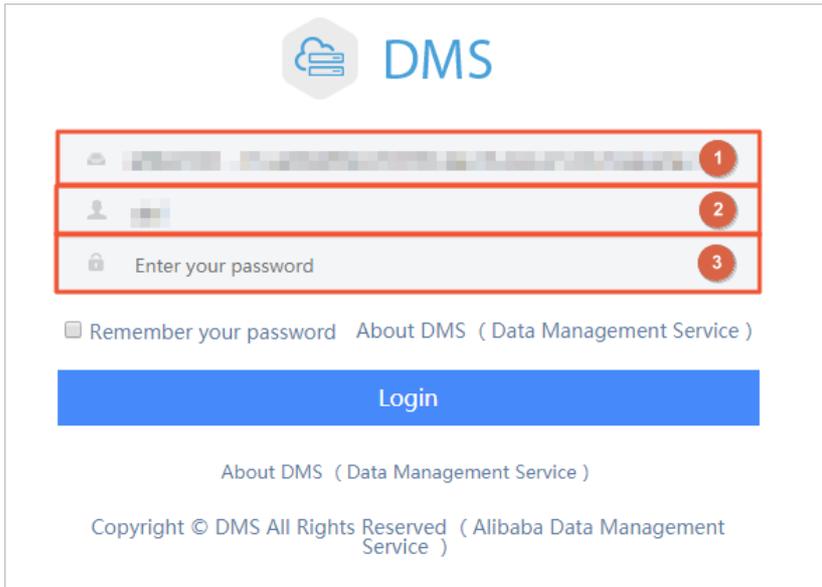
This topic describes how to log on to an ApsaraDB for RDS instance by using Data Management (DMS).

Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see [Configure an IP whitelist](#).

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the RDS Database Logon page.
5. On the logon page, set the following parameters:



- ①: The endpoint and port number that are used to connect to your RDS instance. The endpoint and port number are in the `<Internal endpoint>:<Internal port number>` format. Example: `rm-bpxxxxxx.rds.aliyuncs.com:3433`. For more information about how to view the internal endpoint and port number of an instance, see [View and modify the internal endpoint and port number](#).
 - ②: The account that is used to access the RDS database.
 - ③: The password of the account that is used to access the RDS database.
6. Click **Login**.

Note If you want the browser to remember the password, select **Remember your password** and click **Login**.

11.6.4. View and modify the internal endpoint and port number

You must use the internal endpoint and port number to access an RDS instance. This topic describes how to view and modify the internal endpoint and port number of an ApsaraDB for RDS instance in the ApsaraDB for RDS console.

View the internal endpoint and port number

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Basic Information** section, view the internal and public endpoints and port numbers.

Note The endpoints and port numbers are displayed only after you configure an [IP address whitelist or security group](#) for the RDS instance.

The screenshot shows the 'Basic Information' tab of an instance. The 'Internal Endpoint' field is highlighted with a red box, and the 'Internal Port' field is also highlighted with a red box. Other visible fields include Instance ID, Name, Region and Zone, Instance Role & Edition (Primary Instance (High-availability)), and Storage Type (Local SSD).

Modify the internal endpoint and port number

1. Log on to the [ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the right side of the page, click **Change Endpoint**.
6. In the dialog box that appears, set **Connection Type** to **Internal Endpoint**.
7. Modify the endpoint prefix and port number and then click **OK**.

Note

- The prefix of the endpoint must be 8 to 64 characters in length and can contain letters, digits, and hyphens (-). It must start with a lowercase letter.
- The port number must be within the range of 1000 to 5999.

The 'Change Endpoint' dialog box is shown. The 'Connection Type' dropdown is set to 'Internal Endpoint'. The 'Endpoint' field has a placeholder and a note: 'Starts with a lower-case letter, consists of 8 to 64 characters, including letters, digits, or hyphen (-)'. The 'Port' field contains '3433' and a note: 'Port Range: 1000 to 5999'. There are 'OK' and 'Cancel' buttons at the bottom.

FAQ

- **Q:** Do I need to modify the endpoint or port number in my application after I modify the endpoint or port number of an instance?
A: Yes, you must modify the endpoint or port number in the application after you have modified them. Otherwise, the application cannot connect to databases of the instance.
- **Q:** Does the modification of the endpoint take effect immediately? Do I need to restart the instance?
A: No, you do not need to restart the instance. The modification takes effect immediately.

11.7. Accounts

11.7.1. Create an account

Before you start to use ApsaraDB for RDS, you must create an account for the RDS instance. This topic describes how to create an account for a PolarDB instance.

Precautions

- Databases within the same instance share all of the resources that belong to the instance. Each RDS instance supports one initial account and multiple standard accounts. You can execute SQL statements to create and manage standard accounts and databases.
- To migrate an on-premises database to an RDS instance, you must create a database and an account with the same names on the RDS instance.
- Follow the least privilege principle to create accounts and grant them appropriate read-only and read/write permissions on databases based on the required roles. When necessary, you can create more than one account and allow each account to access only the data within its authorized workloads. If an account does not need to write data to a database, assign read-only permissions to the account.
- For security purposes, we recommend that you configure strong passwords for the accounts that you create and change the passwords on a regular basis.
- The initial account cannot be deleted after it is created.

Create an initial account

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Initial Account** and configure the following parameters.

Parameter	Description
Database Account	<ul style="list-style-type: none"> ◦ The name of the account must be 2 to 16 characters in length. ◦ The name of the account can contain lowercase letters, digits, and underscores (_). ◦ The name of the account must start with a letter and end with a letter or digit.
Password	<ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the same password again.

6. Click **Create**.

Create a standard account

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS) logon** page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.

The screenshot shows the DMS login page. At the top is the DMS logo. Below it are three input fields, each with a red circle containing a number: 1, 2, and 3. Field 1 is for IP Address:Port, field 2 is for Database Username, and field 3 is for the password. Below the fields is a checkbox for 'Remember your password' and a link for 'About DMS (Data Management Service)'. At the bottom is a blue 'Login' button.

Parameter	Description
IP Address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the name of the account used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP whitelist](#).

- In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
- In the SQL window, execute the following statement to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ]
where option can be:
    SUPERUSER | NOSUPERUSER
    | CREATEDB | NOCREATEDB
    | CREATEROLE | NOCREATEROLE
    | CREATEUSER | NOCREATEUSER
    | INHERIT | NOINHERIT
    | LOGIN | NOLOGIN
    | REPLICATION | NOREPLICATION
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | VALID UNTIL 'timestamp'
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

9. Click **execute**.

11.7.2. Reset the password

This topic describes how to reset the password of your database account in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. In the **Actions** column corresponding to the target account, click **Reset Password**.
6. In the **Reset Account Password** dialog box, enter a new password and click **OK**.

-  **Note** The password must meet the following requirements:
- The password must be 8 to 32 characters in length.
 - The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
 - Special characters include ! @ # \$ % ^ & * () _ + - =

11.8. Databases

11.8.1. Create a database

Before you start to use ApsaraDB for RDS, you must create a database for an RDS instance. This topic describes how to create a database for a PolarDB instance.

Prerequisites

A PolarDB instance has been created. For more information about how to create an instance, see [Create an instance](#).

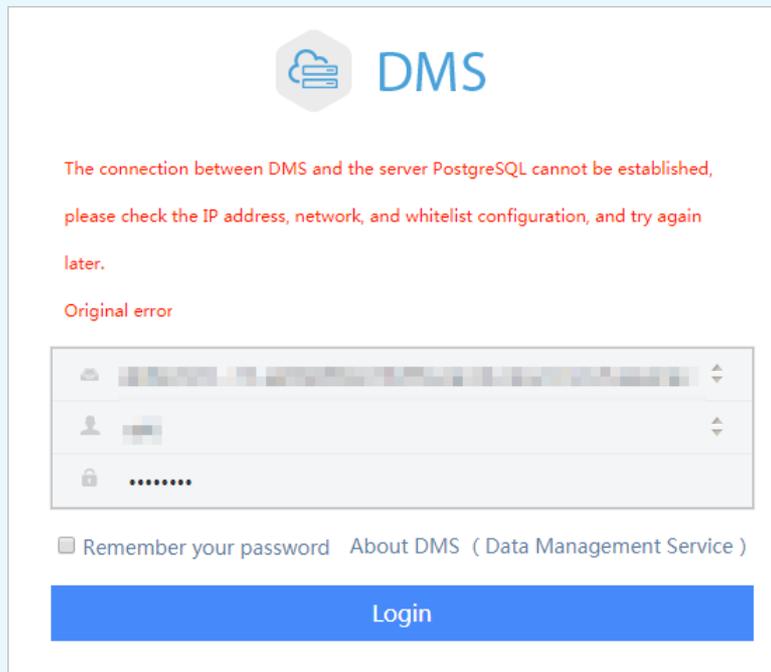
Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS)** logon page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.

Parameter	Description
IP Address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the name of the account used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP whitelist](#).



7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
8. In the SQL window, execute the following statement to create a database:

```
CREATE DATABASE name
[ [ WITH ] [ OWNER [=] user_name ]
  [ TEMPLATE [=] template ]
  [ ENCODING [=] encoding ]
  [ LC_COLLATE [=] lc_collate ]
  [ LC_CTYPE [=] lc_ctype ]
  [ TABLESPACE [=] tablespace_name ]
  [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

9. Click **execute**.

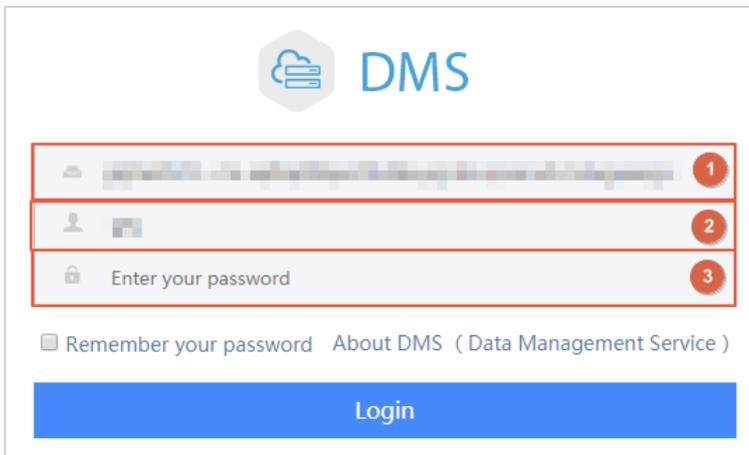
11.8.2. Delete a database

This topic describes how to delete a database from a PolarDB instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

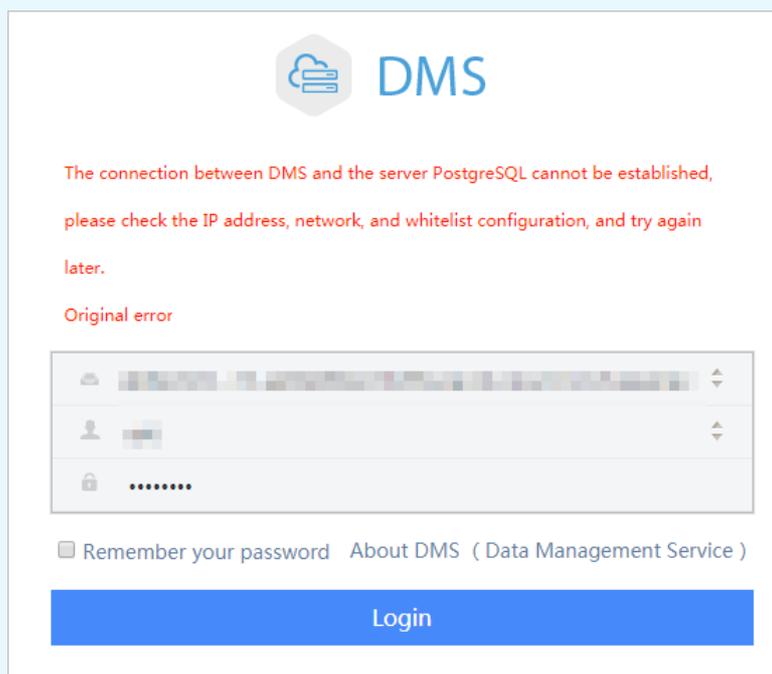
- Click **Log On to DB** in the upper-right corner of the page.
- On the Data Management (DMS) logon page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.



Parameter	Description
IP Address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the name of the account used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account used to access the database.

- Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP whitelist](#).



7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
8. Execute the following statement to delete the database:

```
drop database <database name>;
```

9. Click **execute**.

11.9. Network, VPC, and VSwitch

11.9.1. Change the network type

This topic describes how to change the network type of an ApsaraDB for RDS instance between classic network and VPC.

Context

- **Classic network:** RDS instances in the classic network are not isolated. Unauthorized access to these instances can be blocked only by whitelists.
- **VPC:** Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you select the VPC network type because it is more secure than the classic network.

You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can connect your own data center to a VPC by using leased lines or VPNs. This allows you to build a virtual data center on the cloud.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the internal endpoint of the RDS instance remains unchanged, but the IP address bound to the internal endpoint changes.
- After the network type is changed, ECS instances in the same VPC as the RDS instance can no longer connect to the RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- Changing the network type may result in a network interruption of 30 seconds. To avoid interruptions to your business, we recommend that you change the network type during off-peak hours or configure automatic reconnection policies for your applications.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Switch to Classic Network**.
6. In the dialog box that appears, click **OK**.

 **Note** After the network type is changed to classic network, only ECS instances in the classic network can access your RDS instance over an internal network. Configure the internal endpoint for these ECS instances.

7. Configure a whitelist of your RDS instance to allow access from the ECS instances over the internal network.

Note

- If the network isolation mode of the RDS instance is standard whitelist, add the internal IP addresses of the ECS instances to a whitelist of your RDS instance.
- If the network isolation mode of the RDS instance is **enhanced whitelist**, add the internal IP addresses of the ECS instances to a classic network whitelist. If no classic network whitelists are available, create a whitelist.

Change the network type from classic network to VPC

1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to VPC**.
6. In the **Switch to VPC** dialog box, select a VPC and VSwitch and specify whether to **Reserve Original Classic Network Endpoint**.

Note

- Select a VPC. We recommend that you select the VPC where your ECS instances reside. Otherwise, the ECS instances cannot communicate with the RDS instance over the internal network.
- Select a VSwitch. If no VSwitches are available in the selected VPC, create one in the same zone where the RDS instance resides. Use the following path to navigate the guide: *Virtual Private Cloud User Guide > Quick Start > Create a VSwitch*.
- Determine whether to select the **Reserve Original Classic Endpoint** option. The following table describes the details.

Operation	Description
Not selected	<p>The endpoint used in the classic network is replaced with an endpoint in the VPC.</p> <p>When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the RDS instance are interrupted.</p>
Selected	<p>The endpoint used in the classic network is retained, and a new endpoint to be used in the VPC is generated. In such cases, the RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the RDS instance over the internal network.</p> <p>When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the RDS instance will be available until the endpoint used in the classic network expires.</p> <p>To migrate your business to the VPC without interruption, you must add the new endpoint used in the VPC to access the ECS instances before the endpoint used in the classic network expires. Seven days before the endpoint used in the classic network expires, the system will send a text message to the phone number bound to your Alibaba Cloud account every day.</p> <p>For more information, see Hybrid access from both the classic network and VPCs.</p>

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the RDS instance over the internal network. If no VPC whitelists are available, create a

whitelist.

 **Note**

- If you have retained the classic network endpoint, add the VPC endpoint to the ECS instances before the classic network endpoint expires.
- If you have not retained the classic network endpoint, connections between ECS instances in the classic network and the RDS instance over the internal network are interrupted. You must add the new endpoint to ECS instances in the VPC immediately after the network type is changed.

11.9.2. Hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB for RDS to change the network type of an instance from classic network to Virtual Private Network (VPC) without network interruptions.

Prerequisites

- The network type of the instance is classic network.
- Available VPCs and VSwitches exist in the zone where the instance resides.

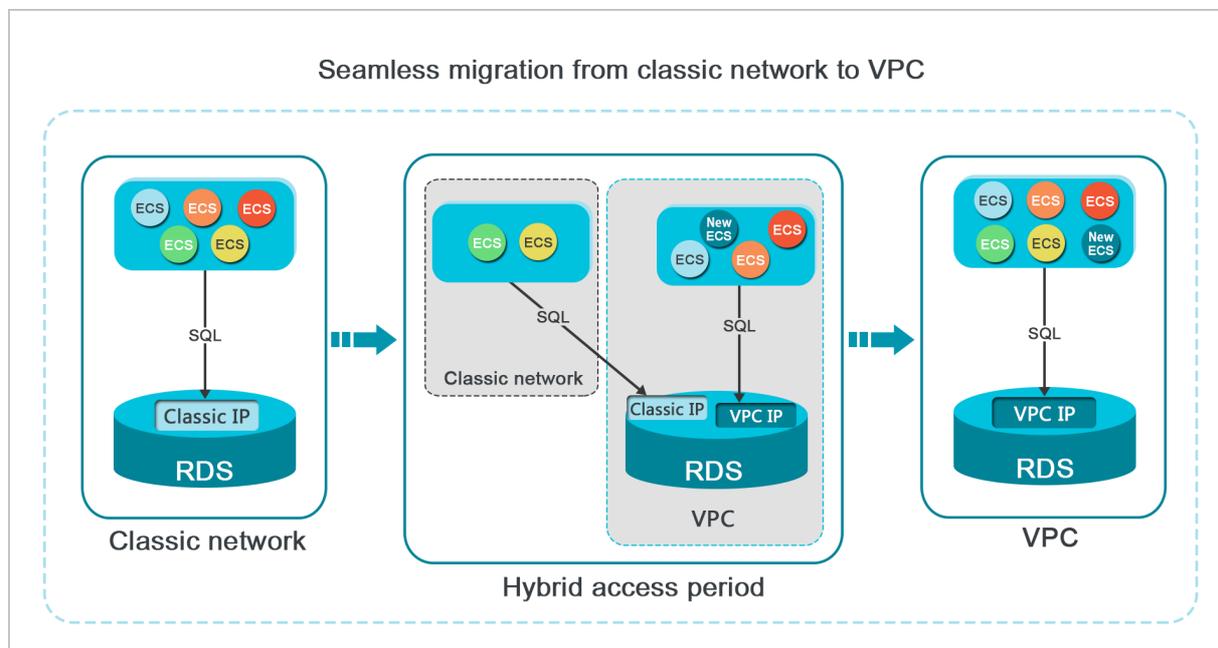
Context

In the past, when you changed the network type of an RDS instance from classic network to VPC, the internal endpoint of the RDS instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the RDS instance through the internal endpoint within this period. To smoothly change the network type, ApsaraDB for RDS provides the hybrid access solution.

Hybrid access refers to the ability of an RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimize the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database through the internal endpoint of VPCs, and the other applications can access the database through the original internal endpoint of the classic network. When all the applications access the database through the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- Changing to the classic network is not supported.
- Migrating the RDS instance to another zone is not supported.

Change the network type from classic network to VPC

For more information, see [Change the network type from classic network to VPC](#).

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be connected over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

Follow these steps to change the expiration time:

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Instance Connection** tab, click **Change Expiration Time**.
6. On the **Change Expiration Time** dialog box that appears, select an expiration time and click **OK**.

11.10. Monitoring

11.10.1. View monitoring data

ApsaraDB for RDS provides a wide range of performance metrics. You can view the resource monitoring data in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, select the time range to query the corresponding monitoring data. The following table lists the specific monitored metrics.

Monitoring metric	Description
Disk Space	The disk space usage of the instance. Unit: MB.
IOPS	The number of I/O requests of the data and log disks per second.
Memory Usage	The memory usage of the instance.
CPU Utilization	The CPU utilization of the instance.

11.10.2. Set the monitoring frequency

This topic describes how to set the monitoring frequency of a PolarDB instance.

Context

PolarDB supports the following monitoring frequencies:

- Every 5 seconds
- Every 60 seconds
- Every 300 seconds

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. In the upper-right corner of the **Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box, select the required monitoring frequency and click **OK**.

11.11. Data security

11.11.1. Switch to the enhanced whitelist mode

This topic describes how to switch from the standard whitelist mode to the enhanced whitelist mode for a PolarDB instance. The enhanced whitelist mode provides higher security.

Network isolation modes

RDS instances support the following network isolation modes:

- **Standard whitelist mode**

IP addresses from both the classic network and VPCs are added to the same whitelist. However, the standard whitelist mode may incur security risks. Therefore, we recommend that you use the enhanced whitelist mode.

- **Enhanced whitelist mode**

An enhanced IP address whitelist can contain only IP addresses from the classic network or VPCs. When you create an IP address whitelist, you must specify its network type.

Changes after you switch to the enhanced whitelist mode

- If the network type of the instance is VPC, a new whitelist is created and contains the same IP addresses as the original whitelists. The new IP whitelist only applies to VPCs.
- If the network type of the instance is classic network, a new whitelist is created and contains the same IP addresses as the original whitelists. The new IP whitelist only applies to the classic network.
- If the instance supports **access from both the classic network and VPCs**, two new whitelists are created, and each contains the same IP addresses as the original whitelists. One whitelist applies to VPCs, and the other applies to the classic network.

 **Note** Switching to enhanced whitelist mode does not affect the ECS instances that are in the security group.

Precautions

- You can switch from standard whitelist mode to enhanced whitelist mode, but not the other way around.
- In enhanced whitelist mode, a classic network whitelist also allows access from the Internet. If you want to access an RDS instance from a host over the Internet, you can add the public IP address of the host to a classic network whitelist.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Switch to Enhanced Whitelist (Recommended)**.
6. In the **Enable Enhanced Whitelist** dialog box, click **Confirm**.

11.11.2. Configure an IP whitelist

This topic describes how to configure a whitelist for an ApsaraDB for RDS instance. Only devices that are listed in a whitelist can access your RDS instance.

Context

Whitelists make your RDS instance more secure and do not interrupt the operations of your RDS instance during configuration. We recommend that you update the IP address whitelists and security groups configured for your RDS instance on a regular basis.

You can use one of the following methods to configure a whitelist.

- **Configure a whitelist:** Add IP addresses to grant them access to the RDS instance.

 **Note** The default IP address whitelist contains only the IP address 127.0.0.1. This indicates that no devices are allowed to access the RDS instance.

- Configure an ECS security group: Add an ECS security group for the RDS instance to allow ECS instances in the group to access the RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**. On the page that appears, click the **Whitelist Settings** tab.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

 **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, enter the IP addresses or CIDR blocks that are allowed to access the instance and click **OK**. The following section describes the rules:
 - If you enter the CIDR block 10.10.10.0/24 in the **IP Addresses** field, all IP addresses in the 10.10.10.X format can access your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all created ECS instances within your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

11.12. Backup

11.12.1. Back up data

This topic describes how to back up a PolarDB instance. You can configure a backup policy that is used to automatically back up your PolarDB instance. If you do not configure a backup policy, the default backup policy is used. You can also manually back up your PolarDB instance.

Precautions

- Do not execute data definition language (DDL) statements during a backup. If you do so, the backup may fail due to table locks.
- We recommend that you back up your instance during off-peak hours.
- If the amount of data is large, it may take a long time to back up your RDS instance.
- Backup files are retained for a specified retention period. We recommend that you download the required backup files to your computer before they are deleted.

Overview of data and log backups

Database engine	Data backup	Log backup
PolarDB	Supports full physical backup.	Write-ahead logs (WALs) are compressed and uploaded immediately after they are generated. Each log takes up 16 MB of storage space. On-premises logs are deleted within 24 hours.

Configure a backup policy to automatically back up your PolarDB instance

ApsaraDB for RDS automatically backs up your instance based on the specified backup policy.

1. [Log on to the ApsaraDB for RDS console.](#)

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Settings** tab and then click the **Edit** button.
6. In the dialog box that appears, configure the following parameters, and then click **OK**. The following table lists the parameters.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Unit: days. Default value: 7.
Backup Cycle	The cycle to create backups. You can select one or more days of a week. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> ? Note To ensure data security, we recommend that you back up your RDS instance at least twice a week. </div>
Backup Time	The hour at which you want to create a backup.
Log Backup	The switch to enable or disable the log backup function. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> 🔊 Notice If you disable this function, all log backup files are deleted and your instance cannot be restored to previous points in time. </div>
Log Retention Period	<ul style="list-style-type: none"> ○ The period of time for which you want to retain log backup files. Valid values: 7 to 730. Unit: days. Default value: 7. ○ The log retention period must be less than or equal to the data retention period.

Back up data manually

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Back Up Instance**.
5. Select the backup mode and backup policy, and click **OK**.

? **Note** The backup mode is **Full Backup** and the backup policy is **Instance Backup**.

6. In the upper-right corner, click the  icon to view the task progress displayed in the **Task Progress** list.

? **Note** You cannot download backup files to your computer.

FAQ

1. Q: Can I disable data backup for a PolarDB instance?

A: No, you cannot disable the data backup function of your PolarDB instance. However, you can reduce the backup frequency to as low as twice a week. The data retention period must be within the range of 7 days to 730 days.

2. Q: Can I disable log backup for a PolarDB instance?

A: Yes, you can disable the log backup function of your PolarDB instance. You can log on to the ApsaraDB for RDS console and navigate to the Backup Settings tab to disable the log backup function of your instance.

11.12.2. Download backup files

This topic describes how to download unencrypted log backup files of a PolarDB instance.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**. The **Backup and Restoration** page appears.
5. Select the **Archived Logs** tab, specify the time range, find the target log backup file, and click **Download** in the Actions column.

Note If you want to download a log backup file that is used to restore data to an on-premises database, make sure that the file meets the following requirements:

- The Instance ID of the log file on the Archived Logs tab must be the same as the Instance No. displayed on the Data Backup tab.
- The start time of the file must be later than the start time of the specified time range. It must also be earlier than the point in time to which you want to restore data.

6. In the **Download Binary Log** dialog box, select the download method.

Download method	Description
Download	Download the backup file.
Copy Internal Endpoint	Copy the internal endpoint that is used to download the backup file. If your ECS and RDS instances reside in the same region, you can log on to the ECS instance. Then, you can use the internal endpoint to download the backup file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint that is used to download the backup file. If you want to use other tools to download the backup file, copy the public endpoint.

Note If you are using a Linux operating system, you can run the following command to download the backup file:

```
wget -c '<The endpoint from which you can download the backup file>' -O <The name of the backup file>
```

- The **-c** option enables resumable download.
- The **-O** option saves the downloaded file by using a specified name. We recommend that you use the file name contained in the download URL.
- If the endpoint contains more than one parameter, enclose the endpoint in a pair of single quotation marks (').

```
[root@i-06Expres=1576Signature=Y8%3D' -O hins8641051_data_20191112155605_qp.xb]# wget -c 'http://rdsbak-ou.aliyuncs.com/xxxxx/hins8641051_data_20191112155605_qp.xb?OSSAc
```

11.13. Logs

The primary/secondary switching logs of a PolarDB instance can be used for troubleshooting. This topic describes how to manage logs of a PolarDB instance in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Logs**.
5. On the **Logs** page, click the **Slow Query Logs** or **Primary/Secondary Switching Logs** tab, select a time range, and click **Search**.

Log type	Description
Slow query log	Records SQL statements that took more than one second to execute in the last month. Duplicated SQL statements are removed.
Primary/secondary switching log	Records switchovers between the primary and secondary instances in the last month.

11.14. Plug-ins supported

This topic lists the plug-ins and plug-in versions supported by PolarDB.

Plug-ins and versions supported by PolarDB

Plug-in	Version
btree_gin	1.3
btree_gist	1.5
citext	1.5
cube	1.4
dict_int	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hstore	1.5
intagg	1.1
intarray	1.2
isn	1.2
ltree	1.1
pg_buffercache	1.3
pg_pathman	1.5

Plug-in	Version
pg_prewarm	1.2
pg_stat_statements	1.6
pg_trgm	1.4
pg_wait_sampling	1.1
pgcrypto	1.3
pgrowlocks	1.2
pgstattuple	1.5
plpgsql	1.0
sslinfo	1.2
tablefunc	1.0
unaccent	1.1
uuid-oss	1.1
zhparser	1.0
ganos_geometry	2.3
ganos_raster	2.3
ganos_geometry_sfcgal	2.3
ganos_geometry_topology	2.3
ganos_tiger_geocode	2.3
ganos_address_standardizer	2.3
ganos_address_standardizer_data_us	2.3
ganos_networking	2.3
ganos_pointcloud	2.3
ganos_trajectory	2.3
plperl	1.0
pltcl	1.0

 **Note** PolarDB updates its kernel to support new plug-ins or plug-in versions. To view the supported plug-ins, execute the following statement:

```
show polar_supported_extensions;
```

11.15. PolarDB development driver

This topic describes how to configure the PolarDB development driver.

The PolarDB development driver provides a variety of driver interfaces for application development:

- For Linux applications, the interfaces include Java interfaces, Oracle Call Interface (OCI), and Open Database Connectivity (ODBC).
- For Windows applications, the interfaces include .Net interfaces, Java interfaces, OCI, and ODBC.

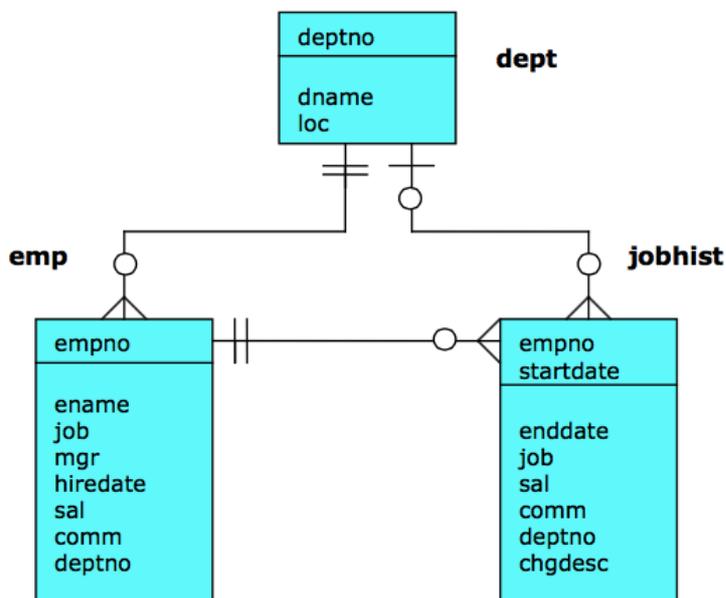
Click [here](#) to download the PolarDB development driver.

- The driver contains the following files:
 - edb_connectors-9.3.5.14-3-linux-x64.run
 - edb_connectors-9.3.5.14-3-linux.run
 - edb_connectors-9.3.5.14-3-windows-x64.exe
 - edb_connectors-9.3.5.14-3-windows.exe
- By default, the driver is installed to the following path:
 - Linux: `/opt/PostgresPlus/9.3AS/connectors`
 - Windows: `C:/Program Files/PostgresPlus/9.3AS/connectors`

11.16. Compatibility for Oracle

This topic helps you understand terms and concepts that are used in PolarDB by examples. You can effectively develop PolarDB databases and migrate data from Oracle to PolarDB databases.

The operations described in this topic are built on a data model that displays basic PolarDB operations, such as database creation, table creation, and user management. The following figure shows the data model:



To simulate an environment similar to Oracle, the following example creates a database named `orcl_polardb`, a role named `scott` in the `orcl_polardb` database, and a schema named `scott`.

Use `psql` to connect to a database

```
psql -h polardbaddress.polaradb.rds.aliyuncs.com -p 3433 -U myuser -d template1
Enter the password of the myuser user:
psql.bin (9.4.1.3, server 9.3.5.14)
Enter help to obtain help information.
template1=>
```

CREATE DATABASE

```
template1=> CREATE DATABASE orcl_polardb;  
CREATE DATABASE  
template1=> \c orcl_polardb  
psql.bin (9.4.1.3, server 9.3.5.14)
```

CREATE ROLE

```
orcl_polardb=> CREATE ROLE scott LOGIN PASSWORD 'scott123';  
CREATE ROLE
```

CREATE SCHEMA

```
orcl_polardb=> CREATE SCHEMA scott;  
CREATE SCHEMA  
orcl_polardb=> GRANT scott TO myuser;  
GRANT ROLE  
orcl_polardb=> ALTER SCHEMA scott OWNER TO scott;  
ALTER SCHEMA  
orcl_polardb=> REVOKE scott FROM myuser;  
REVOKE ROLE
```

Note

- If you have not granted the permissions owned by the scott role to the myuser user when you execute the `ALTER SCHEMA scott OWNER TO scott` statement, the `ERROR:must be member of role "scott"` permission error is displayed.
- For security reasons, revoke the scott permissions from the myuser user after the statement is executed. This provides improved security.

Connect to the orcl_polardb database

 **Note** The following operations must be performed under the scott account. Otherwise, the created tables and objects do not belong to scott and permission errors may occur.

```
[root@localhost bin]# ./psql -h polardbaddress.polardb.rds.aliyuncs.com -p 3433 -U scott -d orcl_polardb  
Enter the password of the user scott:  
psql.bin (9.4.1.3, server 9.3.5.14)  
Enter help to obtain help information.  
orcl_polardb=>
```

CREATE TABLE

```

CREATE TABLE dept (
  deptno    NUMBER(2) NOT NULL CONSTRAINT dept_pk PRIMARY KEY,
  dname     VARCHAR2(14) CONSTRAINT dept_dname_uq UNIQUE,
  lock      VARCHAR2(13)
);
CREATE TABLE emp (
  empno     NUMBER(4) NOT NULL CONSTRAINT emp_pk PRIMARY KEY,
  ename     VARCHAR2(10),
  job       VARCHAR2(9),
  mgr       NUMBER(4),
  hiredate  DATE,
  sal       NUMBER(7,2) CONSTRAINT emp_sal_ck CHECK (sal > 0),
  comm      NUMBER(7,2),
  deptno    NUMBER(2) CONSTRAINT emp_ref_dept_fk
            REFERENCES dept(deptno)
);
CREATE TABLE jobhist (
  empno     NUMBER(4) NOT NULL,
  startdate DATE NOT NULL,
  enddate   DATE,
  job       VARCHAR2(9),
  sal       NUMBER(7,2),
  comm      NUMBER(7,2),
  deptno    NUMBER(2),
  chgdesc   VARCHAR2(80),
  CONSTRAINT jobhist_pk PRIMARY KEY (empno, startdate),
  CONSTRAINT jobhist_ref_emp_fk FOREIGN KEY (empno)
            REFERENCES emp(empno) ON DELETE CASCADE,
  CONSTRAINT jobhist_ref_dept_fk FOREIGN KEY (deptno)
            REFERENCES dept (deptno) ON DELETE SET NULL,
  CONSTRAINT jobhist_date_chk CHECK (startdate <= enddate)
);

```

CREATE OR REPLACE VIEW

```

CREATE OR REPLACE VIEW salesemp AS
  SELECT empno, ename, hiredate, sal, comm FROM emp WHERE job = 'SALESMAN';

```

CREATE SEQUENCE

```

CREATE SEQUENCE next_empno START WITH 8000 INCREMENT BY 1;

```

INSERT INTO

```

INSERT INTO dept VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept VALUES (30,'SALES','CHICAGO');
INSERT INTO dept VALUES (40,'OPERATIONS','BOSTON');
INSERT INTO emp VALUES (7369,'SMITH','CLERK',7902,'17-DEC-80',800,NULL,20);
INSERT INTO emp VALUES (7499,'ALLEN','SALESMAN',7698,'20-FEB-81',1600,300,30);
INSERT INTO emp VALUES (7521,'WARD','SALESMAN',7698,'22-FEB-81',1250,500,30);
INSERT INTO emp VALUES (7566,'JONES','MANAGER',7839,'02-APR-81',2975,NULL,20);
INSERT INTO emp VALUES (7654,'MARTIN','SALESMAN',7698,'28-SEP-81',1250,1400,30);
INSERT INTO emp VALUES (7698,'BLAKE','MANAGER',7839,'01-MAY-81',2850,NULL,30);
INSERT INTO emp VALUES (7782,'CLARK','MANAGER',7839,'09-JUN-81',2450,NULL,10);
INSERT INTO emp VALUES (7788,'SCOTT','ANALYST',7566,'19-APR-87',3000,NULL,20);
INSERT INTO emp VALUES (7839,'KING','PRESIDENT',NULL,'17-NOV-81',5000,NULL,10);
INSERT INTO emp VALUES (7844,'TURNER','SALESMAN',7698,'08-SEP-81',1500,0,30);
INSERT INTO emp VALUES (7876,'ADAMS','CLERK',7788,'23-MAY-87',1100,NULL,20);
INSERT INTO emp VALUES (7900,'JAMES','CLERK',7698,'03-DEC-81',950,NULL,30);
INSERT INTO emp VALUES (7902,'FORD','ANALYST',7566,'03-DEC-81',3000,NULL,20);
INSERT INTO emp VALUES (7934,'MILLER','CLERK',7782,'23-JAN-82',1300,NULL,10);
INSERT INTO jobhist VALUES (7369,'17-DEC-80',NULL,'CLERK',800,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7499,'20-FEB-81',NULL,'SALESMAN',1600,300,30,'New Hire');
INSERT INTO jobhist VALUES (7521,'22-FEB-81',NULL,'SALESMAN',1250,500,30,'New Hire');
INSERT INTO jobhist VALUES (7566,'02-APR-81',NULL,'MANAGER',2975,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7654,'28-SEP-81',NULL,'SALESMAN',1250,1400,30,'New Hire');
INSERT INTO jobhist VALUES (7698,'01-MAY-81',NULL,'MANAGER',2850,NULL,30,'New Hire');
INSERT INTO jobhist VALUES (7782,'09-JUN-81',NULL,'MANAGER',2450,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7788,'19-APR-87','12-APR-88','CLERK',1000,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7788,'13-APR-88','04-MAY-89','CLERK',1040,NULL,20,'Raise');
INSERT INTO jobhist VALUES (7788,'05-MAY-90',NULL,'ANALYST',3000,NULL,20,'Promoted to Analyst');
INSERT INTO jobhist VALUES (7839,'17-NOV-81',NULL,'PRESIDENT',5000,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7844,'08-SEP-81',NULL,'SALESMAN',1500,0,30,'New Hire');
INSERT INTO jobhist VALUES (7876,'23-MAY-87',NULL,'CLERK',1100,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7900,'03-DEC-81','14-JAN-83','CLERK',950,NULL,10,'New Hire');
INSERT INTO jobhist VALUES (7900,'15-JAN-83',NULL,'CLERK',950,NULL,30,'Changed to Dept 30');
INSERT INTO jobhist VALUES (7902,'03-DEC-81',NULL,'ANALYST',3000,NULL,20,'New Hire');
INSERT INTO jobhist VALUES (7934,'23-JAN-82',NULL,'CLERK',1300,NULL,10,'New Hire');

```

ANALYZE

```

ANALYZE dept;
ANALYZE emp;
ANALYZE jobhist;

```

CREATE PROCEDURE

```

CREATE OR REPLACE PROCEDURE list_emp
IS
v empno NUMBER(4);

```

```

v_ename    VARCHAR2(10);
CURSOR emp_cur IS
    SELECT empno, ename FROM emp ORDER BY empno;
BEGIN
    OPEN emp_cur;
    DBMS_OUTPUT.PUT_LINE('EMPNO  ENAME');
    DBMS_OUTPUT.PUT_LINE('-----  -----');
    LOOP
        FETCH emp_cur INTO v_empno, v_ename;
        EXIT WHEN emp_cur%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_empno || ' ' || v_ename);
    END LOOP;
    CLOSE emp_cur;
END;
--
-- Procedure that selects an employee row given the employee
-- number and displays certain columns.
--
CREATE OR REPLACE PROCEDURE select_emp (
    p_empno    IN NUMBER
)
IS
    v_ename    emp.ename%TYPE;
    v_hiredate emp.hiredate%TYPE;
    v_sal      emp.sal%TYPE;
    v_comm     emp.comm%TYPE;
    v_dname    dept.dname%TYPE;
    v_disp_date VARCHAR2(10);
BEGIN
    SELECT ename, hiredate, sal, NVL(comm, 0), dname
        INTO v_ename, v_hiredate, v_sal, v_comm, v_dname
        FROM emp e, dept d
        WHERE empno = p_empno
            AND e.deptno = d.deptno;
    v_disp_date := TO_CHAR(v_hiredate, 'MM/DD/YYYY');
    DBMS_OUTPUT.PUT_LINE('Number   : ' || p_empno);
    DBMS_OUTPUT.PUT_LINE('Name     : ' || v_ename);
    DBMS_OUTPUT.PUT_LINE('Hire Date : ' || v_disp_date);
    DBMS_OUTPUT.PUT_LINE('Salary   : ' || v_sal);
    DBMS_OUTPUT.PUT_LINE('Commission: ' || v_comm);
    DBMS_OUTPUT.PUT_LINE('Department: ' || v_dname);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee ' || p_empno || ' not found');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');

```

```

DBMS_OUTPUT.PUT_LINE(SQLERRM);
DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
DBMS_OUTPUT.PUT_LINE(SQLCODE);
END;
--
-- Procedure that queries the 'emp' table based on
-- department number and employee number or name. Returns
-- employee number and name as IN OUT parameters and job,
-- hire date, and salary as OUT parameters.
--
CREATE OR REPLACE PROCEDURE emp_query (
  p_deptno  IN  NUMBER,
  p_empno   IN OUT NUMBER,
  p_ename   IN OUT VARCHAR2,
  p_job     OUT  VARCHAR2,
  p_hiredate OUT DATE
  p_sal     OUT  NUMBER
)
IS
BEGIN
  SELECT empno, ename, job, hiredate, sal
    INTO p_empno, p_ename, p_job, p_hiredate, p_sal
    FROM emp
    WHERE deptno = p_deptno
    AND (empno = p_empno
    OR ename = UPPER(p_ename));
END;
--
-- Procedure to call 'emp_query_caller' with IN and IN OUT
-- parameters. Displays the results received from IN OUT and
-- OUT parameters.
--
CREATE OR REPLACE PROCEDURE emp_query_caller
IS
  v_deptno  NUMBER(2);
  v_empno   NUMBER(4);
  v_ename   VARCHAR2(10);
  v_job     VARCHAR2(9);
  v_hiredate DATE;
  v_sal     NUMBER;
BEGIN
  v_deptno := 30;
  v_empno := 0;
  v_ename := 'Martin';
  emp_query(v_deptno, v_empno, v_ename, v_job, v_hiredate, v_sal);
  DBMS_OUTPUT.PUT_LINE('Department : ' || v_deptno);
  DBMS_OUTPUT.PUT_LINE('Employee No: ' || v_empno);

```

```

DBMS_OUTPUT.PUT_LINE('Name      : ' || v_ename);
DBMS_OUTPUT.PUT_LINE('Job       : ' || v_job);
DBMS_OUTPUT.PUT_LINE('Hire Date : ' || v_hiredate);
DBMS_OUTPUT.PUT_LINE('Salary   : ' || v_sal);
EXCEPTION
WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('More than one employee was selected');
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No employees were selected');
END;
```

CREATE FUNCTION

```

CREATE OR REPLACE FUNCTION emp_comp (
    p_sal      NUMBER,
    p_comm     NUMBER
) RETURN NUMBER
IS
BEGIN
    RETURN (p_sal + NVL(p_comm, 0)) * 24;
END;
--
-- Function that gets the next number from sequence, 'next_empno',
-- and ensures it is not already in use as an employee number.
--
CREATE OR REPLACE FUNCTION new_empno RETURN NUMBER
IS
    v_cnt      INTEGER := 1;
    v_new_empno NUMBER;
BEGIN
    WHILE v_cnt > 0 LOOP
        SELECT next_empno.nextval INTO v_new_empno FROM dual;
        SELECT COUNT(*) INTO v_cnt FROM emp WHERE empno = v_new_empno;
    END LOOP;
    RETURN v_new_empno;
END;
--
-- EDB-SPL function that adds a new clerk to table 'emp'. This function
-- uses package 'emp_admin'.
--
CREATE OR REPLACE FUNCTION hire_clerk (
    p_ename    VARCHAR2,
    p_deptno   NUMBER
) RETURN NUMBER
IS
    v_empno    NUMBER(4);
    v_ename    VARCHAR2(10);
```

```

v_job      VARCHAR2(9);
v_mgr      NUMBER(4);
v_hiredate DATE;
v_sal      NUMBER(7,2);
v_comm     NUMBER(7,2);
v_deptno   NUMBER(2);
BEGIN
v_empno := new_empno;
INSERT INTO emp VALUES (v_empno, p_ename, 'CLERK', 7782,
    TRUNC(SYSDATE), 950.00, NULL, p_deptno);
SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno INTO
    v_empno, v_ename, v_job, v_mgr, v_hiredate, v_sal, v_comm, v_deptno
    FROM emp WHERE empno = v_empno;
DBMS_OUTPUT.PUT_LINE('Department : ' || v_deptno);
DBMS_OUTPUT.PUT_LINE('Employee No: ' || v_empno);
DBMS_OUTPUT.PUT_LINE('Name      : ' || v_ename);
DBMS_OUTPUT.PUT_LINE('Job       : ' || v_job);
DBMS_OUTPUT.PUT_LINE('Manager  : ' || v_mgr);
DBMS_OUTPUT.PUT_LINE('Hire Date : ' || v_hiredate);
DBMS_OUTPUT.PUT_LINE('Salary   : ' || v_sal);
DBMS_OUTPUT.PUT_LINE('Commission : ' || v_comm);
RETURN v_empno;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
        DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
        DBMS_OUTPUT.PUT_LINE(SQLCODE);
        RETURN -1;
END;
--
-- PostgreSQL PL/pgSQL function that adds a new salesman
-- to table 'emp'.
--
CREATE OR REPLACE FUNCTION hire_salesman (
    p_ename   VARCHAR,
    p_sal     NUMERIC,
    p_comm    NUMERIC
) RETURNS NUMERIC
AS $$
DECLARE
    v_empno   NUMERIC(4);
    v_ename   VARCHAR(10);
    v_job     VARCHAR(9);
    v_mgr     NUMERIC(4);
    v_hiredate DATE;

```

```

v_sal      NUMERIC(7,2);
v_comm     NUMERIC(7,2);
v_deptno   NUMERIC(2);
BEGIN
v_empno := new_empno();
INSERT INTO emp VALUES (v_empno, p_ename, 'SALESMAN', 7698,
    CURRENT_DATE, p_sal, p_comm, 30);
SELECT INTO
    v_empno, v_ename, v_job, v_mgr, v_hiredate, v_sal, v_comm, v_deptno
    empno, ename, job, mgr, hiredate, sal, comm, deptno
    FROM emp WHERE empno = v_empno;
RAISE INFO 'Department : %', v_deptno;
RAISE INFO 'Employee No: %', v_empno;
RAISE INFO 'Name      : %', v_ename;
RAISE INFO 'Job       : %', v_job;
RAISE INFO 'Manager   : %', v_mgr;
RAISE INFO 'Hire Date  : %', v_hiredate;
RAISE INFO 'Salary    : %', v_sal;
RAISE INFO 'Commission : %', v_comm;
RETURN v_empno;
EXCEPTION
WHEN OTHERS THEN
    RAISE INFO 'The following is SQLERRM:';
    RAISE INFO '%', SQLERRM;
    RAISE INFO 'The following is SQLSTATE:';
    RAISE INFO '%', SQLSTATE;
    RETURN -1;
END;

```

CREATE RULE

```

CREATE OR REPLACE RULE salesemp_i AS ON INSERT TO salesemp
DO INSTEAD
    INSERT INTO emp VALUES (NEW.empno, NEW.ename, 'SALESMAN', 7698,
        NEW.hiredate, NEW.sal, NEW.comm, 30);
CREATE OR REPLACE RULE salesemp_u AS ON UPDATE TO salesemp
DO INSTEAD
    UPDATE emp SET empno = NEW.empno,
        ename = NEW.ename,
        hiredate = NEW.hiredate,
        sal = NEW.sal,
        comm = NEW.comm
    WHERE empno = OLD.empno;
CREATE OR REPLACE RULE salesemp_d AS ON DELETE TO salesemp
DO INSTEAD
    DELETE FROM emp WHERE empno = OLD.empno;

```

CREATE TRIGGER

```
CREATE OR REPLACE TRIGGER user_audit_trig
  AFTER INSERT OR UPDATE OR DELETE ON emp
DECLARE
  v_action  VARCHAR2(24);
BEGIN
  IF INSERTING THEN
    v_action := ' added employee(s) on ';
  ELSIF UPDATING THEN
    v_action := ' updated employee(s) on ';
  ELSIF DELETING THEN
    v_action := ' deleted employee(s) on ';
  END IF;
  DBMS_OUTPUT.PUT_LINE('User ' || USER || v_action || TO_CHAR(SYSDATE,'YYYY-MM-DD'));
END;

CREATE OR REPLACE TRIGGER emp_sal_trig
  BEFORE DELETE OR INSERT OR UPDATE ON emp
  FOR EACH ROW
DECLARE
  sal_diff  NUMBER;
BEGIN
  IF INSERTING THEN
    DBMS_OUTPUT.PUT_LINE('Inserting employee ' || :NEW.empno);
    DBMS_OUTPUT.PUT_LINE('..New salary: ' || :NEW.sal);
  END IF;
  IF UPDATING THEN
    sal_diff := :NEW.sal - :OLD.sal;
    DBMS_OUTPUT.PUT_LINE('Updating employee ' || :OLD.empno);
    DBMS_OUTPUT.PUT_LINE('..Old salary: ' || :OLD.sal);
    DBMS_OUTPUT.PUT_LINE('..New salary: ' || :NEW.sal);
    DBMS_OUTPUT.PUT_LINE('..Raise   : ' || sal_diff);
  END IF;
  IF DELETING THEN
    DBMS_OUTPUT.PUT_LINE('Deleting employee ' || :OLD.empno);
    DBMS_OUTPUT.PUT_LINE('..Old salary: ' || :OLD.sal);
  END IF;
END;
```

CREATE PACKAGE

```

CREATE OR REPLACE PACKAGE emp_admin
IS
  FUNCTION get_dept_name (
    p_deptno    NUMBER
  ) RETURN VARCHAR2;
  FUNCTION update_emp_sal (
    p_empno     NUMBER,
    p_raise     NUMBER
  ) RETURN NUMBER;
  PROCEDURE hire_emp (
    p_empno     NUMBER,
    p_ename     VARCHAR2,
    p_job       VARCHAR2,
    p_sal       NUMBER,
    p_hiredate  DATE,
    p_comm      NUMBER,
    p_mgr       NUMBER,
    p_deptno    NUMBER
  );
  PROCEDURE fire_emp (
    p_empno     NUMBER
  );
END emp_admin;

```

CREATE PACKAGE BODY

```

--
-- Package body for the 'emp_admin' package.
--
CREATE OR REPLACE PACKAGE BODY emp_admin
IS
  --
  -- Function that queries the 'dept' table based on the department
  -- number and returns the corresponding department name.
  --
  FUNCTION get_dept_name (
    p_deptno    IN NUMBER
  ) RETURN VARCHAR2
  IS
    v_dname     VARCHAR2(14);
  BEGIN
    SELECT dname INTO v_dname FROM dept WHERE deptno = p_deptno;
    RETURN v_dname;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Invalid department number ' || p_deptno);
  RETURN '';

```

```

RETURN ,
END;
--
-- Function that updates an employee's salary based on the
-- employee number and salary increment/decrement passed
-- as IN parameters. Upon successful completion the function
-- returns the new updated salary.
--
FUNCTION update_emp_sal (
  p_empno    IN NUMBER,
  p_raise    IN NUMBER
) RETURN NUMBER
IS
  v_sal      NUMBER := 0;
BEGIN
  SELECT sal INTO v_sal FROM emp WHERE empno = p_empno;
  v_sal := v_sal + p_raise;
  UPDATE emp SET sal = v_sal WHERE empno = p_empno;
  RETURN v_sal;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee ' || p_empno || ' not found');
    RETURN -1;
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('The following is SQLERRM:');
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
    DBMS_OUTPUT.PUT_LINE('The following is SQLCODE:');
    DBMS_OUTPUT.PUT_LINE(SQLCODE);
    RETURN -1;
END;
--
-- Procedure that inserts a new employee record into the 'emp' table.
--
PROCEDURE hire_emp (
  p_empno    NUMBER,
  p_ename    VARCHAR2,
  p_job      VARCHAR2,
  p_sal      NUMBER,
  p_hiredate DATE,
  p_comm     NUMBER,
  p_mgr      NUMBER,
  p_deptno   NUMBER
)
AS
BEGIN
  INSERT INTO emp(empno, ename, job, sal, hiredate, comm, mgr, deptno)
    VALUES(p_empno, p_ename, p_job, p_sal,

```

```
        p_hiredate, p_comm, p_mgr, p_deptno);
END;
--
-- Procedure that deletes an employee record from the 'emp' table based
-- on the employee number.
--
PROCEDURE fire_emp (
    p_empno    NUMBER
)
AS
BEGIN
    DELETE FROM emp WHERE empno = p_empno;
END;
END;
```

11.17. Management functions

You cannot use superuser accounts to manage database objects of PolarDB. Therefore, PolarDB provides management functions to help you use various PolarDB features. This topic describes how to use the management functions.

Rules of management functions

You must use the root account of ApsaraDB for RDS to run management functions. The root account is a management account specified when an instance is created and has the CREATEDB, CREATEROLE, and LOGIN permissions.

- **rds_manage_extension**

This function enables you to manage plug-ins. You can use this function to create and delete plug-ins supported by PolarDB.

```
rds_manage_extension(operation text, pname text, schema text default NULL, logging bool default false)
```

operation: The parameter value is create or drop.

pname: the name of the supported plugin.

schema: the target plug-in mode.

logging: the log information when the plug-in is created.

The following plug-ins are supported:

pg_stat_statements

btree_gin

btree_gist

chkpass

citext

cube

dblink

dict_int

earthdistance

hstore

intagg

intarray

isn

ltree

pgcrypto

pgrowlocks

pg_prewarm

pg_trgm

postgres_fdw

sslinfo

tablefunc

tsearch2

unaccent

postgis

postgis_topology

fuzzystrmatch

postgis_tiger_geocoder

plperl

pltcl

plv8

"uuid-oss"p

plpgsql

oss_fdw

Examples:

1. Create the dblink plug-in.

```
select rds_manage_extension('create','dblink');
```

2. Delete the dblink plug-in.

```
select rds_manage_extension('drop','dblink');
```

- rds_pg_stat_activity()

This function returns the information of all connected sessions, which is similar to the `pg_stat_activity` view.

- `rds_pg_stat_statements()`

This function encapsulates the `pg_stat_statements` view. You can use this function to view slow SQL statements within your permissions.

- Performance analysis functions

The functions enable you to analyze the real-time performance of PolarDB instances and are similar to Automatic Workload Repository (AWR) of Oracle.

1. `rds_truncsnap()`

Description: The function is used to delete stored snapshots.

2. `rds_get_snaps()`

Description: The function is used to obtain the information of stored snapshots.

3. `rds_snap()`

Description: The function is used to generate a real-time snapshot.

4. `rds_report(beginsnap bigint, endsnap bigint)`

Description: The function is used to generate a performance analysis report based on snapshots that you specify.

Example: The following statements generates a performance analysis report based on snapshots.

```
SELECT * FROM rds_truncsnap(); // Delete stored snapshots.
```

```
SELECT * from rds_snap(); // Generate a snapshot.
```

```
SELECT * from rds_snap(); // Generate a snapshot.
```

```
SELECT * from rds_snap(); // Generate a snapshot.
```

```
SELECT * FROM rds_get_snaps(); // Obtain the IDs of the generated snapshots, which are 1, 2, and 3.
```

```
SELECT * FROM edbreport(1, 3); // Generate a performance analysis report based on the snapshot 1 and snapshot
```

```
3.
```

- Session termination functions

```
rds_pg_terminate_backend(upid int)
```

```
rds_pg_cancel_backend(upid int)
```

The functions are similar to the native `pg_terminate_backend` and `pg_cancel_backend` functions. The functions provided by Alibaba Cloud cannot manage sessions established by the superuser account.

Example: The following statement terminates the session 123456:

```
select rds_pg_cancel_backend(123456);
```

- VPD functions

Virtual Private Database (VPD) is an encapsulation that is compatible with the `DBMS_RLS` package and uses the same parameters as `DBMS_RLS` uses.

1. `rds_drop_policy` is similar to `DBMS_RLS.DROP_POLICY`.

2. `rds_enable_policy` is similar to `DBMS_RLS.ENABLE_POLICY`.

3. `rds_add_policy` is similar to `DBMS_RLS.ADD_POLICY`.

References of VPD

12. ApsaraDB RDS for PostgreSQL

12.1. What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage of Alibaba Cloud, ApsaraDB for RDS allows you to easily perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS for PostgreSQL

ApsaraDB RDS for PostgreSQL is the most advanced open source database. It is fully compatible with SQL and supports a diverse range of data formats such as JSON, IP, and geometric data. In addition to features such as transactions, subqueries, multi-version concurrency control (MVCC), and data integrity check, ApsaraDB RDS for PostgreSQL integrates a series of features including high availability, backup, and restoration to ease operation and maintenance loads.

12.2. Limits

To ensure instance stability and security, ApsaraDB RDS for PostgreSQL has some service limits.

The following table lists the limits.

Operation	Limit
Root privilege of databases	Superuser permissions are not provided.
Database backup	Data can only be backed up by using <code>pg_dump</code> .
Database replication	ApsaraDB RDS for PostgreSQL provides a primary/secondary replication architecture (except in the Basic Edition). The secondary instance in the architecture is hidden and cannot be accessed by your applications.
RDS instance restart	You must restart an ApsaraDB RDS for PostgreSQL instance in the ApsaraDB for RDS console or by using OpenAPI Explorer.

12.3. Log on to the ApsaraDB for RDS console

This topic describes how to log on to the ApsaraDB for RDS console.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Database Services > ApsaraDB for RDS**.

12.4. Quick Start

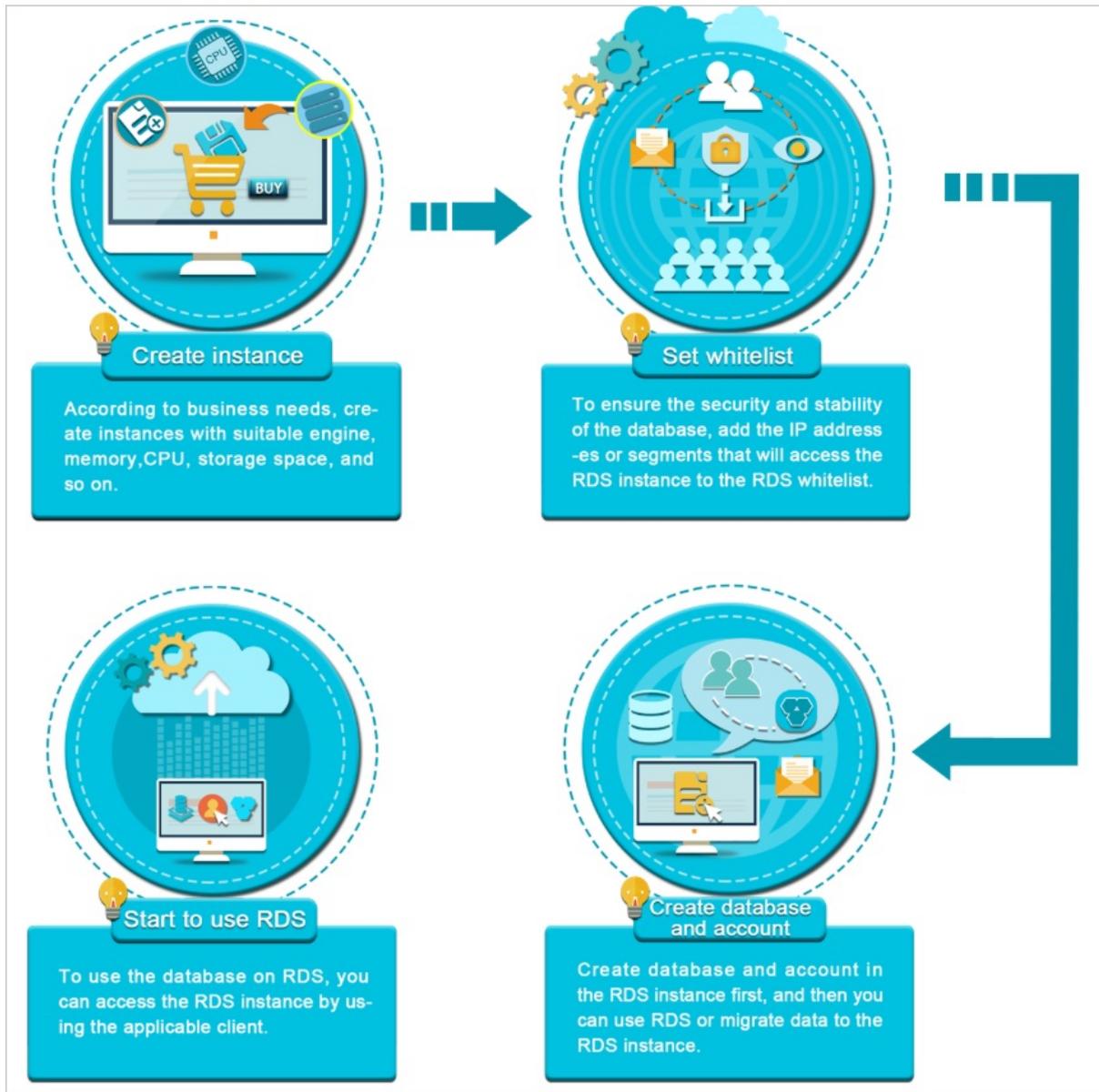
12.4.1. Procedure

This topic describes how to create and use an ApsaraDB RDS for PostgreSQL instance. The following topics describe how to create an instance, configure basic settings, and connect to an instance in details.

Flowchart for an ApsaraDB RDS for PostgreSQL instance

If you are using ApsaraDB for RDS for the first time, you can start with [Limits](#).

To purchase an ApsaraDB RDS for PostgreSQL instance and use the instance, follow these steps:



12.4.2. Create an instance

This topic describes how to create an ApsaraDB RDS for PostgreSQL instance in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides.
	Zone	The zone where the instance resides.

Region Section	Parameter	Description
Specifications	Instance Name	The name of the instance. <ul style="list-style-type: none"> The name must be 2 to 64 characters in length. The name must start with a letter. The name can contain special characters. Special characters include _ - : The name cannot start with http:// or https://.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page. Select PostgreSQL.
	Engine Version	The version of the database engine. Valid values: 9.4 and 10.0.
	Edition	The edition of the database. Select one from the drop-down list.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed on the console.
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files. The minimum storage capacity is 20 GB. You can adjust the storage capacity.
Network Type	Network Type	The network type of the instance. RDS instances support the following network types: <ul style="list-style-type: none"> Classic Network: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. VPC: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize the route table, IP address range, and gateway within a VPC. We recommend that you select VPC for improved security. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note After you select VPC as the network type, you must also select the corresponding VPC and VSwitch. </div>
	IP Whitelist	An IP address whitelist contains the IP addresses of entities that are allowed to access to your RDS instance. For more information, see Configure an IP address whitelist .
Access Mode	Access Mode	RDS instances support two access modes: Standard and Database Proxy . <ul style="list-style-type: none"> Standard: RDS uses Server Load Balancer (SLB) to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception. Database Proxy: This mode prevents 90% of network interruptions and intercepts SQL injection attacks based on semantic analysis. However, it increases the response time by over 20%. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note ApsaraDB RDS for PostgreSQL supports the Standard access mode. </div>

4. After you configure the preceding parameters, click **Submit**.

12.4.3. Configure an IP address whitelist

This topic describes how to configure a whitelist for an ApsaraDB RDS for PostgreSQL instance. Only entities that are listed in a whitelist can access your RDS instance.

Context

Whitelists make your RDS instance more secure without interrupting the operation of your RDS instance during configuration. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- Configure a whitelist: Add IP addresses to allow them to connect to the RDS instance.

 **Note** The IP address whitelist labeled default contains only the default IP address 0.0.0.0/0, which allows all entities access to your RDS instance.

- Configure an ECS security group: Add an ECS security group for the RDS instance to allow ECS instances in the group to connect to the RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, select **Data Security** and click the **Whitelist Settings** tab on the page that appears.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

 **Note** You can also click **Create Whitelist** to create a new whitelist.

6. In the **Edit Whitelist** dialog box that appears, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**. The following section describes the rules:
 - If you enter the CIDR block 10.10.10.0/24 in the **IP Addresses** field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all created ECS instances within your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

12.4.4. Create a database and an account

Before you start to use ApsaraDB for RDS, you must create a database and an account for an ApsaraDB for RDS instance. This topic describes how to create a database and an account for an ApsaraDB RDS for PostgreSQL instance.

Create a privileged account

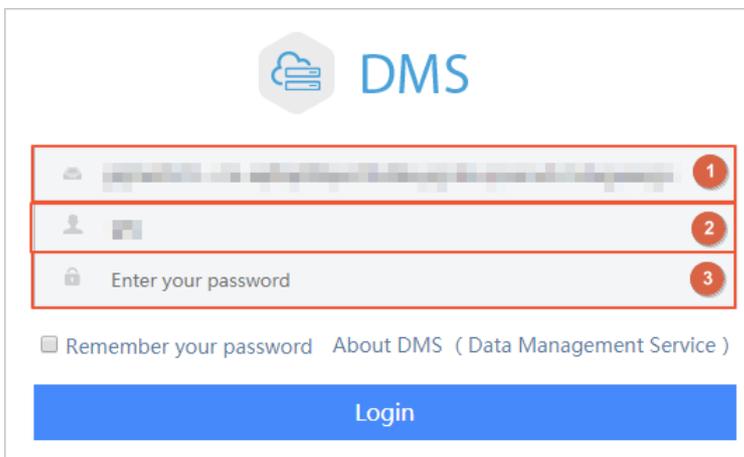
1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the **Accounts** page that appears, click **Create Initial Account** and configure the following parameters.

Parameter	Description
Database Account	<ul style="list-style-type: none"> The name of the account must be 2 to 16 characters in length. The name can contain lowercase letters, digits, and underscores (_). The name must start with a letter and end with a letter or digit.
Password	<ul style="list-style-type: none"> The password of the account must be 8 to 32 characters in length. The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the same password again.

6. Click **Create**.

Create a database and a standard account

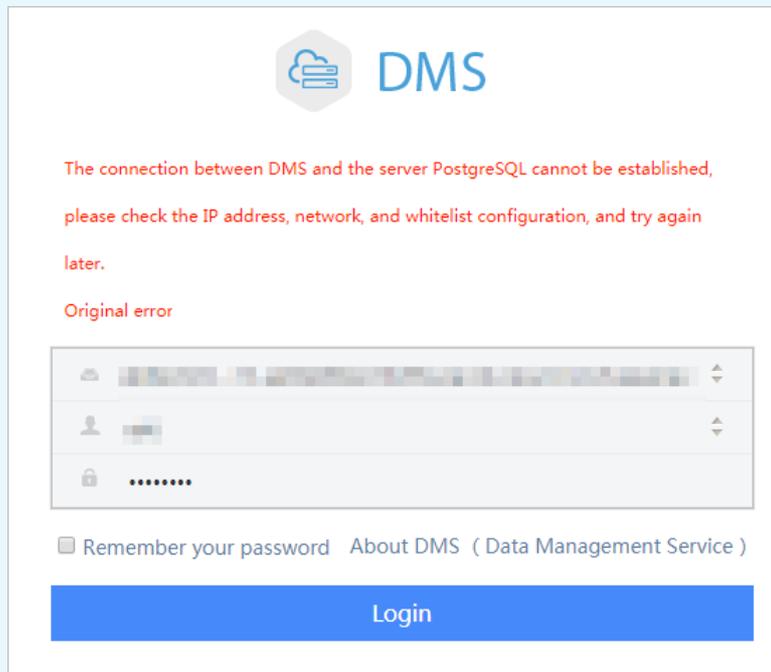
1. Log on to the **ApsaraDB for RDS console**.
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS)** logon page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.



Parameter	Description
IP address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the username of the account that is used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account that is used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP address whitelist](#).



7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
8. In the SQL window, execute the following statement to create a database:

```
CREATE DATABASE name
[ [ WITH ] [ OWNER [=] user_name ]
  [ TEMPLATE [=] template ]
  [ ENCODING [=] encoding ]
  [ LC_COLLATE [=] lc_collate ]
  [ LC_CTYPE [=] lc_ctype ]
  [ TABLESPACE [=] tablespace_name ]
  [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

9. Click **execute**.
10. In the SQL window, execute the following statement to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
    SUPERUSER | NOSUPERUSER
    | CREATEDB | NOCREATEDB
    | CREATEROLE | NOCREATEROLE
    | CREATEUSER | NOCREATEUSER
    | INHERIT | NOINHERIT
    | LOGIN | NOLOGIN
    | REPLICATION | NOREPLICATION
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | VALID UNTIL 'timestamp'
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

11. Click **execute**.

12.4.5. Connect to an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB for RDS instance.

Context

You can log on to DMS from the ApsaraDB for RDS console and then connect to an RDS instance.

Data Management (DMS) is an integrated database for data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, PostgreSQL, MongoDB, and Redis. It can also be used to manage Linux servers.

You can also use a database client to connect to an RDS instance. ApsaraDB RDS for PostgreSQL is fully compatible with PostgreSQL. You can connect to an ApsaraDB RDS for PostgreSQL instance in a similar manner as you would connect to an on-premises PostgreSQL instance. This topic describes how to use the pgAdmin 4 client to connect to an RDS instance.

Use DMS to connect to an RDS instance

For more information about how to connect an RDS instance through DMS, see [Log on to an ApsaraDB for RDS instance by using DMS](#).

Use the pgAdmin 4 client to connect to an RDS instance

1. Add the IP address that requires access to the RDS instance to a whitelist of the RDS instance. For more information about how to configure a whitelist, see [Configure an IP address whitelist](#).

2. Start the pgAdmin 4 client.

 **Note** For more information about how to download the pgAdmin 4 client, visit [pgAdmin 4 \(Windows\)](#).

3. Right-click **Servers** and choose **Create > Server**.
4. On the **General** tab of the **Create - Server** dialog box that appears, enter the name of the server.
5. Click the **Connection** tab and enter the information of the destination instance.

Parameter	Description
Host name/address	The internal endpoint of the RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number .
Port	The internal port number that is used to connect to the RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number .
Username	The name of the privileged account for the RDS instance. For more information about how to obtain a privileged account, see Create a database and an account .
Password	The password of the privileged account of the RDS instance.

6. Click **Save**.
7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**.

 **Notice** The postgres database is the default system database of the RDS instance. Do not perform operations on this database.

12.5. Instances

12.5.1. Create an instance

This topic describes how to create an ApsaraDB RDS for PostgreSQL instance in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, click **Create Instance** in the upper-right corner.
3. Configure the following parameters.

Section	Parameter	Description
Basic Settings	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
Region	Region	The region where the instance resides.
	Zone	The zone where the instance resides.

Section	Parameter	Description
Specifications	Instance Name	<p>The name of the instance.</p> <ul style="list-style-type: none"> The name must be 2 to 64 characters in length. The name must start with a letter. The name can contain special characters. Special characters include _ - : The name cannot start with http:// or https://.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Create ApsaraDB for RDS Instance page. Select PostgreSQL.
	Engine Version	The version of the database engine. Valid values: 9.4 and 10.0.
	Edition	The edition of the database. Select one from the drop-down list.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed on the console.
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files. The minimum storage capacity is 20 GB. You can adjust the storage capacity.
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> Classic Network: Cloud services on the classic network are not isolated. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. VPC: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize the route table, IP address range, and gateway within a VPC. We recommend that you select VPC for improved security. <p> Note After you select VPC as the network type, you must also select the corresponding VPC and VSwitch.</p>
	IP Whitelist	An IP address whitelist contains the IP addresses of entities that are allowed to access to your RDS instance. For more information, see Configure an IP address whitelist .
Access Mode	Access Mode	<p>RDS instances support two access modes: Standard and Database Proxy.</p> <ul style="list-style-type: none"> Standard: RDS uses Server Load Balancer (SLB) to eliminate the impact of instance high-availability switching on the application layer. This mode reduces the response time, but slightly increases the probability of network interruptions and disables SQL interception. Database Proxy: This mode prevents 90% of network interruptions and intercepts SQL injection attacks based on semantic analysis. However, it increases the response time by over 20%. <p> Note ApsaraDB RDS for PostgreSQL supports the Standard access mode.</p>

4. After you configure the preceding parameters, click **Submit**.

12.5.2. View basic information of an instance

This topic describes how to view the details of an ApsaraDB for RDS instance, such as basic information, internal network connection information, status, and configurations.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. Use one of the following methods to go to the **Basic Information** page of an instance:
 - On the **Instances** page, find the target instance and click the instance ID. The **Basic Information** page appears.
 - On the **Instances** page, find the target instance and click **Manage** in the corresponding **Actions** column. The **Basic Information** page appears.

12.5.3. Restart an instance

This topic describes how to manually restart an ApsaraDB RDS for PostgreSQL instance. This applies if the number of connections exceeds the specified threshold or if an instance has any performance issues.

Prerequisites

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

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Restart Instance**.

 **Note** A restart will disconnect the instance. We recommend that you make appropriate service arrangements before you restart an instance. Proceed with caution.

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

12.5.4. Change the specifications of an instance

This topic describes how to change specifications of your instance, such as the instance type and storage space, if the specifications do not meet the requirements of your application.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configure Information** section of the **Basic Information** page, click **Change Specifications**.
5. On the **Change Specifications** page, set **Edition**, **Instance Type**, and **Storage**.
6. After you configure the preceding parameters, click **Submit**.

12.5.5. Set a maintenance window

This topic describes how to set a maintenance window for an ApsaraDB for RDS instance.

Context

To ensure the stability of ApsaraDB for RDS instances, the backend system performs maintenance of the instances at irregular intervals. The default maintenance window is from 02:00 (UTC+8) to 06:00 (UTC+8). We recommend that you set the maintenance window to off-peak hours of your business to avoid impacts on your business.

Precautions

- To ensure the stability of the maintenance process, the instance will enter the **Maintaining Instance** state before the maintenance time. Access to data in the database and query operations such as performance monitoring are not affected while the instance is in this state. However, apart from account and database management and IP address whitelist configuration, modification operations such as upgrade, downgrade, and restart will be temporarily unavailable.
- Network interruptions may occur during a maintenance window. Make sure that your application is configured with automatic reconnection policies.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Configuration Information** section, click **Configure** to the right of **Maintenance Window**.
5. Select a maintenance window and click **Save**.

 **Note** The maintenance window is displayed in UTC+8.

12.5.6. Configure primary/secondary switchover

ApsaraDB for RDS provides the primary/secondary switchover feature to ensure the high availability of databases. The primary/secondary switchover is performed when the primary instance becomes unavailable. You can also manually switch your business to the secondary instance. This topic describes how to manually switch over services between a primary instance and its secondary instance.

Context

Data is synchronized in real time between the primary and secondary instances. You can access only the primary instance. The secondary instance serves only as a backup instance and does not allow external access. After the switchover, the original primary instance becomes the secondary instance.

 **Note** Network interruptions may occur during a switchover. Make sure that your application is configured with automatic reconnection policies.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.
6. In the **Switch Primary/Secondary Instance** message that appears, click **OK**.

Note

- During the switchover, operations such as managing databases and accounts and changing network types cannot be performed. Therefore, we recommend that you select Switch Within Maintenance Window.
- For more information about how to set a maintenance window, see [Set a maintenance window](#).

12.5.7. Release an instance

This topic describes how to manually release an instance.

Precautions

- Only instances in the running state can be released.
- After an instance is released, the instance data is immediately deleted. We recommend that you back up instance data before you release the instance.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. In the Actions column corresponding to the instance you want to release, click **More**, and select **Release Instance** from the drop-down list.
3. In the Release Instance message that appears, click **Confirm**.

12.5.8. Modify parameters of an instance

This topic describes how to view and modify the values of some parameters and query parameter modification records in the console.

Precautions

- To ensure instance stability, you can select specific parameters to modify in the ApsaraDB for RDS console.
- When you modify parameters on the **Editable Parameters** tab, refer to the **Value Range** column corresponding to each parameter.
- After certain parameters are modified, you must restart your RDS instance for the changes to take effect. For more information, see the **Force Restart** column on the **Editable Parameters** tab. We recommend that you modify the parameters of an instance during off-peak hours and make sure that your application is configured with automatic reconnection policies.

Modify parameters

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. You can perform the following operations: Export the parameter settings of the RDS instance to your computer.

On the **Editable Parameters** tab, click **Export Parameters**. The parameter settings of the RDS instance are exported as a TXT file to your computer.

Modify and import the parameter settings.

- i. After you have modified parameters in the exported parameter file, click **Import Parameters** and copy the parameter settings to the field.
- ii. Click **OK**.

- iii. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your application is configured with automatic reconnection policies.
- Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

Modify a single parameter.

- i. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the  icon in the **Actual Value** column.
- ii. Enter a new value based on the prompted value range.
- iii. Click **Confirm**.
- iv. In the upper-right corner of the page, click **Apply Changes**.

 **Note**

- If the new parameter value takes effect only after an instance restart, the system will prompt you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your application is configured with automatic reconnection policies.
- Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

View the parameter modification history

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Parameters**.
5. On the page that appears, click the **Edit History** tab.
6. Select a time range and then click **Search**.

12.5.9. Read-only instances

12.5.9.1. Overview of read-only ApsaraDB RDS for PostgreSQL

instances

This topic provides an overview of read-only ApsaraDB RDS for PostgreSQL instances. If a large number of read requests overwhelm the primary instance, your business may be interrupted. In this situation, you can create one or more read-only instances to offload read requests from the primary instance and increase the throughput of your application.

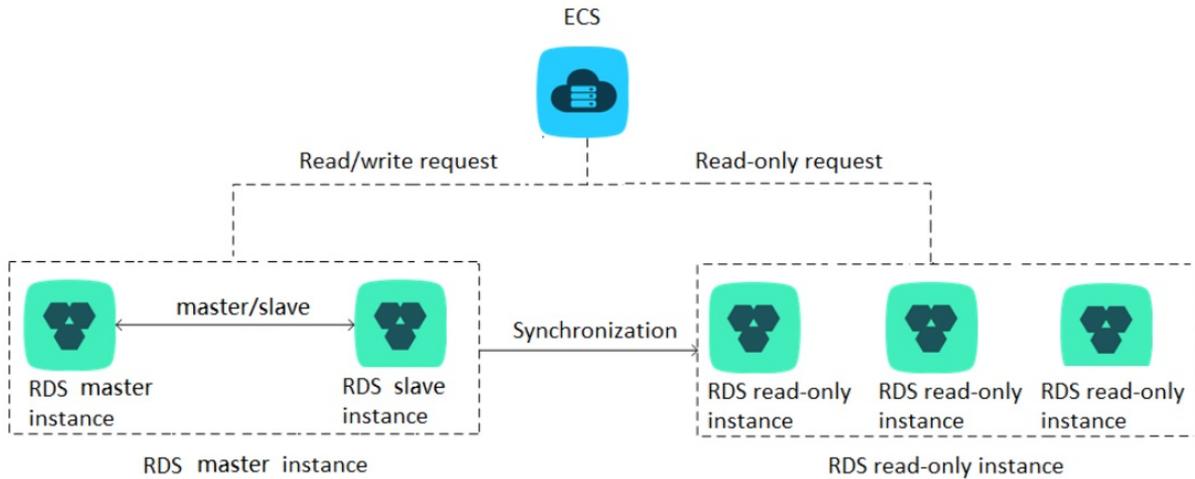
Overview

When a read-only instance is created, the data is replicated from the secondary instance. The data is consistent with that of the primary instance. Data updates of the primary instance are automatically synchronized to all read-only instances immediately after the primary instance completes operations.

Note

- Only ApsaraDB for RDS instances that run PostgreSQL 10.0 support read-only instances.
- The specifications of the primary instance must be at least eight CPU cores and 32 GB of memory.
- Each read-only instance works in a single-node architecture with no backup instances.

The following figure shows the topology of read-only instances.



Features

- **Region and zone:** Read-only instances reside within the same region as the primary instance, but possibly in different zones.
- **Specifications and storage space:** The specifications and storage space of read-only instances cannot be lower than that of the primary instance.
- **Network type:** The network type of a read-only instance can differ from that of the primary instance.
- **Account and database management:** Read-only instances do not require database or account maintenance, because their database and account information is synchronized with the primary instance.
- **Whitelist:** A read-only instance automatically replicates the whitelists of the primary instance. However, the whitelists for the read-only instance are independent of those of the primary instance. For information about how to modify the whitelists of a read-only instance, see [Configure an IP address whitelist](#).
- **Monitoring and alerts:** You can monitor system performance metrics, such as the disk capacity, IOPS, number of connections, and CPU utilization.

Limits

- **Number of read-only instances:** A maximum of five read-only instances can be created for a primary instance.
- **Instance backup:** Read-only instances do not support backup settings or manual backups because backups have been configured or created on the primary instance.
- **Data migration:** You cannot migrate data to read-only instances.
- **Database management:** You cannot create or delete databases on read-only instances.
- **Account management:** You cannot create or delete accounts, authorize accounts, or change the passwords of accounts on read-only instances.

FAQ

Q: Can I manage accounts created in the primary instance from its read-only instances?

A: No, although accounts created on the primary instance are replicated to its read-only instances, you cannot manage the accounts on the read-only instances. The accounts have only read permissions on the read-only instances.

12.5.9.2. Create a read-only ApsaraDB RDS for PostgreSQL instance

This topic describes how to create a read-only instance for your primary ApsaraDB RDS for PostgreSQL instance. This allows your database system to process a large number of read requests and increases the throughput of your application. The data on each read-only instance is a copy of that of the primary instance. Data updates to the primary instance are synchronized to all of its read-only instances.

Prerequisites

- The primary ApsaraDB for RDS instance runs PostgreSQL 10.0.
- The specifications of the primary instance must be at least eight CPU cores and 32 GB of memory.

Precautions

- You can only create read-only instances for your primary instance. You cannot change existing instances to read-only instances.
- When you create a read-only instance, the system replicates data from a secondary instance. Therefore, operations on your primary instance are not interrupted.
- A read-only instance does not inherit the parameter settings of the primary instance. The system generates default parameter settings for the read-only instance, and you can modify the settings in the ApsaraDB for RDS console.
- The specifications and storage space of a read-only instance cannot be lower than that of the primary instance.
- You can create up to five read-only instances.

Create a read-only instance

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Distributed by Instance Role** section of the **Basic Information** page, click **Create Read-only Instance**.
5. On the **Create Read-only RDS Instance** page that appears, set the parameters of the read-only instance according to the following table, and click **Submit**.

Section	Parameter	Description
Region	Region	The region where the ApsaraDB for RDS instance resides.
Specifications	Database Engine	The database engine of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	Engine Version	The engine version of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	Edition	The edition of the read-only instance, which is the same as that of the primary instance and cannot be modified.

Section	Parameter	Description
	Instance Type	The instance type of the read-only instance. The type of the read-only instance can be different from that of the primary instance, and can be modified at any time to facilitate flexible upgrade and downgrade. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same memory as the primary instance for the read-only instance.
	Storage	The storage space of the read-only instance. To ensure sufficient I/O throughput for data synchronization, we recommend that you select at least the same instance type and storage space as the primary instance for the read-only instance.
Network Type	Network Type	The network type of the read-only instance, which is the same as that of the primary instance and cannot be modified.
	VPC	Select a VPC if the network type is set to VPC.
	VSwitch	Select a VSwitch if the network type is set to VPC.

12.5.9.3. View a read-only ApsaraDB RDS for PostgreSQL instance

This topic describes how to view details of a read-only ApsaraDB RDS for PostgreSQL instance. You can go to the Basic Information page of a read-only instance from the Instances page or the read-only instance list of the primary instance. Read-only instances are managed in the same manner as primary instances. The Basic Information page shows the management operations that can be performed.

View instance details of a read-only instance through its ID

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the Instances page, find the target instance.
3. Click the ID of the target instance or click **Manage** in the corresponding Actions column to go to the Basic Information page.

View details of a read-only instance through the primary instance

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the Basic Information page.
4. On the Basic Information page, move the pointer over the number below **Read-only Instance** in the **Distributed by Instance Role** section. The ID of the read-only instance is displayed.
5. Click the ID of the read-only instance to go to the Basic Information page of the read-only instance.

View the latency of a read-only instance

When a read-only instance synchronizes data from its primary instance, latency may occur. You can navigate to the Basic Information page of a read-only instance to view the latency of data synchronization to the instance.

Delay for Read-only Instance			
Delay for Sending Write-Ahead Logging Data: 0MB	Delay for Writing Write-Ahead Logging Data: 0MB	Delay for Syncing Write-Ahead Logging Data: 0MB	Delay for Applying Write-Ahead Logging Data: 0MB
	Delay for Writing Write-Ahead Logging Data: 0.000103Second	Delay for Syncing Write-Ahead Logging Data: 0.000152Second	Delay for Applying Write-Ahead Logging Data: 0.0002Second

12.6. Database connection

12.6.1. Connect to an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use Data Management (DMS) or the pgAdmin 4 client to connect to an ApsaraDB for RDS instance.

Context

You can log on to DMS from the ApsaraDB for RDS console and then connect to an RDS instance.

Data Management (DMS) is an integrated database for data, schema, and server management, access security, BI charts, data trends, data tracking, and performance optimization. DMS can be used to manage relational and non-relational databases, such as MySQL, SQL Server, PostgreSQL, MongoDB, and Redis. It can also be used to manage Linux servers.

You can also use a database client to connect to an RDS instance. ApsaraDB RDS for PostgreSQL is fully compatible with PostgreSQL. You can connect to an ApsaraDB RDS for PostgreSQL instance in a similar manner as you would connect to an on-premises PostgreSQL instance. This topic describes how to use the pgAdmin 4 client to connect to an RDS instance.

Use DMS to connect to an RDS instance

For more information about how to connect an RDS instance through DMS, see [Log on to an ApsaraDB for RDS instance by using DMS](#).

Use the pgAdmin 4 client to connect to an RDS instance

1. Add the IP address that requires access to the RDS instance to a whitelist of the RDS instance. For more information about how to configure a whitelist, see [Configure an IP address whitelist](#).
2. Start the pgAdmin 4 client.

Note For more information about how to download the pgAdmin 4 client, visit [pgAdmin 4 \(Windows\)](#).

3. Right-click **Servers** and choose **Create > Server**.
4. On the **General** tab of the **Create - Server** dialog box that appears, enter the name of the server.
5. Click the **Connection** tab and enter the information of the destination instance.

Parameter	Description
Host name/address	The internal endpoint of the RDS instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number .
Port	The internal port number that is used to connect to the RDS instance. For more information about how to view the internal port number, see View and modify the internal endpoint and port number .
Username	The name of the privileged account for the RDS instance. For more information about how to obtain a privileged account, see Create a database and an account .
Password	The password of the privileged account of the RDS instance.

6. Click **Save**.

7. If the connection information is correct, choose **Servers > Server Name > Databases > postgres**.

 **Notice** The postgres database is the default system database of the RDS instance. Do not perform operations on this database.

12.6.2. Log on to an ApsaraDB for RDS instance by using DMS

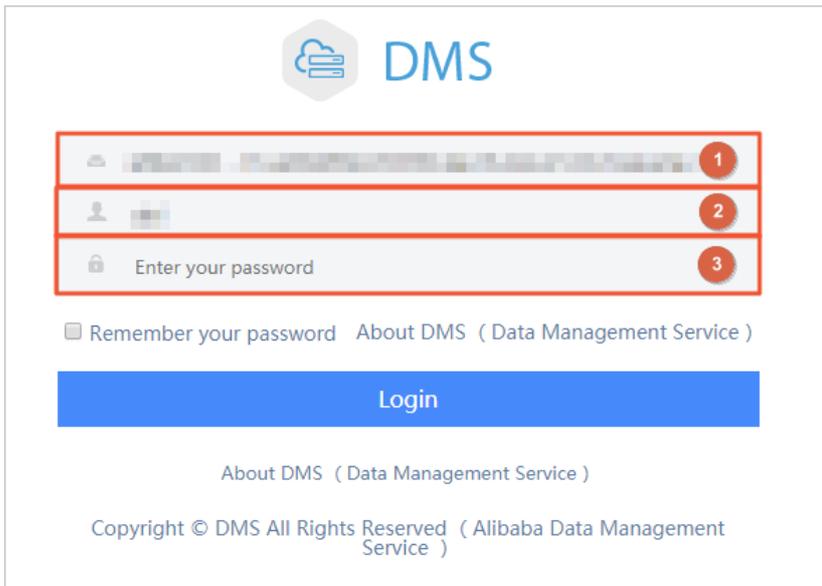
This topic describes how to log on to an ApsaraDB for RDS instance by using Data Management (DMS).

Prerequisites

The IP address whitelist is configured. For more information about how to configure an IP address whitelist, see [Configure an IP address whitelist](#).

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to go to the RDS Database Logon page.
5. On the logon page, set the following parameters:



- ①: The endpoint and port number that are used to connect to your RDS instance. The endpoint and port number are in the `<Internal endpoint>:<Internal port number>` format. Example: `rm-bpxxxxxx.rds.aliyuncs.com:3433`. For more information about how to view the internal endpoint and port number of an instance, see [View and modify the internal endpoint and port number](#).
 - ②: The account that is used to access the RDS database.
 - ③: The password of the account that is used to access the RDS database.
6. Click **Login**.

Note If you want the browser to remember the password, select **Remember your password** and click **Login**.

12.6.3. View and modify the internal endpoint and port number

You must use the internal endpoint and port number to access an RDS instance. This topic describes how to view and modify the internal endpoint and port number of an ApsaraDB for RDS instance in the ApsaraDB for RDS console.

View the internal endpoint and port number

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the **Basic Information** section, view the internal endpoint and internal port number of the instance.

Modify the internal endpoint and port number

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the right side of the page, click **Change Endpoint**.

6. In the dialog box that appears, set **Connection Type** to **Internal Endpoint**.
7. Modify the endpoint prefix and port number and then click **OK**.

FAQ

- **Q:** Do I need to modify the endpoint or port number in my application after I modify the endpoint or port number of an instance?
A: Yes, you must modify the endpoint or port number in the application after you have modified them. Otherwise, the application cannot connect to databases of the instance.
- **Q:** Does the modification of the endpoint take effect immediately? Do I need to restart the instance?
A: No, you do not need to restart the instance. The modification takes effect immediately.

12.7. Accounts

12.7.1. Create an account

Before you start to use ApsaraDB for RDS, you must create an account for an RDS instance. This topic describes how to create an account for an ApsaraDB RDS for PostgreSQL instance.

Precautions

- Databases within the same instance share all resources that belong to the instance. You can create a privileged account and multiple standard accounts for each ApsaraDB RDS for PostgreSQL instance. You can also use SQL statements to create and manage accounts.
- To migrate data from an on-premises database to an RDS instance, you must create databases and accounts on the RDS instance. Make sure that each database or account on the on-premises instance has a counterpart with an identical name on the RDS instance.
- Follow the least privilege principle to create accounts and grant them appropriate read-only and read/write permissions on databases. When necessary, you can create multiple accounts and grant them only the permissions on specific databases. If an account does not need to write data to a database, grant only the read-only permissions on the database to the account.
- To ensure database security, set strong passwords for accounts and change the passwords on a regular basis.
- A privileged account cannot be deleted after it is created.

Create a privileged account

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. On the **Accounts** page that appears, click **Create Initial Account** and configure the following parameters.

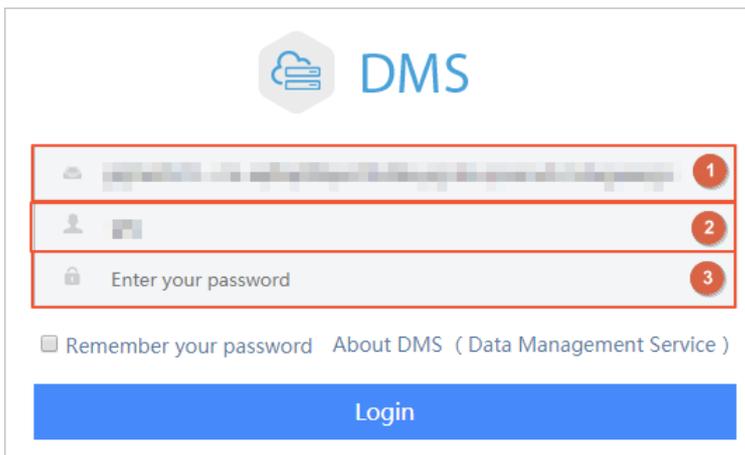
Parameter	Description
Database Account	<ul style="list-style-type: none"> ◦ The database account must be 2 to 16 characters in length. ◦ The account can contain lowercase letters, digits, and underscores (_). ◦ The account must start with a letter and end with a letter or digit.

Parameter	Description
Password	<ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length. ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! @ # \$ % ^ & * () _ + - =
Re-enter Password	Enter the same password again.

6. Click **Create**.

Create a standard account

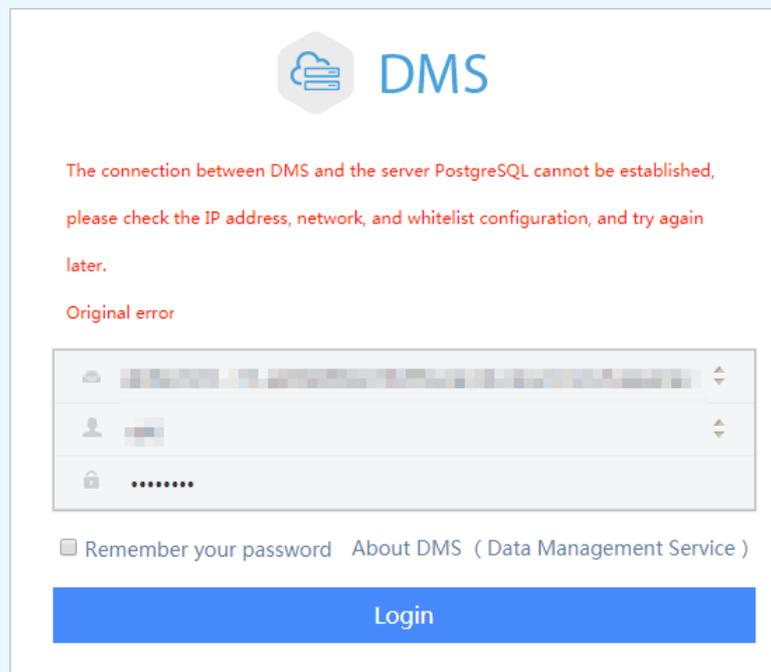
1. **Log on to the ApsaraDB for RDS console.**
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS)** logon page that appears, check the endpoint and port number displayed on the page. If the information is correct, enter the username and password of the database.



Parameter	Description
IP address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance.
Database Username	This parameter is marked with ② in the figure. Enter the username of the account that is used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account that is used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP address whitelist](#).



7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
8. In the SQL window, execute the following statement to create a standard account:

```
CREATE USER name [ [ WITH ] option [ ... ] ]
where option can be:
  SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | CREATEUSER | NOCREATEUSER
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

For example, if you want to create a user account named test2 whose password is 123456, execute the following statement:

```
create user test2 password '123456';
```

9. Click execute.

12.7.2. Reset the password

This topic describes how to use the RDS console to reset the password of your database account if you forget the password.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Accounts**.
5. In the **Actions** column corresponding to the target account, click **Reset Password**.
6. In the dialog box that appears, enter a new password, and click **OK**.

-  **Note** The password must meet the following requirements:
- The password must be 8 to 32 characters in length.
 - The password must contain at least three of the following characters: uppercase letters, lowercase letters, digits, and special characters.
 - Special characters include ! @ # \$ % ^ & * () _ + - =

12.8. Databases

12.8.1. Create a database

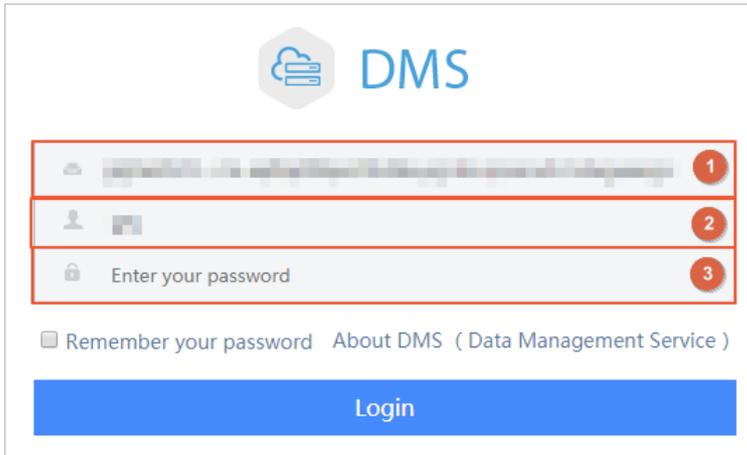
Before you start to use ApsaraDB for RDS, you must create a database and account for an RDS instance. This topic describes how to create a database for an ApsaraDB RDS for PostgreSQL instance.

Prerequisites

- An ApsaraDB RDS for PostgreSQL instance is created. For more information, see [Connect to an ApsaraDB RDS for PostgreSQL instance](#).
- An account is created. For more information, see [Create an account](#).

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the **Data Management (DMS) logon** page that appears, check the endpoint and port number displayed on the page.



Parameter	Description
IP address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance. For more information, see View and modify the internal endpoint and port number .
Database Username	This parameter is marked with ② in the figure. Enter the username of the account that is used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account that is used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP address whitelist](#).

7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.

8. In the SQL window, execute the following statement to create a database:

```
CREATE DATABASE name
[ [ WITH ] [ OWNER [=] user_name ]
  [ TEMPLATE [=] template ]
  [ ENCODING [=] encoding ]
  [ LC_COLLATE [=] lc_collate ]
  [ LC_CTYPE [=] lc_ctype ]
  [ TABLESPACE [=] tablespace_name ]
  [ CONNECTION LIMIT [=] connlimit ] ]
```

For example, if you want to create a database named test, execute the following statement:

```
create database test;
```

9. Click **execute**.

12.8.2. Delete a database

This topic describes how to delete a database in the ApsaraDB RDS for PostgreSQL console.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. On the Instances page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. Click **Log On to DB** in the upper-right corner of the page.
5. On the Data Management (DMS) logon page that appears, check the endpoint and port number displayed on the page.

Parameter	Description
IP address:Port	This parameter is marked with ① in the figure. Enter the endpoint and port number of the instance. For more information, see View and modify the internal endpoint and port number .
Database Username	This parameter is marked with ② in the figure. Enter the username of the account that is used to access the database.
Enter your password	This parameter is marked with ③ in the figure. Enter the password of the account that is used to access the database.

6. Click **Login**.

Note If a message prompts you that the connection fails, the problem may be caused by an improperly configured whitelist. Reconfigure the whitelist in the console. For more information, see [Configure an IP address whitelist](#).

7. In the top navigation bar, choose **SQL Operations > SQL Window** after you have logged on to the RDS instance.
8. Execute the following statement to delete the database:

```
drop database <database name>;
```

9. Click **execute**.

12.9. Network, VPC, and VSwitch

12.9.1. Change the network type of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to change the network type of an ApsaraDB RDS for PostgreSQL instance between classic network and VPC.

Context

- **Classic network:** RDS instances in the classic network are not isolated. Unauthorized access to these instances can be blocked only by whitelists.
- **VPC:** Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you use the VPC network type because it provides a higher security level.

You can configure route tables, CIDR blocks, and gateways in a VPC. To smoothly migrate applications to the cloud, you can use leased lines or VPNs to connect on-premises data center to a VPC to create a virtual data center.

Change the network type from VPC to classic network

Precautions

- After you change the network type from VPC to classic network, the IP address bound to the internal endpoint will change but the internal endpoint of the RDS instance will remain unchanged.
- After the network type is changed, ECS instances in the same VPC as the RDS instance will no longer be able to connect to the RDS instance by using the internal endpoint. You must update the endpoint for the applications deployed on the ECS instances.
- When you change the network type, a 30-second network interruption may occur. To avoid business interruption, change the network type during off-peak hours or make sure that your application is configured with automatic reconnection policies.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the **Database Connection** section, click **Switch to Classic Network**.
6. In the dialog box that appears, click **OK**.

 **Note** After the network type is changed to classic network, only ECS instances within the classic network can connect to the RDS instance by using the internal endpoint. You must configure the internal endpoint for the ECS instances.

7. Configure a whitelist to allow ECS instances within the classic network to connect to the RDS instance by using the internal endpoint.

 **Note**

- If the network isolation mode of the RDS instance is standard whitelist mode, add the private IP addresses of the ECS instances to a whitelist of your RDS instance.
- If the network isolation mode of the RDS instance is enhanced whitelist mode, add the internal IP addresses of the ECS instances to a classic network whitelist. If no classic network whitelists are available, create a whitelist. For more information about the enhanced whitelist mode, see [Switch to the enhanced whitelist mode](#).

Change the network type from classic network to VPC

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.

4. In the left-side navigation pane, click **Database Connection**.
5. In the upper-right corner of the Database Connection section, click **Switch to VPC**.
6. In the Switch to VPC dialog box, select a VPC and VSwitch and specify whether to retain the endpoint used in the classic network.

 **Note**

- Select a VPC. We recommend that you select the VPC where your ECS instances reside. Otherwise, the ECS instances cannot communicate with the RDS instance over the internal network.
- Select a VSwitch. If no VSwitches are available in the selected VPC, create one in the same zone where the RDS instance resides. Use the following path to navigate the guide: *Virtual Private Cloud > User Guide > Quick Start > Create a VSwitch*.
- Determine whether to select the **Reserve Original Classic Endpoint** option. The following table describes the details.

Operation	Description
Not selected	<p>The endpoint used in the classic network is replaced with an endpoint in the VPC.</p> <p>When you change the network type, a 30-second network interruption may occur, and connections between ECS instances in the classic network and the RDS instance are interrupted.</p>
Selected	<p>The endpoint used in the classic network is retained, and a new endpoint to be used in the VPC is generated. In such cases, the RDS instance runs in hybrid access mode. ECS instances in both the classic network and a VPC can connect to the RDS instance over the internal network. For more information, see Hybrid network access mode.</p> <p>When you change the network type, no network interruptions occur. Connections between ECS instances in the classic network and the RDS instance will be available until the endpoint used in the classic network expires.</p> <p>To migrate your business to the VPC without interruption, you must add the new endpoint used in the VPC to access the ECS instances before the endpoint used in the classic network expires. Seven days before the endpoint used in the classic network expires, the system will send a text message to the phone number bound to your Alibaba Cloud account every day.</p> <p>For more information, see Hybrid access from both the classic network and VPCs.</p>

7. Add the internal IP addresses of ECS instances in the selected VPC to a VPC whitelist. This allows the ECS instances to access the RDS instance over the internal network. If no VPC whitelists are available, create a whitelist.

 **Note**

- If you have retained the classic network endpoint, add the VPC endpoint to the ECS instances before the classic network endpoint expires.
- If you have not retained the classic network endpoint, connections between ECS instances in the classic network and the RDS instance over the internal network are interrupted. You must add the new endpoint to ECS instances in the VPC immediately after the network type is changed.

12.9.2. Hybrid access from both the classic network and VPCs

This topic describes how to use the hybrid access solution of ApsaraDB for RDS to change the network type of an instance from classic network to Virtual Private Network (VPC) without network interruptions.

Prerequisites

- The network type of the RDS instance is classic network.
- Available VPCs and VSwitches exist in the zone where the RDS instance resides.

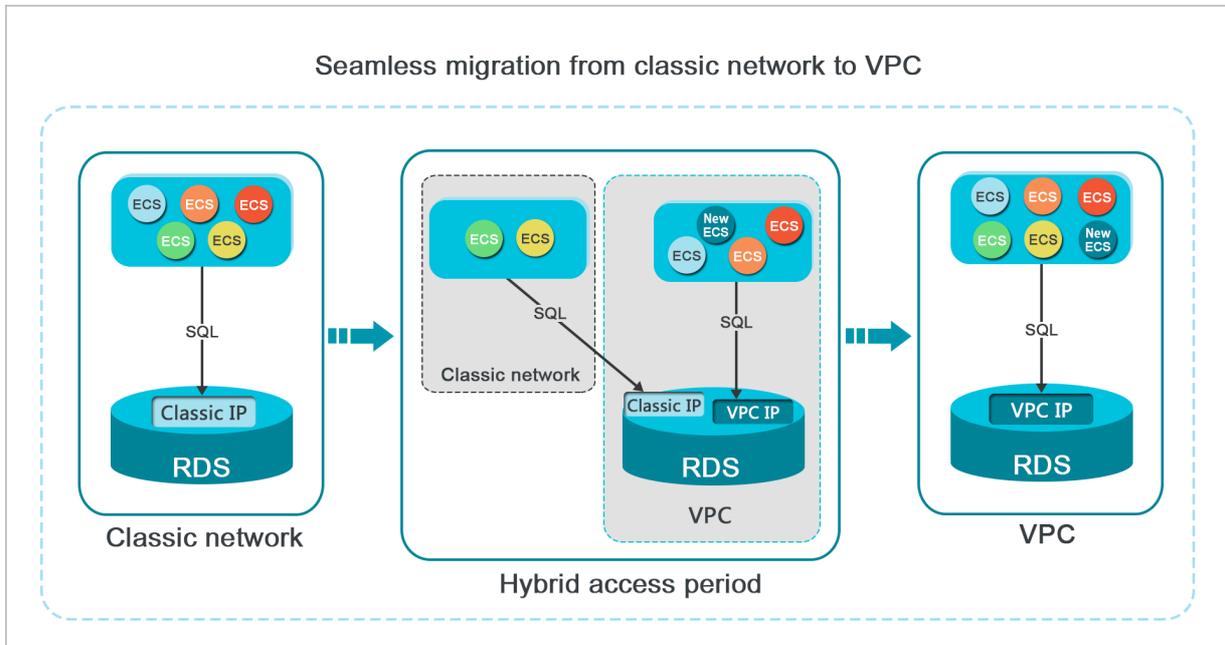
Context

In the past, when you changed the network type of an RDS instance from classic network to VPC, the internal endpoint of the RDS instance would remain the same but the IP address bound to the endpoint would change to the corresponding IP address in the VPC. This change would cause a 30-second network interruption, and ECS instances within the classic network would not be able to access the RDS instance through the internal endpoint within this period. To smoothly change the network type, ApsaraDB for RDS provides the hybrid access solution.

Hybrid access refers to the ability of an RDS instance to be accessed by ECS instances in both the classic network and VPCs. During the hybrid access period, the RDS instance reserves the original internal endpoint of the classic network and adds the internal endpoint of VPCs. This prevents network interruptions during the network type switchover.

For better security and performance, we recommend that you use the internal endpoint of VPCs. Hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPCs in all your applications during the hybrid access period. This ensures smooth network switchover and minimize the impact on your services.

For example, your company wants to use the hybrid access solution to change the network type from classic network to VPC. During the hybrid access period, some applications can access the database through the internal endpoint of VPCs, and the other applications can access the database through the original internal endpoint of the classic network. When all the applications access the database through the internal endpoint of VPCs, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.



Limits

During the hybrid access period, the instance has the following limits:

- Changing to the classic network is not supported.

- Migrating the RDS instance to another zone is not supported.

Change the network type from classic network to VPC

For more information, see [Change the network type from classic network to VPC](#).

Change the expiration time for the original internal endpoint of the classic network

During the period in which your instance can be connected over the classic network or VPCs, you can specify the expiration time for the endpoint of the classic network. The setting takes effect immediately. For example, if the endpoint of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the endpoint of the classic network is released on August 29, 2017.

Follow these steps to change the expiration time:

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Instance Connection** tab, click **Change Expiration Time**.
6. On the **Change Expiration Time** dialog box that appears, select an expiration time and click **OK**.

12.10. Monitoring

12.10.1. View monitored resources

ApsaraDB for RDS provides a wide range of performance metrics. This topic describes how to view resource monitoring data in the ApsaraDB for RDS console.

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, select the time range to query the corresponding monitoring data. The following table lists the specific monitoring metrics.

Monitoring metric	Description
Disk Space	The used disk space of the instance. Unit: MB.
IOPS	The number of I/O requests of the data disk and log disk per second.
Memory Usage	The memory usage of the instance.
CPU Utilization	The CPU utilization of the instance.
Total Connections	The total number of current connections of the instance.

 **Note** You can click **Refresh** in the upper-right corner of the **Monitoring** tab to refresh the monitoring information.

12.10.2. Set the monitoring frequency

This topic describes how to set the monitoring frequency of an ApsaraDB RDS for PostgreSQL instance.

Context

ApsaraDB RDS for PostgreSQL provides three monitoring frequencies.

- Every 5 seconds
- Every 60 seconds
- Every 300 seconds

Procedure

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, click **Set Monitoring Frequency**.
6. In the **Set Monitoring Frequency** dialog box that appears, select a monitoring frequency and click **OK**.

12.11. Data security and encryption

12.11.1. Switch to the enhanced whitelist mode

This topic describes how to switch from the standard whitelist mode to the enhanced whitelist mode for an ApsaraDB RDS for PostgreSQL instance. The enhanced whitelist mode offers higher security.

Network isolation modes

RDS instances support the following network isolation modes:

- **Standard whitelist mode**
IP addresses from both the classic network and VPCs are added to the same whitelist. However, standard whitelist mode may incur security risks. We recommend that you switch the network isolation mode to enhanced whitelist.
- **Enhanced whitelist mode**
IP addresses from the classic network and VPCs are added to different whitelists. When you create an enhanced IP address whitelist, you must specify its network type.

Changes after you switch to the enhanced whitelist mode

- If the network type of the instance is VPC, a new whitelist is created and contains the same IP addresses as the original whitelists. The new whitelist only applies to VPCs.
- If the network type of the instance is classic network, a new whitelist is created and contains the same IP addresses as the original whitelists. The new IP whitelist only applies to the classic network.
- If the instance supports [access from both the classic network and VPCs](#), two new whitelists are created, and each contains the same IP addresses as the original whitelists. One whitelist applies to VPCs, and the other applies to the classic network.

 **Note** Switching to the enhanced whitelist mode does not affect ECS security groups in the whitelist.

Precautions

- You can switch from standard whitelist mode to enhanced whitelist mode, but not the other way around.
- In the enhanced whitelist mode, a classic network whitelist also allows access from the Internet. If you want to access the RDS instance from a host over the Internet, you can add the public IP address of the host to a

classic network whitelist.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Enable Enhanced Whitelist (Recommended)**.
6. In the Enable Enhanced Whitelist message, click **Confirm**.

12.11.2. Configure an IP address whitelist

This topic describes how to configure a whitelist for an ApsaraDB RDS for PostgreSQL instance. Only entities that are listed in a whitelist can access your RDS instance.

Context

Whitelists make your RDS instance more secure without interrupting the operation of your RDS instance during configuration. We recommend that you perform maintenance on your whitelists on a regular basis.

To configure a whitelist, perform the following operations:

- **Configure a whitelist:** Add IP addresses to allow them to connect to the RDS instance.

 **Note** The IP address whitelist labeled default contains only the default IP address 0.0.0.0/0, which allows all entities access to your RDS instance.

- **Configure an ECS security group:** Add an ECS security group for the RDS instance to allow ECS instances in the group to connect to the RDS instance.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the Actions column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, select **Data Security** and click the **Whitelist Settings** tab on the page that appears.
5. On the **Whitelist Settings** tab, click **Edit** corresponding to the default whitelist.

 **Note** You can also click **Create Whitelist** to create a new whitelist.

6. In the **Edit Whitelist** dialog box that appears, enter the IP addresses or CIDR blocks used to access the instance, and then click **OK**. The following section describes the rules:
 - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
 - If you enter more than one IP address or CIDR block, you must separate them with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - If you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all created ECS instances within your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

12.12. Log and audit

12.12.1. SQL audit (database audit)

You can use the SQL audit feature to audit SQL executions and check the details. SQL audit does not affect instance performance.

Precautions

- SQL audit does not affect instance performance.
- SQL audit logs are retained for 30 days.
- Log files exported from SQL audit are retained for two days. The system deletes files that are retained for longer than two days.
- SQL audit is disabled by default. You must manually enable it.
- You cannot view logs that are generated before SQL audit is enabled.

Enable SQL audit

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab.
6. Click **Enable SQL audit** or **Enable now**.
7. In the message that appears, click **Confirm**.

 **Note** After enabling SQL audit, you can query SQL information based on conditions such as the time, database, user, and keyword.

Disable SQL audit

You can disable SQL audit when it is no longer needed. To disable SQL audit, follow these steps:

 **Notice** If SQL audit is disabled, all SQL audit logs are deleted. We recommend that you export and store audit logs locally before you disable SQL audit.

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Data Security**.
5. Click the **SQL Audit** tab, click **Export File**, and then store the exported file locally.
6. Click **Disable SQL Audit**.
7. In the message that appears, click **Confirm**.

12.12.2. Manage logs

You can view logs for errors, slow queries, and primary/secondary instance switching for ApsaraDB RDS for PostgreSQL instances in the ApsaraDB for RDS console or by executing SQL statements. These logs help you troubleshoot errors. This topic describes how to manage logs in the console.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)

2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Logs**.
5. On the **Logs** page that appears, click the **Error Logs**, **Slow Query Logs**, or **Primary/Secondary Switching Logs** tab, select a time range, and click **Search**.

Tab	Description
Error Logs	Records database running errors that occurred within the last month.
Slow Log Details	Records SQL statements within the last month that took longer than one second to execute. Duplicated SQL statements are removed.
Primary/Secondary Instance Switching Log	Records switchovers between the primary and secondary instances within the last month.

12.13. Backup

12.13.1. Back up an ApsaraDB RDS for PostgreSQL instance

This topic describes how to back up an ApsaraDB RDS for PostgreSQL instance. You can configure a backup policy that is used to automatically back up your RDS instance. If you do not configure a backup policy, the default backup policy is used. You can also manually back up your RDS instance.

Precautions

- Do not perform data definition language (DDL) operations during a backup. If you do so, the backup may fail due to table locks.
- We recommend that you back up your RDS instance during off-peak hours.
- If the amount of data is large, it may take a long time to back up your RDS instance.
- Backups are retained for a specified retention period. We recommend that you download the required backups to your computer before they are deleted.

Overview of data and log backups

Database engine	Data backup	Log backup
PostgreSQL	Supports full physical backup.	These are backups of the archived log files of your RDS instance.

Configure a backup policy to automatically back up your RDS instance

ApsaraDB for RDS automatically backs up your RDS instance based on the specified backup policy.

1. [Log on to the ApsaraDB for RDS console](#).
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. On the **Backup and Restoration** page, click the **Backup Settings** tab, and then click the **Edit** button.
6. In the dialog box that appears, configure the following parameters, and then click **OK**. The following table lists the parameters.

Parameter	Description
Data Retention Period	The number of days for which you want to retain data backup files. Valid values: 7 to 730. Unit: days. Default value: 7.
Backup Cycle	The cycle to create backups. You can select one or more days of the week. <div style="border: 1px solid #add8e6; padding: 5px;"> ? Note To ensure data security, we recommend that you back up your RDS instance at least twice a week. </div>
Backup Time	The hour at which you want to create a backup.
Log Backup	The switch to enable or disable the log backup function. <div style="border: 1px solid #add8e6; padding: 5px;"> 🔊 Notice If you disable this function, all log backup files are deleted and your RDS instance will not be able to restored to previous points in time. </div>
Log Retention Period	<ul style="list-style-type: none"> ◦ The period of time for which you want to retain log backup files. Valid values: 7 to 730. Unit: days. Default value: 7. ◦ The log retention period must be less than or equal to the data retention period.

Manually back up your RDS instance

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the upper-right corner of the page, click **Back Up Instance** to go to the **Back Up Instance** dialog box.
5. Select the backup mode and backup policy, and click **OK**.

? **Note** The backup mode is **Full Backup** and the backup policy is **Instance Backup**.

What's next

You can click the  icon in the upper-right corner of the page to view the task progress displayed in the **Task Progress** list.

12.13.2. Download data and log backup files

This topic describes how to download unencrypted data and log backup files in the ApsaraDB for RDS console to archive the files and restore data to an on-premises database.

Procedure

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration** to go to the **Backup and Restoration** page.
5. Click the **Data Backup** or **Archived Logs** tab.
 - To download data backup files, click the **Data Backup** tab.

- To download log backup files, click the **Archived Logs** tab.
- 6. Select a time range to which you want to restore the instance.
- 7. Find the data backup or log file that you want to download, and click **Download** in the **Actions** column.

Note

- If you want to use a data backup file to restore data, select the backup file that is the closest to the time for restoration.
- If you want to use a log file to restore data to an on-premises database, note the following items:
 - The instance No. of the log file must be the same as that of the data backup file.
 - The start time of the log file must be later than the data backup time and earlier than the time when you want to restore data.

- 8. In the download message that appears, select a download method.

Download method	Description
Download	Download the file by using the public endpoint.
Copy Internal Endpoint	Copy the internal endpoint to download the file. If your ECS and RDS instances reside within the same region, you can log on to the ECS instance and use the internal endpoint to download the file. This method is fast and secure.
Copy Public Endpoint	Copy the public endpoint to download the file. If you want to use other tools to download the file, use the public endpoint.

Note If you use a Linux operating system, you can run the following command to download the file:

```
wget -c '<Public endpoint of the backup file, which is the download URL>' -O <File name>
```

- The `-c` option enables resumable download.
- The `-O` option saves the downloaded file by using a specified name. We recommend that you use the file name contained in the download URL.
- If the URL contains more than one parameter, enclose the download URL in a pair of single quotation marks (').

```
root@izbp:~# wget -c 'http://rdslog-hz-...-cn-hangzhou.aliyuncs.com/.../hostins=...mysql-bin.000457' -O mysql-bin.000457
```

12.13.3. Create a logical backup for an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use `pg_dump` to create a logical backup for an ApsaraDB RDS for PostgreSQL instance and export the backup file to your computer.

Context

The `pg_dump` utility provided with PostgreSQL is used to back up individual databases. For more information, visit [pg_dump](#).

In this example, an ApsaraDB RDS for PostgreSQL instance that runs Linux 7 and PostgreSQL 10 is used.

Prerequisites

- The IP address of your ECS instance or on-premises host is added to a whitelist of your ApsaraDB RDS for PostgreSQL instance. For more information, see [Configure an IP address whitelist](#).
- Your ECS instance or on-premises host runs the same version of PostgreSQL as your ApsaraDB RDS for PostgreSQL instance.

Precautions

We recommend that you use the privileged account of the ApsaraDB RDS for PostgreSQL instance to ensure that you have all the required permissions.

Back up a database

1. Log on to your ECS instance or on-premises host. Then, run the following command to back up a database from the ApsaraDB RDS for PostgreSQL instance:

```
pg_dump -h '<hostname>' -U <username> -p <port> -Fc <dbname> > <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p> </div>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
-Fc	The output file format. <code>-Fc</code> specifies to use the custom format, which is ideal when you use <code>pg_restore</code> to import logical backup files and restore databases. For more information, visit pg_dump .
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -Fc testdb > /tmp/testdb.dump
```

2. When `Password:` appears, enter the password of the privileged account of the RDS instance and press Enter.

```
[root@izbp... etc]# pg_dump -h 'pgm-... pg.rds.aliyuncs.com' -U l... -p 3433 -Fc testdb > /tmp/testdb.dump
Password:
[root@izbp... etc]# ll /tmp/testdb.dump
-rw-r--r-- 1 root root 2006 Nov  5 16:05 /tmp/testdb.dump
[root@izbp... etc]#
```

Back up one or more tables

1. Log on to your ECS instance or on-premises host. Then, run the following command to back up one or more tables from a database in the ApsaraDB RDS for PostgreSQL instance:

```
pg_dump -h '<hostname>' -U <username> -p <port> -t <table> -Fc <dbname> > <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. <div style="border: 1px solid #add8e6; padding: 5px;"> <p>? Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS instance and the RDS instance have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p> </div>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
table	The name of the table you want to back up. You can use <code>-t <table></code> to specify more than one table.
-Fc	The output file format. <code>-Fc</code> specifies to use the custom format, which is ideal when you use <code>pg_restore</code> to import logical backup files and restore databases. For more information, visit pg_dump .
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -t products1 -Fc testdb2 > /tmp/testdb2.dump
```

- When `Password:` appears, enter the password of the privileged account of the RDS instance and press Enter.

```
[root@iZ... ~]# pg_dump -h 'pgm-bpxxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -t products1 -Fc testdb2 > /tmp/testdb2.d
ump
Password:
[root@iZ... ~]#
```

Back up a database with one or more tables excluded

- Log on to your ECS instance or on-premises host. Then, run the following command to back up a database from the ApsaraDB RDS for PostgreSQL instance with one or more tables excluded:

```
pg_dump -h '<hostname>' -U <username> -p <port> -T <table> -Fc <dbname> > <dumpdir>
```

Parameter	Description
-----------	-------------

Parameter	Description
hostname	<p>The endpoint of the ApsaraDB RDS for PostgreSQL instance.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p>? Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p> </div>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
table	The name of the table you want to exclude. You can use <code>-T <table></code> to specify more than one table.
-Fc	The output file format. <code>-Fc</code> specifies to use the custom format, which is ideal when you use <code>pg_restore</code> to import logical backup files and restore databases. For more information, visit pg_dump .
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -T products1 -Fc testdb2 > /tmp/testdb2.dump
```

- When `Password:` appears, enter the password of the privileged account of your RDS instance and press Enter.

```
[root@iZL... ~]# pg_dump -h 'pgm-bp...pg.rds.aliyuncs.com' -U test123 -p 3433 -T products1 -Fc testdb2 > /tmp/testdb2.d
ump
Password:
```

Back up the schema of a database with data excluded

- Log on to your ECS instance or on-premises host. Then, run the following command to back up the schema of a database from the ApsaraDB RDS for PostgreSQL instance.

```
pg_dump -h '<hostname>' -U <username> -p <port> -s -Fc <dbname> > <dumpdir>
```

Parameter	Description
-----------	-------------

Parameter	Description
hostname	<p>The endpoint of the ApsaraDB RDS for PostgreSQL instance.</p> <p>Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
-s	Specifies to only back up the schema of the database. The data of the database is not backed up. For more information, visit pg_dump .
-Fc	The output file format. <code>-Fc</code> specifies to use the custom format, which is ideal when you use <code>pg_restore</code> to import logical backup files and restore databases. For more information, visit pg_dump .
dbname	The name of the database that you want to back up.
dumpdir	The directory and name of the logical backup file to export.

Example:

```
pg_dump -h 'pgm-bpxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -s -Fc testdb2 > /tmp/testdb2.dump
```

- When `Password:` appears, enter the password of the privileged account of your RDS instance and press Enter.

```
[root@izb1-20130101020 ~]# pg_dump -h 'pgm-bpxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -s -Fc testdb2 > /tmp/testdb2.dump
Password:
[root@izb1-20130101020 ~]# ll /tmp/
total 16
srwxr-xr-x 1 root root 0 Nov 5 15:28 Aegis-...
-rw-r--r-- 1 root root 4 Nov 5 15:27 CmsGoAgent.pid
drwx----- 3 root root 4096 Nov 5 15:27 systemd-private-...
-rw-r--r-- 1 root root 2013 Nov 7 14:43 testdb2.dump
```

12.13.4. Create a full backup of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use the `pg_basebackup` utility provided by open source PostgreSQL to create a full backup of your ApsaraDB RDS for PostgreSQL instance and export the backup files to your computer.

Prerequisites

- The IP address of your ECS instance or on-premises host is added to a whitelist of your ApsaraDB RDS for PostgreSQL instance. For more information, see [Configure an IP address whitelist](#).

- Your ECS instance or on-premises host runs the same version of PostgreSQL as your ApsaraDB RDS for PostgreSQL instance.

Context

pg_basebackup backs up all data of a PostgreSQL instance. Backup files can be used for point-in-time recovery. For more information, see [pg_basebackup](#).

In this example, CentOS 7 and PostgreSQL 12 are used to create a full backup.

Precautions

We recommend that you use the privileged account of the ApsaraDB RDS for PostgreSQL instance to ensure that you have all the required permissions.

Procedure

Note pg_basebackup cannot back up a single database or database object. For more information about how to back up a single database or database object, see [Create a logical backup for an ApsaraDB RDS for PostgreSQL instance](#).

1. Log on to your ECS instance or on-premises host. Then, run the following command to back up a database from your RDS instance:

```
pg_basebackup -Ft -Pv -Xf -z -D <backupdir> -Z5 -h '<hostname>' -p <port> -U <username> -W
```

The following table describes parameters in this command. For more information, visit [pg_basebackup](#).

Parameter	Description
backupdir	The directory of backup files that are exported. The system automatically creates this directory. However, if this directory already exists and is not empty, the system reports an error.
hostname	The endpoint that you use to connect to your ApsaraDB RDS for PostgreSQL instance. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number .
port	The port that you use to connect to your ApsaraDB RDS for PostgreSQL instance.
username	A username of your ApsaraDB RDS for PostgreSQL instance.

Example:

```
pg_basebackup -Ft -Pv -Xf -z -D /pg12/backup1/ -Z5 -h pgm-bpxxxxx.pg.rds.aliyuncs.com -p 1433 -U test1 -W
```

2. When `Password:` appears, enter the password of the username of your RDS instance and press Enter.

```
[root@izbp-... ~]# pg_basebackup -Ft -Pv -Xf -z -D /pg12/backup/ -Z5 -h pgm-bpxxxxx.pg.rds.aliyuncs.com -p 1433 -U test1 -W
Password:
pg_basebackup: initiating base backup, waiting for checkpoint to complete
WARNING: skipping special file "/.s.PGSQL.3002"
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 14/8F000028 on timeline 1
WARNING: skipping special file "/.s.PGSQL.3002"/base.tar.gz
49065/49065 kB (100%), 1/1 tablespace
pg_basebackup: write-ahead log end point: 14/8F0003A0
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed
[root@izbp-... jqz ~]# ll /pg12/backup/
total 3956
-rw----- 1 root root 4047901 Apr 13 14:04 base.tar.gz
[root@izbp-... ~]#
```

12.14. Restoration

12.14.1. Restore data of an ApsaraDB RDS for PostgreSQL instance

This topic describes how to use the backup data of an ApsaraDB RDS for PostgreSQL instance to restore data.

Precautions

- The new instance must have the same whitelist, backup, and parameter settings as the original instance.
- The new instance must have the same data and account information as the backup set or instance at the time point.

Prerequisites

The original instance must meet the following conditions:

- The instance is in the Running state and is not locked.
- The original RDS instance does not have an ongoing migration task.
- If you want to restore an instance to a point in time, the log backup function is enabled for the original RDS instance.
- If you want to restore an instance from a backup set, the original RDS instance has at least one backup set.

Restore data of an ApsaraDB RDS for PostgreSQL instance

1. [Log on to the ApsaraDB for RDS console.](#)
2. On the **Instances** page, find the target instance.
3. Click the instance ID or click **Manage** in the **Actions** column corresponding to the instance to go to the **Basic Information** page.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. In the upper-right corner of the page, click **Restore Database (Previously Clone Database)**.
6. Configure the following parameters.

Section	Parameter	Description
Region	Region	The region where the instance resides.
Database Restoration	Restore Mode	<ul style="list-style-type: none"> ◦ By Time: You can restore data to any point in time within the retention period of the log backup. For more information about how to view or change the retention period of log backups, see Back up an ApsaraDB RDS for PostgreSQL instance. ◦ By Backup Set <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ? Note By Time is displayed only when the log backup feature is enabled. </div>
	Time	The time to which the database is restored. This parameter is displayed when you set Restore Mode to By Time.
	Backup Set	The backup set used to restore the database. This parameter is displayed when you set Restore Mode to By Backup Set.
	Instance Name	The name of the instance.
	Database Engine	The engine of the database, which varies with regions. The available database engines are displayed on the Restore RDS Instance page. The value of this parameter is set to PostgreSQL and cannot be changed.

Section	Parameter	Description
Specifications	Engine Version	The version of the database engine.
	Edition	The edition of the database. Select one from the drop-down list.
	Storage Type	None.
	Instance Type	The type of the instance. Memory size determines the maximum number of connections and IOPS. The actual values are displayed on the console.
	Storage	The storage capacity of the instance, including the space to store data, system files, binary log files, and transaction files. The minimum storage capacity is 20 GB. You can adjust the storage capacity.
Network Type	Network Type	<p>The network type of the instance. RDS instances support the following network types:</p> <ul style="list-style-type: none"> ◦ Classic Network: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ VPC: A Virtual Private Cloud (VPC) helps you to build an isolated network environment on Alibaba Cloud. You can customize the route table, IP address range, and gateway within a VPC. We recommend that you select VPC for improved security. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note After you select VPC as the network type, you must also select the corresponding VPC and VSwitch.</p> </div>

7. Click **Submit**.

12.14.2. Restore data from a logical backup file

This topic describes how to restore data from a logical backup file to an ApsaraDB RDS for PostgreSQL instance or an on-premises PostgreSQL database.

Context

A logical backup file is used to restore a small amount of data, such as data in a table. For a large amount of data, we recommend that you restore it from a full physical backup file to a new RDS instance and then use Alibaba Cloud Data Transmission Service (DTS) to migrate data to the original RDS instance.

Prerequisites

Data in the ApsaraDB RDS for PostgreSQL instance has been logically backed up. For more information, see [Create a logical backup for an ApsaraDB RDS for PostgreSQL instance](#).

Precautions

- We recommend that you do not restore data to the default postgres database.
- When you restore the data of a specific table, the system does not restore the database objects on which the table depends. Restoring a database may fail.

Restore the data of a database

1. Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to restore the data of a database:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> <dumpdir>
```

Parameter	Description
hostname	The endpoint of the ApsaraDB RDS for PostgreSQL instance. <div style="border: 1px solid #add8e6; padding: 5px;"> <p>Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p> </div>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose data you want to restore.
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb2 /tmp/testdb.dump
```

2. When **Password:** appears, enter the password of the privileged account of your RDS instance and press Enter.

Note You can ignore alerts generated by the embedded plpgsql plug-in.

```
[root@iZbj... ~]# pg_restore -h 'pgm-bp...pg.rds.aliyuncs.com' -U ... -p 3433 -d testdb4 /tmp/testdb2.dump
Password:
pg_restore: [archiver (db)] Error while PROCESSING TOC:
pg_restore: [archiver (db)] Error from TOC entry 3076; 0 0 COMMENT EXTENSION plpgsql
pg_restore: [archiver (db)] could not execute query: ERROR: must be owner of extension plpgsql
Command was: COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

WARNING: errors ignored on restore: 1
[...]
```

Restore the data of a table

1. Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to restore the data of a table:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> -t <table> -c <dumpdir>
```

Parameter	Description
-----------	-------------

Parameter	Description
hostname	<p>The endpoint of the ApsaraDB RDS for PostgreSQL instance.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p>? Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p> </div>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose data you want to restore.
table	The name of the table whose data you want to restore.
-c	-c : specifies to delete the database objects on which the table depends before data restoration. For more information, visit pg_restore .
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb2 -t products -c /tmp/testdb.dump
```

- When `Password:` appears, enter the password of the privileged account of your RDS instance and press **Enter**.



Restore the schema of a database with data excluded

- Log on to the ECS instance or on-premises host that houses the logical backup file and run the following command to only restore the schema of a database:

```
pg_restore -h '<hostname>' -U <username> -p <port> -d <dbname> -s <dumpdir>
```

Parameter	Description
-----------	-------------

Parameter	Description
hostname	<p>The endpoint of the ApsaraDB RDS for PostgreSQL instance.</p> <p>Note If your ECS instance connects to the ApsaraDB RDS for PostgreSQL instance by using an internal endpoint, make sure that the ECS and RDS instances have the same network type. If both instances use the VPC network type, make sure that both instances reside within the same VPC. For more information about how to view the internal endpoint, see View and modify the internal endpoint and port number.</p>
username	The username of the privileged account of the ApsaraDB RDS for PostgreSQL instance.
port	The port number of the ApsaraDB RDS for PostgreSQL instance.
dbname	The name of the database whose schema you want to restore.
-s	<code>-s</code> : specifies to restore only the schema of the database. The data of the database is not restored. For more information, visit pg_restore .
dumpdir	The directory and name of the logical backup file to use.

Example:

```
pg_restore -h 'pgm-bpxxxxx.pg.rds.aliyuncs.com' -U test123 -p 3433 -d testdb4 -s /tmp/testdb2.dump
```

- When `Password:` appears, enter the password of the privileged account of your RDS instance and press Enter.

Note You can ignore alerts generated by the embedded plpgsql plug-in.

```
[root@iZbp... ~]# pg_restore -h 'pgm-bp...pg.rds.aliyuncs.com' -U ... -p 3433 -d testdb4 -s /tmp/testdb2.dump
Password:
pg_restore: [archiver (db)] Error while PROCESSING TOC:
pg_restore: [archiver (db)] Error from TOC entry 3075; 0 0 COMMENT EXTENSION plpgsql
pg_restore: [archiver (db)] could not execute query: ERROR: must be owner of extension plpgsql
Command was: COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

WARNING: errors ignored on restore: 1
```

12.15. Plug-ins

12.15.1. Plug-ins supported

This topic lists the plug-ins that are supported by ApsaraDB RDS for PostgreSQL and their available versions.

PostgreSQL 10

Plug-in	Version
pg_stat_statements	1.6

Plug-in	Version
btree_gin	1.2
btree_gist	1.5
chkpass	1
citext	1.4
cube	1.2
dblink	1.2
dict_int	1
earthdistance	1.1
hstore	1.4
intagg	1.1
intarray	1.2
isn	1.1
ltree	1.1
pgcrypto	1.3
pgrowlocks	1.2
pg_prewarm	1.1
pg_trgm	1.3
postgres_fdw	1
sslinfo	1.2
tablefunc	1
unaccent	1.1
postgis_sfcgal	2.5.1
postgis_topology	2.5.1
fuzzystrmatch	1.1
postgis_tiger_geocoder	2.5.1
address_standardizer	2.5.1
address_standardizer_data_us	2.5.1
ogr_fdw	1
plperl	1
plv8	1.4.2

Plug-in	Version
plls	1.4.2
plcoffee	1.4.2
uuid-osp	1.1
zhparser	1
pgrouting	2.6.2
pg_hint_plan	1.3.0
pgstattuple	1.5
oss_fdw	1.1
ali_decoding	0.0.1
varbitx	1
pg_buffercache	1.3
q3c	1.5.0
pg_sphere	1
smlar	1
rum	1.3
pg_pathman	1.5
aggs_for_arrays	1.3.1
mysql_fdw	1
orafce	3.6
plproxy	2.8.0
pg_concurrency_control	1
postgis	2.5.1
ganos_geometry_sfcgal	2.2
ganos_geometry_topology	2.2
ganos_geometry	2.2
ganos_networking	2.2
ganos_pointcloud_geometry	2.2
ganos_pointcloud	2.2
ganos_raster	2.2
ganos_spatialref	2.2

Plug-in	Version
ganos_trajectory	2.2
ganos_tiger_geocoder	2.2
ganos_address_standardizer	2.2
ganos_address_standardizer_data_us	2.2

PostgreSQL 9.4

Plug-in	Version
plpgsql	1
pg_stat_statements	1.2
btree_gin	1
btree_gist	1
chkpass	1
citext	1
cube	1
dblink	1.1
dict_int	1
earthdistance	1
hstore	1.3
intagg	1
intarray	1
isn	1
ltree	1
pgcrypto	1.1
pgrowlocks	1.1
pg_prewarm	1
pg_trgm	1.1
postgres_fdw	1
sslinfo	1
tablefunc	1
tsearch2	1
unaccent	1

Plug-in	Version
postgis	2.2.8
postgis_topology	2.2.8
fuzzystrmatch	1
postgis_tiger_geocoder	2.2.8
plperl	1
pltcl	1
plv8	1.4.2
plls	1.4.2
plcoffee	1.4.2
uuid-oss	1
zhparser	1
pgrouting	2.0.0
rdkit	3.4
pg_hint_plan	1.1.3
pgstattuple	1.2
oss_fdw	1.1
jsonbx	1
ali_decoding	0.0.1
varbitx	1
pg_buffercache	1
smlar	1
pg_sphere	1
q3c	1.5.0
pg_awr	1
imgsmr	1
orafce	3.6
pg_concurrency_control	1

12.15.2. Use mysql_fdw to read and write data from and to a MySQL database

This topic describes how to use the `mysql_fdw` plug-in of ApsaraDB RDS for PostgreSQL to read and write data from and to a database on an ApsaraDB RDS for MySQL instance or a user-created MySQL database.

Prerequisites

- Your RDS instance runs PostgreSQL 10.
- Communication between your ApsaraDB RDS for PostgreSQL instance and the target MySQL database is normal.

Context

PostgreSQL 9.6 and later support parallel computing. PostgreSQL 11 can use joins on up to a billion data records to complete queries in seconds. A number of users prefer to use PostgreSQL to build small sized data warehouses and process highly concurrent access requests. PostgreSQL 13 is under development. It will support columnar storage engines that further improve analysis capabilities.

The `mysql_fdw` plug-in establishes a connection to synchronize data from a MySQL database to your ApsaraDB RDS for PostgreSQL instance.

Procedure

1. Create the `mysql_fdw` plug-in.

```
postgres=> create extension mysql_fdw;  
CREATE EXTENSION
```

2. Define a MySQL server.

```
postgres=> CREATE SERVER <The name of the MySQL server>  
postgres-> FOREIGN DATA WRAPPER mysql_fdw  
postgres-> OPTIONS (host '<The endpoint used to connect to the MySQL server>', port '<The port used to connect the MySQL server>');  
CREATE SERVER
```

Example:

```
postgres=> CREATE SERVER mysql_server  
postgres-> FOREIGN DATA WRAPPER mysql_fdw  
postgres-> OPTIONS (host 'rm-xxx.mysql.rds.aliyuncs.com', port '3306');  
CREATE SERVER
```

3. Map the MySQL server to an account created on your ApsaraDB RDS for PostgreSQL instance. The account can then be used to read and write data to the target MySQL database on the MySQL server.

```
postgres=> CREATE USER MAPPING FOR <The username of the account to which the MySQL server is mapped>  
SERVER <The name of the MySQL server>  
OPTIONS (username '<The username used to log on to the target MySQL database>', password '<The password used to log on to the target MySQL database>');  
CREATE USER MAPPING
```

Example:

```
postgres=> CREATE USER MAPPING FOR pgtest  
SERVER mysql_server  
OPTIONS (username 'mysqltest', password 'Test1234!');  
CREATE USER MAPPING
```

4. Create a foreign MySQL table by using the account that you mapped to the MySQL server in the previous

step.

Note The field names in the foreign MySQL table must be the same as those in the target table of the target MySQL database. You can choose to create only the fields you want to query. For example, if the target table in the target MySQL database contains three fields, ID, NAME, and AGE, you only need to create two fields, ID and NAME, in the foreign MySQL table.

```
postgres=> CREATE FOREIGN TABLE <The name of the foreign MySQL table> (<The name of Field 1> <The data type of Field 1>,<The name of Field 2> <The data type of Field 2>...) server <The name of the MySQL server> options (dbname '<The name of the target MySQL database>', table_name '<The name of the target table in the target MySQL database>');
```

CREATE FOREIGN TABLE

Example:

```
postgres=> CREATE FOREIGN TABLE ft_test (id1 int, name1 text) server mysql_server options (dbname 'test123', table_name 'test');
```

CREATE FOREIGN TABLE

What to do next

You can use the foreign MySQL table to test the performance of reading and writing data from and to the target MySQL database.

Note Data can be written to the target table in the target MySQL database only when the target table is assigned a primary key. If the target table is not assigned a primary key, the following error is displayed:

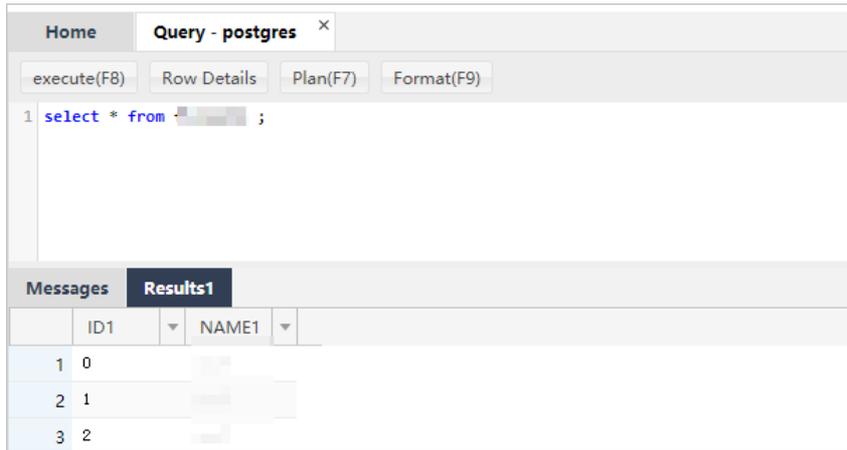
```
ERROR: first column of remote table must be unique for INSERT/UPDATE/DELETE operation.
```

```
postgres=> select * from ft_test ;

postgres=> insert into ft_test values (2,'abc');
INSERT 0 1

postgres=> insert into ft_test select generate_series(3,100),'abc';
INSERT 0 98

postgres=> select count(*) from ft_test ;
count
-----
    99
(1 row)
```



Check query plans to find out how the requests sent from your ApsaraDB RDS for PostgreSQL instance are executed to query data from the target MySQL database.

```
postgres=> explain verbose select count(*) from ft_test ;
                QUERY PLAN
-----
Aggregate  (cost=1027.50..1027.51 rows=1 width=8)
  Output: count(*)
  -> Foreign Scan on public.ft_test (cost=25.00..1025.00 rows=1000 width=0)
    Output: id, info
    Remote server startup cost: 25
    Remote query: SELECT NULL FROM `test123`.`test`
(6 rows)

postgres=> explain verbose select id from ft_test where id=2;
                QUERY PLAN
-----
Foreign Scan on public.ft_test (cost=25.00..1025.00 rows=1000 width=4)
  Output: id
  Remote server startup cost: 25
  Remote query: SELECT `id` FROM `test123`.`test` WHERE ((`id` = 2))
(4 rows)
```

12.15.3. Read and write foreign data files by using oss_fdw

This topic describes how to use the oss_fdw plug-in to load data between Object Storage Service (OSS) and PostgreSQL or PPAS databases.

oss_fdw parameters

The oss_fdw plug-in uses a method similar to other Foreign Data Wrapper (FDW) interfaces to encapsulate foreign data stored in OSS. You can use oss_fdw to read data stored in OSS. This process is similar to reading data tables. oss_fdw provides unique parameters to connect and parse file data in OSS.

Note

- `oss_fdw` can read and write files of the following types in OSS: TEXT and CSV files as well as GZIP-compressed TEXT and CSV files.
- The value of each parameter must be enclosed in double quotation marks (") and cannot contain unnecessary spaces.

CREATE SERVER parameters

- `ossendpoint`: the endpoint used to access OSS through the internal network, also known as the host.
- `id oss`: the AccessKey ID of the OSS account.
- `key oss`: the AccessKey secret of the OSS account.
- `bucket`: the bucket where the data you want to access is stored. You must create an OSS account before you specify this parameter.

The following fault tolerance parameters can be used for data import and export. If network connectivity is poor, you can adjust these parameters as necessary to ensure successful import and export.

- `oss_connect_timeout`: indicates the connection timeout period. Default value: 10. Unit: seconds.
- `oss_dns_cache_timeout`: indicates the DNS timeout period. Default value: 60. Unit: seconds.
- `oss_speed_limit`: indicates the minimum data transmission rate. Default value: 1024. Unit: byte/s.
- `oss_speed_time`: the maximum waiting period during which the data transmission rate is lower than its minimum value. Default value: 15. Unit: seconds.

If the default values of `oss_speed_limit` and `oss_speed_time` are used, a timeout error occurs when the transmission rate is lower than 1,024 byte/s for 15 consecutive seconds.

CREATE FOREIGN TABLE parameters

- `filepath`: a file name that contains a path in OSS.
 - The file name specified by this parameter contains the directory name but not the bucket name.
 - This parameter matches multiple files in the corresponding path in OSS. You can load multiple files to a database.
 - You can import files that adhere to the `filepath` or `filepath.x` format to a database. The values of `x` must be consecutive numbers starting from 1.

For example, among the files named `filepath`, `filepath.1`, `filepath.2`, `filepath.3`, and `filepath.5`, the first four files are matched and imported. The `filepath.5` file is not imported.

- `dir`: the virtual file directory in OSS.
 - The specified directory must end with a forward slash (/).
 - All files (excluding subfolders and files in subfolders) in the virtual file directory specified by `dir` will be matched and imported to a database.
- `prefix`: the prefix of the path name corresponding to the data file. The prefix does not support regular expressions. The `prefix`, `filepath`, and `dir` parameters are mutually exclusive, so only one of them can be specified at a time.
- `format`: the file format, which can only be `csv`.
- `encoding`: the file data encoding format. It supports common PostgreSQL encoding formats, such as UTF-8.
- `parse_errors`: the fault-tolerant parsing mode. If an error occurs during the parsing process, the entire row of data is ignored.
- `delimiter`: the string used to delimit columns.
- `quote`: the quote character for files.
- `escape`: the escape character for files.
- `null`: sets the column matching a specified string to null. For example, `null 'test'` is used to set the value of the `'test'` column to null.

- **force_not_null**: sets the value of a column to a non-null value. For example, `force_not_null 'id'` is used to set the value of the 'id' column to empty strings.
- **compressiontype**: specifies the format of the files to be read or written in OSS.
 - **none**: The files are uncompressed. This is the default value.
 - **gzip**: The files are compressed in the GZIP format.
- **compressionlevel**: specifies the degree to which data files written to OSS are compressed. Valid values: 1 to 9. Default value: 6.

 **Note**

- You must specify `filepath` and `dir` in the `OPTIONS` parameter.
- You must specify either `filepath` or `dir`.
- Export mode only supports `dir`.

Export mode parameters for CREATE FOREIGN TABLE

- **oss_flush_block_size**: the buffer size for the data written to OSS at a time. Default value: 32 MB. Valid values: 1 MB to 128 MB.
- **oss_file_max_size**: the maximum size of a data file allowed to be written to OSS. If a data file reaches the maximum size, the remaining data is written to another data file. Default value: 1024. Valid values: 8 to 4000. Unit: MB.
- **num_parallel_worker**: the maximum number of threads that are allowed to run in parallel to compress the data written to OSS. Valid values: 1 to 8. Default value: 3.

Auxiliary functions

`FUNCTION oss_fdw_list_file (relname text, schema text DEFAULT 'public')`

- This function obtains the name and size of the OSS file that a foreign table matches.
- The unit of file size is byte.

```
select * from oss_fdw_list_file('t_oss');
      name      | size
-----+-----
oss_test/test.gz.1 | 739698350
oss_test/test.gz.2 | 739413041
oss_test/test.gz.3 | 739562048
(3 rows)
```

Auxiliary features

`oss_fdw.rds_read_one_file`: In read mode, this feature is used to specify a file to match the foreign table. The foreign table matches only the specified file during data import.

Example: `set oss_fdw.rds_read_one_file = 'oss_test/example16.csv.1';`

```
set oss_fdw.rds_read_one_file = 'oss_test/test.gz.2';
select * from oss_fdw_list_file('t_oss');
      name      | size
-----+-----
oss_test/test.gz.2 | 739413041
(1 rows)
```

oss_fdw example

```
# Create the plug-in for a PostgreSQL database.
create extension oss_fdw; --- For a PPAS database, execute select rds_manage_extension('create','oss_fdw');
# Create a server.
CREATE SERVER ossserver FOREIGN DATA WRAPPER oss_fdw OPTIONS
  (host 'oss-cn-hangzhou.aliyuncs.com', id 'xxx', key 'xxx', bucket 'mybucket');
# Create an OSS foreign table.
CREATE FOREIGN TABLE ossexample
  (date text, time text, open float,
   high float, low float, volume int);
SERVER ossserver
OPTIONS ( filepath 'osstest/example.csv', delimiter ',',
          format 'csv', encoding 'utf8', PARSE_ERRORS '100');
# Create a table named example to which to import data.
create table example
  (date text, time text, open float,
   high float, low float, volume int);
# Load data from ossexample to example.
insert into example select * from ossexample;
# Result
# oss_fdw estimates the file size in OSS and formulates a query plan.
explain insert into example select * from ossexample;
          QUERY PLAN
-----
Insert on example (cost=0.00..1.60 rows=6 width=92)
-> Foreign Scan on ossexample (cost=0.00..1.60 rows=6 width=92)
    Foreign OssFile: osstest/example.csv.0
    Foreign OssFile Size: 728
(4 rows)
# Write the data in the example table to OSS.
insert into ossexample select * from example;
explain insert into ossexample select * from example;
          QUERY PLAN
-----
Insert on ossexample (cost=0.00..16.60 rows=660 width=92)
-> Seq Scan on example (cost=0.00..16.60 rows=660 width=92)
(2 rows)
```

Additional considerations

- `oss_fdw` is a foreign table plug-in developed based on the PostgreSQL FOREIGN TABLE framework.
- The data import performance is subject to the PostgreSQL cluster resources (CPU, I/O, and memory) and OSS.
- To ensure data import performance, the ApsaraDB RDS for PostgreSQL instance must be in the same region as the OSS bucket.
- If the error `ERROR: oss endpoint userendpoint not in aliyun white list` is reported during foreign table reading, use the endpoints listed in [Regions and endpoints](#). If the problem persists, submit a ticket.

Troubleshooting

When an import or export error occurs, the log displays the following error information:

- **code**: the HTTP status code of the request that has failed.
- **error_code**: the error code returned by OSS.
- **error_msg**: the error message returned by OSS.
- **req_id**: the universally unique identifier (UUID) that identifies the request. If you require assistance in solving a problem, you can submit a ticket containing the req_id of the failed request to OSS developers.

For more information about errors, see the following references. Timeout errors can be handled by using `oss_ext` parameters.

- [Object Storage Service](#)
- [PostgreSQL CREATE FOREIGN TABLE manual](#)
- [OSS error response](#)

ID and key encryption

If the id and key parameters for `CREATE SERVER` are not encrypted, the `select * from pg_foreign_server` statement execution result will display the information. Your AccessKey ID and AccessKey secret will be exposed. You can use symmetric encryption to hide your AccessKey ID and AccessKey secret. Use different AccessKey pairs for different instances to further protect your information. However, to avoid incompatibility with earlier versions, do not add data types as you would in Greenplum.

Encrypted information:

```
postgres=# select * from pg_foreign_server ;
  srvname | srvowner | srvfdw | srvtype | srvversion | srvacl |                                srvoptions
-----+-----+-----+-----+-----+-----+-----
ossserver |    10 | 16390 |         |           |        |  |{host=oss-cn-hangzhou-zmf.aliyuncs.com,id=MD5xxxxxxxx,key=MD5xxxxx
xxx,bucket=067862}
```

The encrypted information is preceded by the MD5 hash value. The remainder of the total length divided by 8 is 3. Therefore, encryption is not performed again when the exported data is imported. You cannot create an AccessKey pair that is preceded by MD5.

12.16. Use Pgpool for read/write splitting in ApsaraDB RDS for PostgreSQL

This topic describes how to use the Pgpool tool of PostgreSQL installed on an ECS instance to implement read/write splitting for your primary and read-only ApsaraDB RDS for PostgreSQL instances.

Context

If you do not use Pgpool to ensure high availability, Pgpool is stateless. The decrease in performance can be ignored. Additionally, Pgpool supports horizontal scaling of your database system. You can use Pgpool with the high availability architecture of ApsaraDB RDS for PostgreSQL to implement read/write splitting.

Set up a test environment

If you have purchased a primary RDS instance that runs PostgreSQL 10 and have attached read-only instances to the primary instance, the only thing you need to do is to [install Pgpool](#) . For more information, see [Create an instance](#) and [Create a read-only ApsaraDB RDS for PostgreSQL instance](#). After you install Pgpool, go to [Configure Pgpool](#).

1. Modify the sysctl.conf file.

```
vi /etc/sysctl.conf

# add by digoyal.zhou
fs.aio-max-nr = 1048576
fs.file-max = 76724600

# Optional. Set the kernel.core_pattern parameter to /data01/corefiles/core_%e_%u_%t_%s.%p.
# The /data01/corefiles directory that is used to store core dumps is created with the 777 permission before testing. If a symbolic link is used, change the directory to 777.

kernel.sem = 4096 2147483647 2147483646 512000
# Specify the semaphore. You can run the ipcs -l or -u command to obtain the semaphore count. Each group of 16 processes requires a semaphore with a count of 17.

kernel.shmall = 107374182
# Specify the total size of shared memory segments. Recommended value: 80% of the memory capacity. Unit: pages.
kernel.shmmax = 274877906944
# Specify the maximum size of a single shared memory segment. Recommended value: 50% of the memory capacity. Unit: bytes. In PostgreSQL versions later than 9.2, the use of shared memory significantly drops.
kernel.shmni = 819200
# Specify the total number of shared memory segments that can be generated. At least two shared memory segments must be generated within each PostgreSQL cluster.

net.core.netdev_max_backlog = 10000
net.core.rmem_default = 262144
# The default setting of the socket receive buffer in bytes.
net.core.rmem_max = 4194304
# The maximum receive socket buffer size in bytes
net.core.wmem_default = 262144
# The default setting (in bytes) of the socket send buffer.
net.core.wmem_max = 4194304
# The maximum send socket buffer size in bytes.
net.core.somaxconn = 4096
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_keepalive_intvl = 20
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_mem = 8388608 12582912 16777216
net.ipv4.tcp_fin_timeout = 5
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syncookies = 1
# Enable SYN cookies. If an SYN waiting queue overflows, you can enable SYN cookies to defend against a small number of SYN attacks.
net.ipv4.tcp_timestamps = 1
```

```
# Reduce the time after which a network socket enters the TIME-WAIT state.
net.ipv4.tcp_tw_recycle = 0
# If you set this parameter to 1 to enable the recycle function, network sockets in the TIME-WAIT state over TCP c
onnections are recycled. However, if network address translation (NAT) is used, TCP connections may fail. We rec
ommend that you set this parameter to 0 on the database server.
net.ipv4.tcp_tw_reuse = 1
# Enable the reuse function. This function enables network sockets in the TIME-WAIT state to be reused over new
TCP connections.
net.ipv4.tcp_max_tw_buckets = 262144
net.ipv4.tcp_rmem = 8192 87380 16777216
net.ipv4.tcp_wmem = 8192 65536 16777216

net.nf_contrack_max = 1200000
net.netfilter.nf_contrack_max = 1200000

vm.dirty_background_bytes = 409600000
# If the size of dirty pages reaches the specified limit, a background scheduling process (for example, pdflush) is i
nvoked to flush the dirty pages to disks. These are the pages that are generated n seconds earlier. The value of
n is calculated by using the following formula: n = The value of the dirty_expire_centisecs parameter/100.

# The default limit is 10% of the memory capacity. If the memory capacity is large, we recommend that you specify
the limit in bytes.

vm.dirty_expire_centisecs = 3000
# Specify the maximum period to retain dirty pages. Dirty pages are flushed to disks after the time period specifie
d by this parameter elapses. A value of 3000 indicates 30 seconds.
vm.dirty_ratio = 95
# The processes that users call to write data onto disks must actively flush dirty pages to disks. This applies whe
n the background scheduling process to flush dirty pages is slow and the size of dirty pages exceeds 95% of the
memory capacity. These processes include fsync and fdatasync.
# Set this parameter properly to prevent user-called processes from flushing dirty pages to disks. This allows you
to create multiple RDS instances on a single server and use control groups to limit the input/output operations pe
r second (IOPS) per instance.

vm.dirty_writeback_centisecs = 100
# Specify the time interval at which the background scheduling process (such as pdflush) flushes dirty pages to di
sks. The value 100 indicates 1 second.

vm.swappiness = 0
# Disable the swap function.

vm.mmap_min_addr = 65536
vm.overcommit_memory = 0
# Specify whether you can allocate more memory space than the physical host has available. If you set this param
eter to 1, the system always considers the available memory space sufficient. If the memory capacity provided in t
he test environment is low, we recommend that you set this parameter to 1.
```

```

vm.overcommit_ratio = 90
# Specify the memory capacity that can be allocated when the overcommit_memory parameter is set to 2.
vm.swappiness = 0
# Disable the swap function.
vm.zone_reclaim_mode = 0
# Disable non-uniform memory access (NUMA). You can also disable NUMA in the vmlinux file.
net.ipv4.ip_local_port_range = 40000 65535
# Specify the range of TCP or UDP port numbers for the physical host to allocate.
fs.nr_open=20480000
# Specify the maximum number of file handles that a single process can open.

# Note the following parameters:
#vm.extra_free_kbytes = 4096000 # If the physical host provides a low memory capacity, do not specify a large value such as 4096000. If you specify a large value, the physical host may not start.
#vm.min_free_kbytes = 6291456 # We recommend that you increase the value of the vm.min_free_kbytes parameter by 1 GB for every 32 GB of memory.
# If the physical host does not provide much memory, we recommend that you do not configure vm.extra_free_kbytes and vm.min_free_kbytes.
# vm.nr_hugepages = 66536
# If the size of the shared buffer exceeds 64 GB, we recommend that you use huge pages. You can specify the page size by setting the Hugepagesize parameter in the /proc/meminfo file.
#vm.lowmem_reserve_ratio = 1 1 1
# If the memory capacity exceeds 64 GB, we recommend that you set this parameter. Otherwise, we recommend that you retain the default value 256 256 32.

```

2. Modify the limits.conf file.

```

vi /etc/security/limits.conf

* soft nfile 1024000
* hard nfile 1024000
* soft nproc unlimited
* hard nproc unlimited
* soft core unlimited
* hard core unlimited
* soft memlock unlimited
* hard memlock unlimited

# Comment out the other parameters in the limits.conf file.
# Comment out the /etc/security/limits.d/20-nproc.conf file.

```

3. Disable transparent huge pages, configure huge pages, and start PostgreSQL.

```

chmod +x /etc/rc.d/rc.local

vi /etc/rc.local

# Disable transparent huge pages.
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi

# Configure huge pages for two instances. Each instance has a shared buffer of 16 GB.
#sysctl -w vm.nr_hugepages=17000

# Start the two instances.
su - postgres -c "pg_ctl start -D /data01/pg12_3389/pg_root"
su - postgres -c "pg_ctl start -D /data01/pg12_8002/pg_root"

```

4. Create a file system.

 **Warning** If you use a new disk, you must verify that the new disk belongs to the vdb partition instead of the vda partition. If the new disk belongs to the vda partition, data may be deleted from the new disk.

```

parted -a optimal -s /dev/vdb mklabel gpt mkpart primary 1MiB 100%FREE
mkfs.ext4 /dev/vdb1 -m 0 -O extent,uninit_bg -E lazy_itable_init=1 -b 4096 -T largefile -L vdb1
vi /etc/fstab
LABEL=vdb1 /data01 ext4 defaults,noatime,nodiratime,nodelalloc,barrier=0,data=writeback 0 0

mkdir /data01
mount -a

```

5. Start the irqbalance command line tool.

```

systemctl status irqbalance
systemctl enable irqbalance
systemctl start irqbalance
systemctl status irqbalance

```

6. Install PostgreSQL 10 and Pgpool.

```

yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum install -y https://download.postgresql.org/pub/repos/yum/repopms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
yum search all postgresql
yum search all pgpool

yum install -y postgresql12*
yum install -y pgpool-II-12-extensions

```

7. Initialize the data directory of your database system.

```
mkdir /data01/pg12_3389
chown postgres:postgres /data01/pg12_3389
```

8. Configure environment variables for the postgres user.

```
su - postgres
vi .bash_profile

# Append the following parameters:
export PS1="$USER@`/bin/hostname -s`-> "
export PGPORT=3389
export PGDATA=/data01/pg12_${PGPORT}/pg_root

export LANG=en_US.utf8
export PGHOME=/usr/pgsql-12
export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD_LIBRARY_
PATH
export DATE=`date +%Y%m%d%H%M`
export PATH=$PGHOME/bin:$PATH:.
export MANPATH=$PGHOME/share/man:$MANPATH
export PGHOST=$PGDATA
export PGUSER=postgres
export PGDATABASE=db1
alias rm='rm -i'
alias ll='ls -lh'
unalias vi
```

9. Initialize your primary RDS instance.

```
initdb -D $PGDATA -U postgres -E UTF8 --lc-collate=C --lc-ctype=en_US.utf8
```

10. Modify the postgresql.conf file.

```
listen_addresses = '0.0.0.0'
port = 3389
max_connections = 1500
superuser_reserved_connections = 13
unix_socket_directories = '., /var/run/postgresql, /tmp'
tcp_keepalives_idle = 60
tcp_keepalives_interval = 10
tcp_keepalives_count = 10
shared_buffers = 16GB
huge_pages = on
work_mem = 8MB
maintenance_work_mem = 1GB
dynamic_shared_memory_type = posix
vacuum_cost_delay = 0
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10.0
```

```
bgwriter_wal_multiplier = 100  
bgwriter_flush_after = 512kB  
effective_io_concurrency = 0  
max_worker_processes = 128  
max_parallel_maintenance_workers = 3  
max_parallel_workers_per_gather = 4  
parallel_leader_participation = off  
max_parallel_workers = 8  
backend_flush_after = 256  
wal_level = replica  
synchronous_commit = off  
full_page_writes = on  
wal_compression = on  
wal_buffers = 16MB  
wal_writer_delay = 10ms  
wal_writer_flush_after = 1MB  
checkpoint_timeout = 15min  
max_wal_size = 64GB  
min_wal_size = 8GB  
checkpoint_completion_target = 0.2  
checkpoint_flush_after = 256kB  
random_page_cost = 1.1  
effective_cache_size = 48GB  
log_destination = 'csvlog'  
logging_collector = on  
log_directory = 'log'  
log_filename = 'postgresql-%a.log'  
log_truncate_on_rotation = on  
log_rotation_age = 1d  
log_rotation_size = 0  
log_min_duration_statement = 1s  
log_checkpoints = on  
log_connections = on  
log_disconnections = on  
log_line_prefix = '%m [%p] '  
log_statement = 'ddl'  
log_timezone = 'Asia/Shanghai'  
autovacuum = on  
log_autovacuum_min_duration = 0  
autovacuum_vacuum_scale_factor = 0.1  
autovacuum_analyze_scale_factor = 0.05  
autovacuum_freeze_max_age = 800000000  
autovacuum_multixact_freeze_max_age = 900000000  
autovacuum_vacuum_cost_delay = 0  
vacuum_freeze_table_age = 750000000  
vacuum_multixact_freeze_table_age = 750000000  
datestyle = 'iso, mdy'
```

```

timezone = 'Asia/Shanghai'
lc_messages = 'en_US.utf8'
lc_monetary = 'en_US.utf8'
lc_numeric = 'en_US.utf8'
lc_time = 'en_US.utf8'
default_text_search_config = 'pg_catalog.english'

```

11. Modify the pg_hba.conf file.

 **Note** Pgpool-II is installed on the same ECS instance as the database server where PostgreSQL resides. If you specify the 127.0.0.1 IP address in the pg_hba.conf file, you must enter the correct password to ensure a successful logon.

```

# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
host db123 digoal 0.0.0.0/0 md5

```

12. Create a user authorized with streaming replication permissions. Example:

```

db1=# create role rep123 login replication encrypted password 'xxxxxxx';
CREATE ROLE

```

13. Create a user and authorize them to manage your RDS instances. Example:

```

db1=# create role digoal login encrypted password 'xxxxxxx';
CREATE ROLE

db1=# create database db123 owner digoal;
CREATE DATABASE

```

14. Create a user who is authorized to check the health heartbeats between Pgpool and your read-only RDS instances. With the parameters of Pgpool properly configured, this user can check the write-ahead logging (WAL) replay latency on each read-only RDS instance. Example:

```

create role nobody login encrypted password 'xxxxxxx';

```

Create a secondary RDS instance

To simplify the test procedure, follow these steps to create a secondary RDS instance on the same ECS instance as your primary RDS instance.

1. Use the pg_basebackup tool to create an online secondary RDS instance. Example:

```

pg_basebackup -D /data01/pg12_8002/pg_root -F p --checkpoint=fast -P -h 127.0.0.1 -p 3389 -U rep123

```

2. Modify the postgresql.conf file of the secondary RDS instance.

```
cd /data01/pg12_8002/pg_root

vi postgresql.conf

# The secondary RDS instance has the following configuration different from the primary RDS instance:
port = 8002
primary_conninfo = 'hostaddr=127.0.0.1 port=3389 user=rep123' # You do not need to set the password. This is because trust relationships are configured on the primary RDS instance.
hot_standby = on
wal_receiver_status_interval = 1s
wal_receiver_timeout = 10s
recovery_target_timeline = 'latest'
```

3. Configure the standby.signal file of the secondary RDS instance.

```
cd /data01/pg12_8002/pg_root

touch standby.signal
```

4. Check whether the data synchronization between the primary and secondary RDS instances is normal.

```
db1=# select * from pg_stat_replication ;
-[ RECORD 1 ]-----+-----
pid          | 21065
usesysid     | 10
username     | postgres
application_name | walreceiver
client_addr  | 127.0.0.1
client_hostname |
client_port  | 47064
backend_start | 2020-02-29 00:26:28.485427+08
backend_xmin  |
state        | streaming
sent_lsn     | 0/52000060
write_lsn    | 0/52000060
flush_lsn    | 0/52000060
replay_lsn   | 0/52000060
write_lag    |
flush_lag    |
replay_lag   |
sync_priority | 0
sync_state   | async
reply_time   | 2020-02-29 01:32:40.635183+08
```

Configure Pgpool

1. Query the location where Pgpool is installed.

```
# rpm -qa|grep pgpool
pgpool-II-12-extensions-4.1.1-1.rhel7.x86_64
pgpool-II-12-4.1.1-1.rhel7.x86_64

# rpm -ql pgpool-II-12-4.1.1
```

2. Modify the pgpool.conf file.

```
# cd /etc/pgpool-II-12/

cp pgpool.conf.sample-stream pgpool.conf

vi pgpool.conf

# -----
# pgPool-II configuration file
# -----
#
# This file consists of lines of the form:
#
# name = value
#
# Whitespace may be used. Comments are introduced with "#" anywhere on a line.
# The complete list of parameter names and allowed values can be found in the
# pgPool-II documentation.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal. If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pgpool reload". Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#

#-----
# CONNECTIONS
#-----

# - pgpool Connection Settings -

listen_addresses = '0.0.0.0'
    # Host name or IP address to listen on:
    # '*' for all, '' for no TCP/IP connections
    # (change requires restart)

port = 8001
    # Port number
    # (change requires restart)

socket_dir = /tmp/
```

```
socket_dir = '/tmp'
    # Unix domain socket path
    # The Debian package defaults to
    # /var/run/postgresql
    # (change requires restart)
reserved_connections = 0
    # Number of reserved connections.
    # Pgpool-II does not accept connections if over
    # num_init_children - reserved_connections.

# - pgpool Communication Manager Connection Settings -

pcp_listen_addresses = "
    # Host name or IP address for pcp process to listen on:
    # '*' for all, '' for no TCP/IP connections
    # (change requires restart)
pcp_port = 9898
    # Port number for pcp
    # (change requires restart)
pcp_socket_dir = '/tmp'
    # Unix domain socket path for pcp
    # The Debian package defaults to
    # /var/run/postgresql
    # (change requires restart)
listen_backlog_multiplier = 2
    # Set the backlog parameter of listen(2) to
    # num_init_children * listen_backlog_multiplier.
    # (change requires restart)
serialize_accept = off
    # whether to serialize accept() call to avoid thundering herd problem
    # (change requires restart)

# - Backend Connection Settings -

backend_hostname0 = '127.0.0.1'
    # Host name or IP address to connect to for backend 0
backend_port0 = 3389
    # Port number for backend 0
backend_weight0 = 1
    # Weight for backend 0 (only in load balancing mode)
backend_data_directory0 = '/data01/pg12_3389/pg_root'
    # Data directory for backend 0
backend_flag0 = 'ALWAYS_MASTER'
    # Controls various backend behavior
    # ALLOW_TO_FAILOVER, DISALLOW_TO_FAILOVER
    # or ALWAYS_MASTER
```

```
backend_application_name0 = 'server0'
    # walsender's application_name, used for "show pool_nodes" command
backend_hostname1 = '127.0.0.1'
backend_port1 = 8002
backend_weight1 = 1
backend_data_directory1 = '/data01/pg12_8002/pg_root'
backend_flag1 = 'DISALLOW_TO_FAILOVER'
backend_application_name1 = 'server1'

# - Authentication -

enable_pool_hba = on
    # Use pool_hba.conf for client authentication
pool_passwd = 'pool_passwd'
    # File name of pool_passwd for md5 authentication.
    # "" disables pool_passwd.
    # (change requires restart)
authentication_timeout = 60
    # Delay in seconds to complete client authentication
    # 0 means no timeout.

allow_clear_text_frontend_auth = off
    # Allow Pgpool-II to use clear text password authentication
    # with clients, when pool_passwd does not
    # contain the user password

# - SSL Connections -

ssl = off
    # Enable SSL support
    # (change requires restart)
ssl_key = './server.key'
    # Path to the SSL private key file
    # (change requires restart)
ssl_cert = './server.cert'
    # Path to the SSL public certificate file
    # (change requires restart)
ssl_ca_cert = ""
    # Path to a single PEM format file
    # containing CA root certificate(s)
    # (change requires restart)
ssl_ca_cert_dir = ""
    # Directory containing CA root certificate(s)
    # (change requires restart)

ssl_ciphers = 'HIGH:MEDIUM:+3DES:! aNULL'
    # Allowed SSL ciphers
```

```
        # (change requires restart)
ssl_prefer_server_ciphers = off
        # Use server's SSL cipher preferences,
        # rather than the client's
        # (change requires restart)
ssl_ecdh_curve = 'prime256v1'
        # Name of the curve to use in ECDH key exchange
ssl_dh_params_file = ""
        # Name of the file containing Diffie-Hellman parameters used
        # for so-called ephemeral DH family of SSL cipher.

#-----
# POOLS
#-----

# - Concurrent session and pool size -

num_init_children = 128
        # Number of concurrent sessions allowed
        # (change requires restart)
max_pool = 4
        # Number of connection pool caches per connection
        # (change requires restart)

# - Life time -

child_life_time = 300
        # Pool exits after being idle for this many seconds
child_max_connections = 0
        # Pool exits after receiving that many connections
        # 0 means no exit
connection_life_time = 0
        # Connection to backend closes after being idle for this many seconds
        # 0 means no close
client_idle_limit = 0
        # Client is disconnected after being idle for that many seconds
        # (even inside an explicit transactions!)
        # 0 means no disconnection

#-----
# LOGS
#-----

# - Where to log -
```

```
log_destination = 'syslog'
    # Where to log
    # Valid values are combinations of stderr,
    # and syslog. Default to stderr.

# - What to log -

log_line_prefix = '%t: pid %p: ' # printf-style string to output at beginning of each log line.

log_connections = on
    # Log connections

log_hostname = off
    # Hostname will be shown in ps status
    # and in logs if connections are logged

log_statement = off
    # Log all statements

log_per_node_statement = off
    # Log all statements
    # with node and backend informations

log_client_messages = off
    # Log any client messages

log_standby_delay = 'if_over_threshold'
    # Log standby delay
    # Valid values are combinations of always,
    # if_over_threshold, none

# - Syslog specific -

syslog_facility = 'LOCAL0'
    # Syslog local facility. Default to LOCAL0

syslog_ident = 'pgpool'
    # Syslog program identification string
    # Default to 'pgpool'

# - Debug -

#log_error_verbosity = default      # terse, default, or verbose messages

#client_min_messages = notice      # values in order of decreasing detail:
    # debug5
    # debug4
    # debug3
    # debug2
    # debug1
    # log
    # notice
    # warning
```

```
        # error

#log_min_messages = warning      # values in order of decreasing detail:
        # debug5
        # debug4
        # debug3
        # debug2
        # debug1
        # info
        # notice
        # warning
        # error
        # log
        # fatal
        # panic

#-----
# FILE LOCATIONS
#-----

pid_file_name = '/var/run/pgpool-II-12/pgpool.pid'
        # PID file name
        # Can be specified as relative to the"
        # location of pgpool.conf file or
        # as an absolute path
        # (change requires restart)

logdir = '/tmp'
        # Directory of pgPool status file
        # (change requires restart)

#-----
# CONNECTION POOLING
#-----

connection_cache = on
        # Activate connection pools
        # (change requires restart)

        # Semicolon separated list of queries
        # to be issued at the end of a session
        # The default is for 8.3 and later
reset_query_list = 'ABORT; DISCARD ALL'
        # The following one is for 8.2 and before
#reset_query_list = 'ABORT; RESET ALL; SET SESSION AUTHORIZATION DEFAULT'
```

```
#-----  
# REPLICATION MODE  
#-----  
  
replication_mode = off  
    # Activate replication mode  
    # (change requires restart)  
replicate_select = off  
    # Replicate SELECT statements  
    # when in replication mode  
    # replicate_select is higher priority than  
    # load_balance_mode.  
  
insert_lock = off  
    # Automatically locks a dummy row or a table  
    # with INSERT statements to keep SERIAL data  
    # consistency  
    # Without SERIAL, no lock will be issued  
loobj_lock_table = "  
    # When rewriting lo_creat command in  
    # replication mode, specify table name to  
    # lock  
  
# - Degenerate handling -  
  
replication_stop_on_mismatch = off  
    # On disagreement with the packet kind  
    # sent from backend, degenerate the node  
    # which is most likely "minority"  
    # If off, just force to exit this session  
  
failover_if_affected_tuples_mismatch = off  
    # On disagreement with the number of affected  
    # tuples in UPDATE/DELETE queries, then  
    # degenerate the node which is most likely  
    # "minority".  
    # If off, just abort the transaction to  
    # keep the consistency  
  
#-----  
# LOAD BALANCING MODE  
#-----  
  
load_balance_mode = on  
    # Activate load balancing mode
```

```
# (change requires restart)
ignore_leading_white_space = on
    # Ignore leading white spaces of each query
white_function_list = "
    # Comma separated list of function names
    # that don't write to database
    # Regexp are accepted
black_function_list = 'currval,lastval,nextval,setval'
    # Comma separated list of function names
    # that write to database
    # Regexp are accepted

black_query_pattern_list = "
    # Semicolon separated list of query patterns
    # that should be sent to primary node
    # Regexp are accepted
    # valid for streaming replicaton mode only.

database_redirect_preference_list = "
    # comma separated list of pairs of database and node id.
    # example: postgres:primary,mydb[0-4]:1,mydb[5-9]:2'
    # valid for streaming replicaton mode only.

app_name_redirect_preference_list = "
    # comma separated list of pairs of app name and node id.
    # example: 'psql:primary,myapp[0-4]:1,myapp[5-9]:standby'
    # valid for streaming replicaton mode only.

allow_sql_comments = off
    # if on, ignore SQL comments when judging if load balance or
    # query cache is possible.
    # If off, SQL comments effectively prevent the judgment
    # (pre 3.4 behavior).

disable_load_balance_on_write = 'transaction'
    # Load balance behavior when write query is issued
    # in an explicit transaction.
    # Note that any query not in an explicit transaction
    # is not affected by the parameter.
    # 'transaction' (the default): if a write query is issued,
    # subsequent read queries will not be load balanced
    # until the transaction ends.
    # 'trans_transaction': if a write query is issued,
    # subsequent read queries in an explicit transaction
    # will not be load balanced until the session ends.
    # 'always': if a write query is issued, read queries will
    # not be load balanced until the session ends.
```

```
statement_level_load_balance = off
    # Enables statement level load balancing

#-----
# MASTER/SLAVE MODE
#-----

master_slave_mode = on
    # Activate master/slave mode
    # (change requires restart)

master_slave_sub_mode = 'stream'
    # Master/slave sub mode
    # Valid values are combinations stream, slony
    # or logical. Default is stream.
    # (change requires restart)

# - Streaming -

sr_check_period = 3
    # Streaming replication check period
    # Disabled (0) by default

sr_check_user = 'nobody'
    # Streaming replication check user
    # This is necessary even if you disable streaming
    # replication delay check by sr_check_period = 0

sr_check_password = ''
    # Password for streaming replication check user
    # Leaving it empty will make Pgpool-II to first look for the
    # Password in pool_passwd file before using the empty password

sr_check_database = 'postgres'
    # Database name for streaming replication check

delay_threshold = 512000
    # Threshold before not dispatching query to standby node
    # Unit is in bytes
    # Disabled (0) by default

# - Special commands -

follow_master_command = ''
    # Executes this command after master failover
    # Special values:
    # %d = failed node id
    # %h = failed node host name
    # %p = failed node port number
    # %D = failed node database cluster path
    # %m = new master node id
```

```

# %I = new master node id
# %H = new master node hostname
# %M = old master node id
# %P = old primary node id
# %r = new master port number
# %R = new master database cluster path
# %N = old primary node hostname
# %S = old primary node port number
# %% = '%' character

#-----
# HEALTH CHECK GLOBAL PARAMETERS
#-----

health_check_period = 5
    # Health check period
    # Disabled (0) by default

health_check_timeout = 10
    # Health check timeout
    # 0 means no timeout

health_check_user = 'nobody'
    # Health check user

health_check_password = "
    # Password for health check user
    # Leaving it empty will make Pgpool-II to first look for the
    # Password in pool_passwd file before using the empty password

health_check_database = "
    # Database name for health check. If "", tries 'postgres' first,

health_check_max_retries = 60
    # Maximum number of times to retry a failed health check before giving up.

health_check_retry_delay = 1
    # Amount of time to wait (in seconds) between retries.

connect_timeout = 10000
    # Timeout value in milliseconds before giving up to connect to backend.
    # Default is 10000 ms (10 second). Flaky network user may want to increase
    # the value. 0 means no timeout.
    # Note that this value is not only used for health check,
    # but also for ordinary connection to backend.

#-----
# HEALTH CHECK PER NODE PARAMETERS (OPTIONAL)
#-----

#health_check_period0 = 0
#health_check_timeout0 = 20
#health_check_user0 = 'nobody'
#health_check_password0 = "

```

```

#health_check_database0 = "
#health_check_max_retries0 = 0
#health_check_retry_delay0 = 1
#connect_timeout0 = 10000

#-----
# FAILOVER AND FAILBACK
#-----

failover_command = "
    # Executes this command at failover
    # Special values:
    # %d = failed node id
    # %h = failed node host name
    # %p = failed node port number
    # %D = failed node database cluster path
    # %m = new master node id
    # %H = new master node hostname
    # %M = old master node id
    # %P = old primary node id
    # %r = new master port number
    # %R = new master database cluster path
    # %N = old primary node hostname
    # %S = old primary node port number
    # %% = '%' character
failback_command = "
    # Executes this command at failback.
    # Special values:
    # %d = failed node id
    # %h = failed node host name
    # %p = failed node port number
    # %D = failed node database cluster path
    # %m = new master node id
    # %H = new master node hostname
    # %M = old master node id
    # %P = old primary node id
    # %r = new master port number
    # %R = new master database cluster path
    # %N = old primary node hostname
    # %S = old primary node port number
    # %% = '%' character

failover_on_backend_error = off
    # Initiates failover when reading/writing to the
    # backend communication socket fails
    # If set to off, pgpool will report an
    # error and disconnect the session.

```

```
detach_false_primary = off
    # Detach false primary if on. Only
    # valid in streaming replicaton
    # mode and with PostgreSQL 9.6 or
    # after.

search_primary_node_timeout = 300
    # Timeout in seconds to search for the
    # primary node when a failover occurs.
    # 0 means no timeout, keep searching
    # for a primary node forever.

#-----
# ONLINE RECOVERY
#-----

recovery_user = 'nobody'
    # Online recovery user
recovery_password = ""
    # Online recovery password
    # Leaving it empty will make Pgpool-II to first look for the
    # Password in pool_passwd file before using the empty password

recovery_1st_stage_command = ""
    # Executes a command in first stage
recovery_2nd_stage_command = ""
    # Executes a command in second stage
recovery_timeout = 90
    # Timeout in seconds to wait for the
    # recovering node's postmaster to start up
    # 0 means no wait

client_idle_limit_in_recovery = 0
    # Client is disconnected after being idle
    # for that many seconds in the second stage
    # of online recovery
    # 0 means no disconnection
    # -1 means immediate disconnection

auto_failback = off
    # Dettached backend node reattach automatically
    # if replication_state is 'streaming'.

auto_failback_interval = 60
    # Min interval of executing auto_failback in
    # seconds.

#-----
```

```
#-----  
# WATCHDOG  
#-----  
  
# - Enabling -  
  
use_watchdog = off  
    # Activates watchdog  
    # (change requires restart)  
  
# -Connection to up stream servers -  
  
trusted_servers = "  
    # trusted server list which are used  
    # to confirm network connection  
    # (hostA,hostB,hostC,...)  
    # (change requires restart)  
ping_path = '/bin'  
    # ping command path  
    # (change requires restart)  
  
# - Watchdog communication Settings -  
  
wd_hostname = "  
    # Host name or IP address of this watchdog  
    # (change requires restart)  
wd_port = 9000  
    # port number for watchdog service  
    # (change requires restart)  
wd_priority = 1  
    # priority of this watchdog in leader election  
    # (change requires restart)  
  
wd_authkey = "  
    # Authentication key for watchdog communication  
    # (change requires restart)  
  
wd_ipc_socket_dir = '/tmp'  
    # Unix domain socket path for watchdog IPC socket  
    # The Debian package defaults to  
    # /var/run/postgresql  
    # (change requires restart)  
  
# - Virtual IP control Setting -  
  
delegate_IP = "
```

```
# delegate IP address
# If this is empty, virtual IP never bring up.
# (change requires restart)
if_cmd_path = '/sbin'
# path to the directory where if_up/down_cmd exists
# If if_up/down_cmd starts with "/", if_cmd_path will be ignored.
# (change requires restart)
if_up_cmd = '/usr/bin/sudo /sbin/ip addr add $_IP_$ /24 dev eth0 label eth0:0'
# startup delegate IP command
# (change requires restart)
if_down_cmd = '/usr/bin/sudo /sbin/ip addr del $_IP_$ /24 dev eth0'
# shutdown delegate IP command
# (change requires restart)
arping_path = '/usr/sbin'
# arping command path
# If arping_cmd starts with "/", if_cmd_path will be ignored.
# (change requires restart)
arping_cmd = '/usr/bin/sudo /usr/sbin/arping -U $_IP_$ -w 1 -I eth0'
# arping command
# (change requires restart)

# - Behavior on escalation Setting -

clear_memqcache_on_escalation = on
# Clear all the query cache on shared memory
# when standby pgpool escalate to active pgpool
# (= virtual IP holder).
# This should be off if client connects to pgpool
# not using virtual IP.
# (change requires restart)
wd_escalation_command = "
# Executes this command at escalation on new active pgpool.
# (change requires restart)
wd_de_escalation_command = "
# Executes this command when master pgpool resigns from being master.
# (change requires restart)

# - Watchdog consensus settings for failover -

failover_when_quorum_exists = on
# Only perform backend node failover
# when the watchdog cluster holds the quorum
# (change requires restart)

failover_require_consensus = on
# Perform failover when majority of Pgpool-II nodes
# agrees on the backend node status change
```

```
# (change requires restart)

allow_multiple_failover_requests_from_node = off
    # A Pgpool-II node can cast multiple votes
    # for building the consensus on failover
    # (change requires restart)

enable_consensus_with_half_votes = off
    # apply majority rule for consensus and quorum computation
    # at 50% of votes in a cluster with even number of nodes.
    # when enabled the existence of quorum and consensus
    # on failover is resolved after receiving half of the
    # total votes in the cluster, otherwise both these
    # decisions require at least one more vote than
    # half of the total votes.
    # (change requires restart)

# - Lifecheck Setting -

# -- common --

wd_monitoring_interfaces_list = " # Comma separated list of interfaces names to monitor.
    # if any interface from the list is active the watchdog will
    # consider the network is fine
    # 'any' to enable monitoring on all interfaces except loopback
    # '' to disable monitoring
    # (change requires restart)

wd_lifecheck_method = 'heartbeat'
    # Method of watchdog lifecheck ('heartbeat' or 'query' or 'external')
    # (change requires restart)

wd_interval = 10
    # lifecheck interval (sec) > 0
    # (change requires restart)

# -- heartbeat mode --

wd_heartbeat_port = 9694
    # Port number for receiving heartbeat signal
    # (change requires restart)

wd_heartbeat_keepalive = 2
    # Interval time of sending heartbeat signal (sec)
    # (change requires restart)

wd_heartbeat_deadtime = 30
    # Deadtime interval for heartbeat signal (sec)
    # (change requires restart)
```

```
# (change requires restart)
heartbeat_destination0 = 'host0_ip1'
    # Host name or IP address of destination 0
    # for sending heartbeat signal.
    # (change requires restart)
heartbeat_destination_port0 = 9694
    # Port number of destination 0 for sending
    # heartbeat signal. Usually this is the
    # same as wd_heartbeat_port.
    # (change requires restart)
heartbeat_device0 = ""
    # Name of NIC device (such like 'eth0')
    # used for sending/receiving heartbeat
    # signal to/from destination 0.
    # This works only when this is not empty
    # and pgsu has root privilege.
    # (change requires restart)

#heartbeat_destination1 = 'host0_ip2'
#heartbeat_destination_port1 = 9694
#heartbeat_device1 = ""

# -- query mode --

wd_life_point = 3
    # lifecheck retry times
    # (change requires restart)
wd_lifecheck_query = 'SELECT 1'
    # lifecheck query to pgsu from watchdog
    # (change requires restart)
wd_lifecheck_dbname = 'template1'
    # Database name connected for lifecheck
    # (change requires restart)
wd_lifecheck_user = 'nobody'
    # watchdog user monitoring pgsus in lifecheck
    # (change requires restart)
wd_lifecheck_password = ""
    # Password for watchdog user in lifecheck
    # Leaving it empty will make Pgsu-II to first look for the
    # Password in pool_passwd file before using the empty password
    # (change requires restart)

# - Other pgsu Connection Settings -

#other_pgsu_hostname0 = 'host0'
    # Host name or IP address to connect to for other pgsu 0
    # (change requires restart)
```

```
#other_pgpool_port0 = 5432
    # Port number for other pgpool 0
    # (change requires restart)
#other_wd_port0 = 9000
    # Port number for other watchdog 0
    # (change requires restart)
#other_pgpool_hostname1 = 'host1'
#other_pgpool_port1 = 5432
#other_wd_port1 = 9000

#-----
# OTHERS
#-----

relcache_expire = 0
    # Life time of relation cache in seconds.
    # 0 means no cache expiration(the default).
    # The relation cache is used for cache the
    # query result against PostgreSQL system
    # catalog to obtain various information
    # including table structures or if it's a
    # temporary table or not. The cache is
    # maintained in a pgpool child local memory
    # and being kept as long as it survives.
    # If someone modify the table by using
    # ALTER TABLE or some such, the relcache is
    # not consistent anymore.
    # For this purpose, cache_expiration
    # controls the life time of the cache.

relcache_size = 8192
    # Number of relation cache
    # entry. If you see frequently:
    # "pool_search_relcache: cache replacement happend"
    # in the pgpool log, you might want to increate this number.

check_temp_table = catalog
    # Temporary table check method. catalog, trace or none.
    # Default is catalog.

check_unlogged_table = on
    # If on, enable unlogged table check in SELECT statements.
    # This initiates queries against system catalog of primary/master
    # thus increases load of master.
    # If you are absolutely sure that your system never uses unlogged tables
    # and you want to save access to primary/master, you could turn this off.
    # Default is on.

enable_shared_relcache = on
```

```

        # If on, relation cache stored in memory cache,
        # the cache is shared among child process.
        # Default is on.
        # (change requires restart)

relcache_query_target = master # Target node to send relcache queries. Default is master (primary) node.
        # If load_balance_node is specified, queries will be sent to load balance node.
#-----
# IN MEMORY QUERY MEMORY CACHE
#-----
memory_cache_enabled = off
        # If on, use the memory cache functionality, off by default
        # (change requires restart)

memqcache_method = 'shmem'
        # Cache storage method. either 'shmem'(shared memory) or
        # 'memcached'. 'shmem' by default
        # (change requires restart)

memqcache_memcached_host = 'localhost'
        # Memcached host name or IP address. Mandatory if
        # memqcache_method = 'memcached'.
        # Defaults to localhost.
        # (change requires restart)

memqcache_memcached_port = 11211
        # Memcached port number. Mandatory if memqcache_method = 'memcached'.
        # Defaults to 11211.
        # (change requires restart)

memqcache_total_size = 67108864
        # Total memory size in bytes for storing memory cache.
        # Mandatory if memqcache_method = 'shmem'.
        # Defaults to 64MB.
        # (change requires restart)

memqcache_max_num_cache = 1000000
        # Total number of cache entries. Mandatory
        # if memqcache_method = 'shmem'.
        # Each cache entry consumes 48 bytes on shared memory.
        # Defaults to 1,000,000(45.8MB).
        # (change requires restart)

memqcache_expire = 0
        # Memory cache entry life time specified in seconds.
        # 0 means infinite life time. 0 by default.
        # (change requires restart)

memqcache_auto_cache_invalidation = on
        # If on, invalidation of query cache is triggered by corresponding
        # DDL/DML/DCL(and memqcache_expire). If off, it is only triggered
        # by memqcache_expire. on by default.
        # (change requires restart)

```

```

memqcache_maxcache = 409600
    # Maximum SELECT result size in bytes.
    # Must be smaller than memqcache_cache_block_size. Defaults to 400KB.
    # (change requires restart)
memqcache_cache_block_size = 1048576
    # Cache block size in bytes. Mandatory if memqcache_method = 'shmem'.
    # Defaults to 1MB.
    # (change requires restart)
memqcache_oiddir = '/var/log/pgpool/oiddir'
    # Temporary work directory to record table oids
    # (change requires restart)
white_memqcache_table_list = ""
    # Comma separated list of table names to memcache
    # that don't write to database
    # Regexp are accepted
black_memqcache_table_list = ""
    # Comma separated list of table names not to memcache
    # that don't write to database
    # Regexp are accepted

```

You must reconfigure the following parameters:

```

listen_addresses = '0.0.0.0'
port = 8001
socket_dir = '/tmp'
reserved_connections = 0

pcp_listen_addresses = ""
pcp_port = 9898
pcp_socket_dir = '/tmp'

# - Backend Connection Settings -

backend_hostname0 = '127.0.0.1'
    # Host name or IP address to connect to for backend 0
backend_port0 = 3389
    # Port number for backend 0
backend_weight0 = 1
    # Weight for backend 0 (only in load balancing mode)
backend_data_directory0 = '/data01/pg12_3389/pg_root'
    # Data directory for backend 0
backend_flag0 = 'ALWAYS_MASTER'
    # Controls various backend behavior
    # ALLOW_TO_FAILOVER, DISALLOW_TO_FAILOVER
    # or ALWAYS_MASTER
backend_application_name0 = 'server0'
    # walsender's application_name, used for "show pool_nodes" command
backend_hostname1 = '127.0.0.1'

```

```
backend_port1 = 8002
backend_weight1 = 1
backend_data_directory1 = '/data01/pg12_8002/pg_root'
backend_flag1 = 'DISALLOW_TO_FAILOVER'
backend_application_name1 = 'server1'

# - Authentication -

enable_pool_hba = on

                # Use pool_hba.conf for client authentication
pool_passwd = 'pool_passwd'
                # File name of pool_passwd for md5 authentication.
                # "" disables pool_passwd.
                # (change requires restart)
allow_clear_text_frontend_auth = off
                # Allow Pgpool-II to use clear text password authentication
                # with clients, when pool_passwd does not
                # contain the user password

# - Concurrent session and pool size -

num_init_children = 128
                # Number of concurrent sessions allowed
                # (change requires restart)
max_pool = 4
                # Number of connection pool caches per connection
                # (change requires restart)

# - Life time -

child_life_time = 300
                # Pool exits after being idle for this many seconds
child_max_connections = 0
                # Pool exits after receiving that many connections
                # 0 means no exit
connection_life_time = 0
                # Connection to backend closes after being idle for this many seconds
                # 0 means no close
client_idle_limit = 0
                # Client is disconnected after being idle for that many seconds
                # (even inside an explicit transactions!)
                # 0 means no disconnection

#-----
# LOGS
#-----
```

```

# - Where to log -

log_destination = 'syslog'
    # Where to log
    # Valid values are combinations of stderr,
    # and syslog. Default to stderr.

log_connections = on
    # Log connections

log_standby_delay = 'if_over_threshold'
    # Log standby delay
    # Valid values are combinations of always,
    # if_over_threshold, none

#-----
# FILE LOCATIONS
#-----

pid_file_name = '/var/run/pgpool-II-12/pgpool.pid'
    # PID file name
    # Can be specified as relative to the"
    # location of pgpool.conf file or
    # as an absolute path
    # (change requires restart)

logdir = '/tmp'
    # Directory of pgPool status file
    # (change requires restart)

#-----
# CONNECTION POOLING
#-----

connection_cache = on
    # Activate connection pools
    # (change requires restart)

    # Semicolon separated list of queries
    # to be issued at the end of a session
    # The default is for 8.3 and later

reset_query_list = 'ABORT; DISCARD ALL'

#-----
# LOAD BALANCING MODE
#-----

```

```
load_balance_mode = on
    # Activate load balancing mode
    # (change requires restart)
ignore_leading_white_space = on
    # Ignore leading white spaces of each query
white_function_list = "
    # Comma separated list of function names
    # that don't write to database
    # Regexp are accepted
black_function_list = 'currval,lastval,nextval,setval'
    # Comma separated list of function names
    # that write to database
    # Regexp are accepted

black_query_pattern_list = "
    # Semicolon separated list of query patterns
    # that should be sent to primary node
    # Regexp are accepted
    # valid for streaming replicaton mode only.

database_redirect_preference_list = "
    # comma separated list of pairs of database and node id.
    # example: postgres:primary,mydb[0-4]:1,mydb[5-9]:2'
    # valid for streaming replicaton mode only.

app_name_redirect_preference_list = "
    # comma separated list of pairs of app name and node id.
    # example: 'psql:primary,myapp[0-4]:1,myapp[5-9]:standby'
    # valid for streaming replicaton mode only.

allow_sql_comments = off
    # if on, ignore SQL comments when judging if load balance or
    # query cache is possible.
    # If off, SQL comments effectively prevent the judgment
    # (pre 3.4 behavior).

disable_load_balance_on_write = 'transaction'
    # Load balance behavior when write query is issued
    # in an explicit transaction.
    # Note that any query not in an explicit transaction
    # is not affected by the parameter.
    # 'transaction' (the default): if a write query is issued,
    # subsequent read queries will not be load balanced
    # until the transaction ends.
    # 'trans_transaction': if a write query is issued,
    # subsequent read queries in an explicit transaction
    # will not be load balanced until the session ends.
    # 'always': if a write query is issued, read queries will
```

```

        # not be load balanced until the session ends.

statement_level_load_balance = off
        # Enables statement level load balancing

#-----
# MASTER/SLAVE MODE
#-----

master_slave_mode = on
        # Activate master/slave mode
        # (change requires restart)
master_slave_sub_mode = 'stream'
        # Master/slave sub mode
        # Valid values are combinations stream, slony
        # or logical. Default is stream.
        # (change requires restart)

# - Streaming -

sr_check_period = 3
        # Streaming replication check period
        # Disabled (0) by default
sr_check_user = 'nobody'
        # Streaming replication check user
        # This is necessary even if you disable streaming
        # replication delay check by sr_check_period = 0
sr_check_password = ''
        # Password for streaming replication check user
        # Leaving it empty will make Pgpool-II to first look for the
        # Password in pool_passwd file before using the empty password

sr_check_database = 'postgres'
        # Database name for streaming replication check
delay_threshold = 512000
        # Threshold before not dispatching query to standby node
        # Unit is in bytes
        # Disabled (0) by default

#-----
# HEALTH CHECK GLOBAL PARAMETERS
#-----

health_check_period = 5
        # Health check period
        # Disabled (0) by default

```

```
# Disabled (0) by default
health_check_timeout = 10
    # Health check timeout
    # 0 means no timeout
health_check_user = 'nobody'
    # Health check user
health_check_password = ''
    # Password for health check user
    # Leaving it empty will make Pgpool-II to first look for the
    # Password in pool_passwd file before using the empty password

health_check_database = ''
    # Database name for health check. If "", tries 'postgres' first,
health_check_max_retries = 60
    # Maximum number of times to retry a failed health check before giving up.
health_check_retry_delay = 1
    # Amount of time to wait (in seconds) between retries.
connect_timeout = 10000
    # Timeout value in milliseconds before giving up to connect to backend.
    # Default is 10000 ms (10 second). Flaky network user may want to increase
    # the value. 0 means no timeout.
    # Note that this value is not only used for health check,
    # but also for ordinary connection to backend.

#-----
# FAILOVER AND FAILBACK
#-----

failover_on_backend_error = off
    # Initiates failover when reading/writing to the
    # backend communication socket fails
    # If set to off, pgpool will report an
    # error and disconnect the session.

relcache_expire = 0 # After the configuration file is restructured, we recommend that you set this parameter to 1,
reload the configuration file, and then set this parameter to 0 again. You can also set this parameter to a specific
point in time.
    # Life time of relation cache in seconds.
    # 0 means no cache expiration(the default).
    # The relation cache is used for cache the
    # query result against PostgreSQL system
    # catalog to obtain various information
    # including table structures or if it's a
    # temporary table or not. The cache is
    # maintained in a pgpool child local memory
    # and being kept as long as it survives.
```

```

# If someone modify the table by using
# ALTER TABLE or some such, the relcache is
# not consistent anymore.
# For this purpose, cache_expiration
# controls the life time of the cache.

relcache_size = 8192

# Number of relation cache
# entry. If you see frequently:
# "pool_search_relcache: cache replacement happend"
# in the pgpool log, you might want to increate this number.

```

3. Configure the pool_passwd file.

Note If you connect to your RDS instances by using Pgpool, you must configure the pool_passwd file. This is because Pgpool supports the authentication protocol of PostgreSQL.

```

cd /etc/pgpool-II-12

# Run the following command:
#pg_md5 --md5auth --username=username password
# Generate the passwords of the digoyal and nobody users. The passwords are automatically written into the pool
_passwd file.
pg_md5 --md5auth --username=digoal "xxxxxxx"
pg_md5 --md5auth --username=nobody "xxxxxxx"

```

4. Use the system to automatically generate the pool_passwd file.

```

cd /etc/pgpool-II-12
# cat pool_passwd
digoal:md54dd55116da69d3d03bf2e3a1470564f9
nobody:md54240e76623e2511d607f431043a5d1c1

```

5. Configure the pgpool_hba file.

```

cd /etc/pgpool-II-12
cp pool_hba.conf.sample pool_hba.conf
vi pool_hba.conf

host all all 0.0.0.0/0 md5

```

6. Configure the pcp.conf file.

Note The pcp.conf file is used to manage the users and passwords of Pgpool. It is not related to the users and passwords of your RDS instances.

```
cd /etc/pgpool-II-12

# pg_md5 abc # In this command, you set the password to abc and encrypt it by using the MD5 encryption algorithm.
900150983cd24fb0d6963f7d28e17f72

cp pcp.conf.sample pcp.conf

vi pcp.conf

# USERID:MD5PASSWD
manage:900150983cd24fb0d6963f7d28e17f72 # In this command, the manage user is used to manage PCP.
```

7. Start Pgpool.

```
cd /etc/pgpool-II-12
pgpool -f ./pgpool.conf -a ./pool_hba.conf -F ./pcp.conf
```

Note If you want to view the logs of Pgpool, run the following command:

```
less /var/log/messages
```

8. Use Pgpool to connect to your RDS instances.

```
psql -h 127.0.0.1 -p 8001 -U digoal postgres
```

```
[root@izbp1grrvgf11g1hmd8992Z pgpool-II-12]# psql -h pgm-bp1grrvgf11g1hmd8992Z.pg.rds.aliyuncs.com -p 3433 -U digoal postgres
Password for user digoal:
psql (12.2, server 10.10)
Type "help" for help.
postgres=>
```

FAQ

- **Q:** How do I test whether read/write splitting is enabled?
A: You can connect to your RDS instances by using Pgpool and call the `pg_is_in_recovery()` function. Then, close the connection, establish a connection again, and call the `pg_is_in_recovery()` function again. If you receive a value of false and then a value of true, it indicates that Pgpool routes requests to your primary RDS instance and then to your read-only RDS instances and that read/write splitting is enabled.
- **Q:** Does Pgpool increase the latency?
A: Pgpool increases the latency slightly. In the test environment you set up in this topic, the latency increases by about 0.12 milliseconds.
- **Q:** How does Pgpool check the latency and health on my read-only RDS instances?
 - **A:** If the WAL replay latency on a read-only RDS instance exceeds the specified limit, Pgpool stops routing SQL requests to the read-only instance. Pgpool resumes routing SQL requests to the read-only instance only after it detects that the WAL replay latency on the read-only instance falls below the specified limit.

Note Connect to your primary RDS instance and query the location where the current WAL data record is written. This location is referred to as log sequence number (LSN) 1. Then, connect to a read-only RDS instance and query the location where the current WAL data record is replayed. This location is referred to as LSN 2. You can obtain the number of bytes between LSN 1 and LSN 2. This number indicates the latency.

- Pgpool monitors the health of your read-only RDS instances. If a read-only instance is unhealthy, Pgpool stops routing requests to the read-only instance.

- **Q: How do I stop Pgpool and reload the configuration of Pgpool?**

A: Run the `pgpool --help` command to obtain more information about the commands used in Pgpool.

Example:

```
cd /etc/pgpool-II-12
pgpool -f ./pgpool.conf -m fast stop
```

- **Q: How do I configure Pgpool if more than one read-only RDS instance is attached to my primary RDS instance?**

A: Add the configurations of all the attached read-only RDS instances to the `pgpool.conf` file. **Example:**

```
backend_hostname1 = 'xx.xx.xxx.xx'
backend_port1 = 8002
backend_weight1 = 1
backend_data_directory1 = '/data01/pg12_8002/pg_root'
backend_flag1 = 'DISALLOW_TO_FAILOVER'
backend_application_name1 = 'server1'

backend_hostname2 = 'xx.xx.xx.xx'
backend_port1 = 8002
backend_weight1 = 1
backend_data_directory1 = '/data01/pg12_8002/pg_root'
backend_flag1 = 'DISALLOW_TO_FAILOVER'
backend_application_name1 = 'server1'
```

- **Q: How do I use `pcp` commands to view the status of my read-only RDS instances?**

A: To obtain the status of your read-only RDS instances by using `pcp` commands, run the following command:

```
# pcp_node_info -U manage -h /tmp -p 9898 -n 1 -v
Password: Enter the password.

Hostname      : 127.0.0.1
Port          : 8002
Status        : 2
Weight        : 0.500000
Status Name   : up
Role          : standby
Replication Delay : 0
Replication State :
Replication Sync State :
Last Status Change : 2020-02-29 00:20:29
```

- **Q: Which listening ports are used by Pgpool for read/write splitting?**

A: The following listening ports are used by Pgpool for read/write splitting:

- Primary RDS instance: Port 3389
- Secondary RDS instance: Port 8002

- Pgpool: Port 8001
- PCP: Port 9898

12.17. Use ShardingSphere to develop ApsaraDB RDS for PostgreSQL

ShardingSphere is an open source ecosystem that consists of a set of distributed database middleware solutions.

Prerequisites

All PostgreSQL versions used with ApsaraDB for RDS support ShardingSphere.

Context

ApsaraDB RDS for PostgreSQL supports database-integrated sharding plug-ins (such as Citus, Postgres-XC, and AntDB) and massively parallel processing (MPP) products. It also supports sharding middleware products that are similar to those widely used in MySQL, such as ShardingSphere.

ShardingSphere is suitable for services that run in databases with thorough, well-organized logical sharding. It offers the following features:

- Data sharding
 - Database sharding and table sharding
 - Read/write splitting
 - Sharding strategy customization
 - Decentralized distributed primary key
- Distributed transaction
 - Unified transaction API
 - XA transaction
 - BASE transaction
- Database orchestration
 - Dynamic configuration
 - Orchestration and governance
 - Data encryption
 - Tracing and observability
 - Elastic scaling out (planning)

For more information, visit the [ShardingSphere documentation](#).

ShardingSphere products

ShardingSphere includes three independent products. You can choose the product that best suits your business requirements. The following table describes these products.

Parameter	Sharding-JDBC	Sharding-Proxy	Sharding-Sidecar
Supported database engine	All JDBC-compatible database engines such as MySQL, PostgreSQL, Oracle, and SQL Server	MySQL and PostgreSQL	MySQL and PostgreSQL
Connections consumed	High	Low	High
Supported heterogeneous language	Java	All	All

Parameter	Sharding-JDBC	Sharding-Proxy	Sharding-Sidecar
Performance	Low consumption	Moderate consumption	Low consumption
Decentralized	Yes	No	Yes
Stateless API	No	Yes	No

Prepare configuration templates

1. On your ECS instance, run the following commands to go to the directory where configuration templates are stored:

```
cd apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin
cd conf
```

2. Run the following command to view all files stored in the directory:

```
# ll
total 24
-rw-r--r-- 1 501 games 3019 Jul 30 2019 config-encrypt.yaml
-rw-r--r-- 1 501 games 3582 Apr 22 2019 config-master_slave.yaml
-rw-r--r-- 1 501 games 4278 Apr 22 2019 config-sharding.yaml
-rw-r--r-- 1 501 games 1918 Jul 30 2019 server.yaml
```

Note

- config-encrypt.yaml: the data encryption configuration file.
- config-master_slave.yaml: the read/write splitting configuration file.
- config-sharding.yaml: the data sharding configuration file.
- server.yaml: the common configuration file.

3. Modify the configuration files.

 **Note** For more information about the configuration files, visit the [ShardingSphere documentation](#). In this example, the data sharding and common configuration files are used.

- Example of a data sharding configuration file:

schemaName: # The name of the logical data source.

dataSources: # The configuration of the data source. You can configure more than one data source by using the data_source_name variable.

<data_source_name>: # You do not need to configure a database connection pool. This is different in Sharding-JDBC.

url: # The URL used to connect to your database.

username: # The username used to log on to the database.

password: # The password used to log on to the database.

connectionTimeoutMilliseconds: 30000 # The connection timeout duration in milliseconds.

idleTimeoutMilliseconds: 60000 # The idle-connection reclaiming timeout duration in milliseconds.

maxLifetimeMilliseconds: 1800000 # The maximum connection time to live (TTL) in milliseconds.

maxPoolSize: 65 # The maximum number of connections allowed.

shardingRule: # You do not need to configure a sharding rule, because it is the same in Sharding-JDBC.

- Example of a common configuration file:

Proxy properties

You do not need to configure proxy properties that are the same in Sharding-JDBC.

props:

acceptor.size: # The number of worker threads that receive requests from the client. The default number is equal to the number of CPU cores multiplied by 2.

proxy.transaction.type: # The type of transaction processed by the proxy. Valid values: LOCAL | XA | BASE. Default value: LOCAL. Value XA specifies to use Atomikos as the transaction manager. Value BASE specifies to copy the .jar package that implements the ShardingTransactionManager API to the lib directory.

proxy.opentracing.enabled: # Specifies whether to enable link tracing. Link tracing is disabled by default.

check.table.metadata.enabled: # Specifies whether to check the consistency of metadata among sharding tables during startup. Default value: false.

proxy.frontend.flush.threshold: # The number of packets returned in a batch during a complex query.

Permission verification

This part of the configuration is used to verify your permissions when you attempt to log on to Sharding-Proxy. After you configure the username, password, and authorized databases, you must use the correct username and password to log on to Sharding-Proxy from the authorized databases.

authentication:

users:

root: # The username of the root user.

password: root# The password of the root user.

sharding: # The username of the sharding user.

password: sharding# The password of the sharding user.

authorizedSchemas: sharding_db, masterslave_db # The databases in which the specified user is authorized. If you want to specify more than one database, separate them with commas (.). You are granted the permissions of the root user by default. This means that you can access all databases.

Set up a test environment

- On your ECS instance, install Java.

```
yum install -y java
```

- Configure an ApsaraDB for RDS instance that runs PostgreSQL 10.
 - Create an account with username r1.
 - Set the password of the account to "PW123321!".
 - Creates the following databases whose owners are user r1: db0, db1, db2, and db3.
 - Add the IP address of your ECS instance to an IP address whitelist of the ApsaraDB RDS for PostgreSQL instance.

 Note

- For more information about how to create an ApsaraDB RDS for PostgreSQL instance, database, and account, see [Create an instance](#) and [Create a database and an account](#).
- For information about how to configure an IP address whitelist, see [Configure an IP address whitelist](#).

- Configure the common configuration file.

```
vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/server.yaml

authentication:
  users:
    r1:
      password: PW123321!
      authorizedSchemas: db0,db1,db2,db3
  props:
    executor.size: 16
    sql.show: false
```

Test horizontal sharding

1. Modify the data sharding configuration file.

```
vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/config-sharding.yaml

schemaName: sdb

dataSources:
  db0:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db0
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
  db1:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db1
    username: r1
    password: PW123321!
    connectionTimeoutMilliseconds: 30000
    idleTimeoutMilliseconds: 60000
    maxLifetimeMilliseconds: 1800000
    maxPoolSize: 65
  db2:
    url: jdbc:postgresql://pgm-bpxxxxx.pg.rds.aliyuncs.com:1433/db2
    username: r1
    password: PW123321!
```

```
connectionTimeoutMilliseconds: 30000
idleTimeoutMilliseconds: 60000
maxLifetimeMilliseconds: 1800000
maxPoolSize: 65
db3:
url: jdbc:postgresql://pgm-bpxxxx.pg.rds.aliyuncs.com:1433/db3
username: r1
password: PW123321!
connectionTimeoutMilliseconds: 30000
idleTimeoutMilliseconds: 60000
maxLifetimeMilliseconds: 1800000
maxPoolSize: 65

shardingRule:
tables:
t_order:
actualDataNodes: db${0..3}.t_order${0..7}
databaseStrategy:
inline:
shardingColumn: user_id
algorithmExpression: db${user_id % 4}
tableStrategy:
inline:
shardingColumn: order_id
algorithmExpression: t_order${order_id % 8}
keyGenerator:
type: SNOWFLAKE
column: order_id
t_order_item:
actualDataNodes: db${0..3}.t_order_item${0..7}
databaseStrategy:
inline:
shardingColumn: user_id
algorithmExpression: db${user_id % 4}
tableStrategy:
inline:
shardingColumn: order_id
algorithmExpression: t_order_item${order_id % 8}
keyGenerator:
type: SNOWFLAKE
column: order_item_id
bindingTables:
- t_order,t_order_item
defaultTableStrategy:
none:
```

2. Start ShardingSphere and listen to Port 8001.

```
cd ~/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/bin/  
./start.sh 8001
```

3. Connect to the destination database.

```
psql -h 127.0.0.1 -p 8001 -U r1 sdb
```

4. Create a table.

```
create table t_order(order_id int8 primary key, user_id int8, info text, c1 int, crt_time timestamp);  
create table t_order_item(order_item_id int8 primary key, order_id int8, user_id int8, info text, c1 int, c2 int, c3 int, c  
4 int, c5 int, crt_time timestamp);
```

 **Note** When you create a table, the system automatically creates horizontal shards in the destination database based on the sharding strategy that you specify.

FAQ

- If you want to know the SQL parsing and routing statements used in ShardingSphere, run the following command:

```
vi /root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/conf/server.yaml  
  
authentication:  
  users:  
    r1:  
      password: PW123321!  
      authorizedSchemas: db0,db1,db2,db3  
  props:  
    executor.size: 16  
    sql.show: true # Specifies to log parsed SQL statements.
```

- If you want to test writes and queries, run the following commands:

```
sdb=> insert into t_order (user_id, info, c1, crt_time) values (0,'a',1,now());
```

```
sdb=> insert into t_order (user_id, info, c1, crt_time) values (1,'b',2,now());
```

```
sdb=> insert into t_order (user_id, info, c1, crt_time) values (2,'c',3,now());
```

```
sdb=> insert into t_order (user_id, info, c1, crt_time) values (3,'c',4,now());
```

```
sdb=> select * from t_order;
```

order_id	user_id	info	c1	crt_time
433352561047633921	0	a	1	2020-02-09 19:48:21.856555
433352585668198400	1	b	2	2020-02-09 19:48:27.726815
433352610813050881	2	c	3	2020-02-09 19:48:33.721754
433352628370407424	3	c	4	2020-02-09 19:48:37.907683

(4 rows)

```
sdb=> select * from t_order where user_id=1;
```

order_id	user_id	info	c1	crt_time
433352585668198400	1	b	2	2020-02-09 19:48:27.726815

(1 row)

- If you want to view ShardingSphere logs, find the logs from the following path:

```
/root/apache-shardingsphere-incubating-4.0.0-sharding-proxy-bin/logs/stdout.log
```

- If you want to use pgbench for stress testing, run the following commands:

```
vi test.sql
\set user_id random(1,100000000)
\set order_id random(1,200000000)
\set order_item_id random(1,200000000)
insert into t_order (user_id, order_id, info, c1 , crt_time) values (:user_id, :order_id,random()::text, random()*1000, now()) on conflict (order_id) do update set info=excluded.info,c1=excluded.c1,crt_time=excluded.crt_time;
insert into t_order_item (order_item_id, user_id, order_id, info, c1,c2,c3,c4,c5,crt_time) values (:order_item_id, :user_id, :order_id,random()::text, random()*1000,random()*1000,random()*1000,random()*1000,random()*1000, now()) on conflict(order_item_id) do nothing;

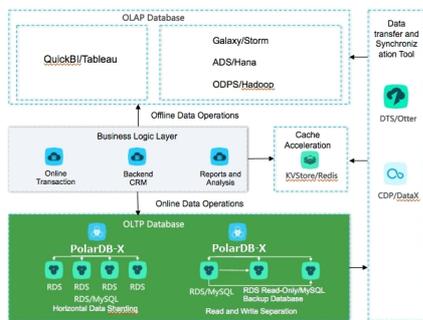
pgbench -M simple -n -r -P 1 -f ./test.sql -c 24 -j 24 -h 127.0.0.1 -p 8001 -U r1 sdb -T 120
progress: 1.0 s, 1100.9 tps, lat 21.266 ms stddev 6.349
progress: 2.0 s, 1253.0 tps, lat 18.779 ms stddev 7.913
progress: 3.0 s, 1219.0 tps, lat 20.083 ms stddev 13.212
```

13. Cloud Native Distributed Database PolarDB-X

13.1. 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). 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. PolarDB-X features lightweight (stateless), flexibility, stability, and high efficiency, 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. By partitioning data in specific business scenarios, PolarDB-X maximizes the operation efficiency, meeting the requirements of online businesses on relational databases.



Problems solved

- **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. Distributed solutions use multiple instances to work jointly, effectively resolving 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 through smooth data migration, supporting the dynamic scale-out of relational databases.

13.2. Quick start

This topic describes how to get started with Cloud Native Distributed Database PolarDB-X.

A PolarDB-X instance is physically a distributed cluster that consists of multiple PolarDB-X server nodes and underlying storage instances. A PolarDB-X database is a logical concept and only contains metadata. Specific data is stored in the physical database of the underlying storage instance. To get started with PolarDB-X, follow these steps:

1. **Create a PolarDB-X instance.**
2. **Create a database.**

To create a database in a PolarDB-X instance, you must select one or more ApsaraDB RDS for MySQL instances as the data storage nodes. If no RDS instance exists, create one first. For more information about how to create and manage ApsaraDB RDS for MySQL instances, see *User Guide of RDS*.

3. After a PolarDB-X database is created, you also need to create tables in the PolarDB-X database like in a single-instance database. However, the syntax is different, mainly in the expression of data partitioning information in the PolarDB-X table creation statement. For more information about how to create a table, see [Table creation syntax](#).

13.3. Log on to the PolarDB-X console

This topic describes how to log on to the Cloud Native Distributed Database PolarDB-X console by using Google Chrome.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Distributed Relational Database Service**.

13.4. Instance management

13.4.1. Create a PolarDB-X instance

To use PolarDB-X, you must first create an instance. This topic describes how to create a PolarDB-X instance.

1. [Log on to the PolarDB-X console](#).
2. On the page that appears, click **Create Instance** in the upper-right corner.
3. On the **Create DRDS Instance** page, set parameters as required.

[Parameters for creating a PolarDB-X instance](#) describes the parameters.

Parameters for creating a PolarDB-X instance

Type	Parameter	Description
Region	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
	Region	The region where the instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The zone where the instance resides.

Type	Parameter	Description
Basic Settings	Instance Type	The type of the instance. Select an instance type from the options available on the page.
	Instance Edition	The edition of the instance. Valid values: <ul style="list-style-type: none"> ◦ Standard ◦ Enterprise ◦ Starter
	Instance Specifications	The specifications of the instance. The rules vary with instance editions. Select the instance specifications from the options available on the page.
Network Type	Network Type	<p>The network type of the instance. PolarDB-X instances support the following network types:</p> <ul style="list-style-type: none"> ◦ Classic Network: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ VPC: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize routing tables, IP address ranges, and gateways in a VPC. We recommend that you select VPC for higher security. Select VPC for Network Type, and then set VPC and VSwitch. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Make sure that the PolarDB-X instance has the same network type as the Elastic Compute Service (ECS) instance to which you want to connect. If the PolarDB-X and ECS instances have different network types, they cannot communicate over an internal network.</p> </div>

4. Click **Submit**.

After the instance is created, it appears in the instance list and its status changes to **Running**. An instance name uniquely identifies a PolarDB-X instance.

13.4.2. Change specifications

When you use PolarDB-X, you can change the specifications of a PolarDB-X instance as needed.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the Basic Information page, click **Upgrade** or **Downgrade** in the **Common Operations** section to access the **Change Specifications** page.

 **Note** Alternatively, on the **DRDS Instance Management** page, choose **More > Downgrade** from the Actions column of the target instance.

5. On the **Change Specifications** page, set **Instance Edition** and **Instance Specifications**, and then click **Submit**. After a few minutes, you can view the new specifications of the instance in the instance list.

 **Note** Specifications downgrade leads to transient disconnections between applications and PolarDB-X within a short period of time. Make sure that your applications can be automatically reconnected.

13.4.3. Read-only PolarDB-X instances

13.4.3.1. Overview

Read-only PolarDB-X instances are extension and supplement to primary PolarDB-X instances and are compatible with SQL query syntax of primary PolarDB-X instances.

Features

Read-only and primary PolarDB-X instances can share the same replica of data. You can perform complex data query and analysis directly on read-only or primary ApsaraDB RDS for MySQL instances. Multiple instance types are provided to handle highly concurrent access requests and reduce the response time (RT) for complex queries. Resource isolation alleviates the load pressure on the primary instances and reduces the link complexity of the business architecture. It reduces the O&M and budget costs, eliminating the need for additional data synchronization.

Instance type

Concurrent read-only instances: For high-concurrency and high-traffic simple queries or offline data extraction, resource isolation protects you against highly concurrent queries, ensuring the stability of online business links.

 **Note** For the businesses with primary PolarDB-X instances, concurrent read-only instances can be used in the following scenarios:

- High-concurrency and high-traffic simple queries are performed.
- Data is extracted offline.

Limits

- Primary and read-only PolarDB-X instances must be in the same region, but they can be in different zones.
- A read-only PolarDB-X instance must belong to a primary PolarDB-X instance. Before creating a read-only instance, you must create a primary instance. After you create a database on the primary instance, the database is replicated to the read-only instance. If you delete the database from the primary instance, the corresponding database on the read-only instance is also deleted.
- You are not allowed to migrate data to read-only PolarDB-X instances.
- You are not allowed to create or delete databases in PolarDB-X read-only instances.
- PolarDB-X read-only instances cannot be cloned.
- PolarDB-X read-only instances support data definition language (DDL) statements but do not support data manipulation language (DML) statements for data modification.

13.4.3.3. Create a read-only PolarDB-X instance

This topic describes how to create a read-only Cloud Native Distributed Database PolarDB-X (PolarDB-X) instance.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.

4. On the Basic Information page, click **Create DRDS Read-only Instance** in the Related Instances section.
5. Set Region, Basic Settings, and Network Type, and then click **Submit**.

Parameters for creating a read-only PolarDB-X instance

Type	Parameter	Description
Region	Region	The region where the read-only instance resides. Services in different regions are not interconnected over the internal network. After the instance is created, the region cannot be changed.
	Zone	The zone where the read-only instance resides.
Basic Settings	Instance Type	The type of the read-only instance. Select an instance type from the options available on the page.
	Instance Edition	The edition of the read-only instance. Valid values: <ul style="list-style-type: none"> ◦ Starter ◦ Standard ◦ Enterprise
	Instance Specifications	The specifications of the read-only instance. The rules vary with instance editions. Select the instance specifications from the options available on the page.
	Description	The description of the read-only instance. We recommend that you provide an informative description to simplify future management operations.
Network Type	Network Type	<p>The network type of the read-only instance. PolarDB-X instances support the following network types:</p> <ul style="list-style-type: none"> ◦ Classic Network: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ VPC: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize routing tables, IP address ranges, and gateways in a VPC. We recommend that you select VPC for higher security. Select VPC for Network Type, and then set VPC and VSwitch. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Make sure that the PolarDB-X instance has the same network type as the Elastic Compute Service (ECS) instance to which you want to connect. If the PolarDB-X and ECS instances have different network types, they cannot communicate over an internal network.</p> </div>

6. It takes several minutes to create the instance. Please wait. After the instance is created, it appears in the instance list in the PolarDB-X console.

13.4.3.4. Manage a read-only PolarDB-X instance

Read-only PolarDB-X instances are managed in a similar way as primary instances. However, databases cannot be created or deleted on the read-only instance management page. Databases on read-only instances are created or deleted with databases on primary instances. In the PolarDB-X console, you can go to the read-only instance management page in two ways.

Manage a read-only PolarDB-X instance by its ID

1. [Log on to the PolarDB-X console](#).
2. On the **DRDS Instance Management** page, find the target read-only instance.

3. Click the target instance ID or choose **More > Manage** from the Actions column of the target read-only instance to access the **Basic Information** page.

Manage a read-only PolarDB-X instance by the ID of its primary instance

1. **Log on to the PolarDB-X console.**
2. On the **DRDS Instance Management** page, find the target primary instance.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the target primary instance to access the **Basic Information** page.
4. On the **Basic Information** page, move the pointer over the number of read-only instances in the **Related Instances** section to view the ID of the read-only PolarDB-X instance.
5. Click the ID of the target read-only PolarDB-X instance. The **Basic Information** page of the read-only instance appears.

13.4.3.5. Release a read-only PolarDB-X instance

If you no longer need a read-only PolarDB-X instance, you can release it.

Prerequisites

The read-only instance must be in the **Running** state.

Procedure

1. **Log on to the PolarDB-X console.**
2. Find the target instance in the instance list.
3. In the PolarDB-X instance list, find the target instance, and choose **More > Release** from the Actions column.

 **Notice** You cannot recover the PolarDB-X instances that have been released. Exercise caution when you perform this operation.

4. In the **Release DRDS Instance** dialog box, click **OK**.

13.4.4. Restart a PolarDB-X instance

This topic describes how to restart a PolarDB-X instance.

Prerequisites

The PolarDB-X instance must be in the **Running** state.

Procedure

1. **Log on to the PolarDB-X console.**
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. Click **Restart Instance** in the upper-right corner.
5. In the **Restart Instance** dialog box, click **OK**.

 **Notice** Restarting a PolarDB-X instance terminates all its connections. Make appropriate service arrangements before you restart a PolarDB-X instance. Exercise caution when you perform this operation.

13.4.5. Release a PolarDB-X instance

This topic describes how to release a running PolarDB-X instance in the PolarDB-X console.

Prerequisites

- All databases on the PolarDB-X instance have been deleted.
- The PolarDB-X instance must be in the Running state.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. In the PolarDB-X instance list, find the target instance, and choose **More > Release** from the Actions column.
4. In the Release DRDS Instance dialog box, click **OK**.

 **Warning** After the PolarDB-X instance is released, data is not deleted from its attached ApsaraDB RDS for MySQL instances. However, a released PolarDB-X instance cannot be restored. Exercise caution when you perform this operation.

13.4.6. Recover data

13.4.6.1. Backup and recovery

PolarDB-X allows you to back up data of instances and databases and recover them by using the backup data. Instances can be automatically and manually backed up in a quick or consistent manner. The instance recovery capability restores data to the new PolarDB-X and ApsaraDB RDS for MySQL instances based on the existing backup set.

Backup methods

For different scenarios, PolarDB-X provides quick backup and consistent backup, as well as corresponding recovery capabilities. The following table compares the two backup methods.

Method	Scenario	Advantage	Disadvantage
Quick backup	It applies to routine backup and recovery scenarios.	<ul style="list-style-type: none"> • It provides faster data backup and recovery. • It supports recovery at any time based on backup sets. • It supports all PolarDB-X instance versions. 	It ensures data consistency only at a single ApsaraDB RDS for MySQL instance but not global data consistency in database and table sharding scenarios.

Method	Scenario	Advantage	Disadvantage
Consistent backup	It applies to backup and recovery for online core transaction businesses and the financial industry with a high-consistency requirement.	It ensures global data consistency in database and table sharding scenarios.	<ul style="list-style-type: none"> • It provides slower backup and recovery. • It supports recovery based on backup sets but not at any time. • It is supported only for PolarDB-X 5.3.8 and later. • During the backup, distributed transactions are locked within seconds for PolarDB-X instances. During the locking process, SQL execution response time (RT) may vary by milliseconds. Therefore, we recommend that you perform consistent backup during off-peak hours.

Limits

- The PolarDB-X automatic backup policy is disabled by default. You must manually enable it.
- The log backup capability of PolarDB-X depends on underlying ApsaraDB RDS for MySQL instances. Therefore, the log backup policy configured in the PolarDB-X console is automatically synchronized to all underlying ApsaraDB RDS for MySQL instances. After the policy is configured, do not modify it in the ApsaraDB for RDS console.
- The backup and recovery feature of PolarDB-X depends on log backups. We recommend that you enable the log backup policy by default to avoid invalid backup sets.
- Data definition language (DDL) operations cannot be performed during the backup process.
- During the backup, the underlying ApsaraDB RDS for MySQL instances of the PolarDB-X instance must be normal to avoid a backup failure.
- Consistent backup and recovery is only supported by PolarDB-X 5.3.8 and later versions.
- All tables must have primary keys to ensure data accuracy during data backup and recovery.
- During consistent backup, distributed transactions are locked within seconds for PolarDB-X instances. During the locking process, the execution of non-transactional SQL statements and single-instance transactions are not affected, but the committing of distributed transactions is blocked and the SQL execution RT may vary by milliseconds. We recommend that you perform consistent backup during off-peak hours.
- Due to the inventory of PolarDB-X and ApsaraDB RDS for MySQL, PolarDB-X automatically adjusts the instance type and zone during instance recovery. We recommend that you confirm and adjust the instance type and zone after recovery.

13.4.6.2. Configure an automatic backup policy

PolarDB-X provides the automatic backup feature. This topic describes how to configure an automatic backup policy.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the

Basic Information page.

4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, choose **Backup Policy > Edit**.
6. In the **Backup Policy** dialog box, set parameters as needed, and click **OK**.

13.4.6.3. Configure local logs

You can use local logs with the backup and recovery feature or the SQL flashback feature of PolarDB-X to accurately recover an instance or a database to the desired time point. This topic describes how to configure local logs.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, click the **Local Log Settings** tab and then click **Edit**.
6. In the **Local Binlog Settings** dialog box, set parameters as needed, and click **OK**.

 **Notice** The local log settings are applied to all underlying ApsaraDB RDS for MySQL instances.

13.4.6.4. Manual backup

PolarDB-X also provides the manual backup capability, so that you can back up data at any time. This topic describes how to manually back up instances and databases.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, click **Data Backup** on the right.
6. In the dialog box that appears, set **Backup Method** and **Backup Level**.
 - **Backup Method** can be set to **Fast Backup** and **Consistent Backup**. For more information about differences between the two methods, see [Backup methods](#).

 **Notice** If you select **Consistent Backup**, distributed transactions are locked within seconds and the response time (RT) may vary by sub-seconds. Therefore, we recommend that you perform this operation during off-peak hours.

- **Backup Level** can be set to **Instance Backup** or **Database Backup**. You can select **Instance Backup** to back up the entire instance, or select **Database Backup** to back up a database as needed.
7. Click **OK**.

13.4.6.5. Recover data

You can use the data recovery feature of PolarDB-X to recover an instance or a database to the time when the backup is created. You can perform this operation at any time. This topic describes how to recover the data of an instance or a database to a specific point in time.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Backup and Recovery**.
5. On the page that appears, click **Data Recovery (Original Clone Instance)** on the right.
6. Select a recovery method
 - **By Time:** Recover data to the selected point in time. You must set **Restoration Time** and **Recovery Level**.
 - **By Backup Set:** Recover data from the selected backup file.

 **Note** You can also click **Recover** in the Actions column of the target backup set to recover data by backup set

7. Click **Precheck** to check whether a valid backup set is available for data recovery. If the precheck fails, the data cannot be restored.
8. Click **Enable** to access the order confirmation page.
9. Confirm the order details and then click **Enable** to recover the data. You can view the data recovery progress in **Task Progress** in the upper-right corner of the page.

13.4.6.6. SQL flashback

13.4.6.6.1. Overview

PolarDB-X provides the SQL flashback feature to recover data of particular rows.

When you mistakenly run an SQL statement such as **INSERT**, **UPDATE**, or **DELETE** on PolarDB-X, provide the relevant SQL information to match the event in the binary log file and generate the corresponding recovery file. You can download the file and recover data as needed. SQL flashback automatically chooses **fuzzy match** or **exact match** to locate lost data caused by the error. For more information, see [Exact match and fuzzy match](#) and [Rollback SQL statements and original SQL statements](#).

Features

- **Easy-to-use:** SQL flashback allows you to retrieve the lost data by entering required information about the corresponding SQL statement.
- **Fast and lightweight:** Regardless of the backup policy of ApsaraDB RDS for MySQL instances, you only need to enable log backup before an SQL statement error occurs.
- **Flexible recovery:** Rollback SQL statements and original SQL statements are available for different scenarios.
- **Exact match:** SQL flashback supports exact match of data about the corresponding SQL statement, which improves precision of data recovery.

Limits

- SQL flashback depends on the binary log retention time and the log backup feature of ApsaraDB RDS for MySQL must be enabled. Binary log files can be retained only for a certain period. Use SQL flashback to generate files for recovery as soon as possible when an error occurs.
- The recovery files generated by SQL flashback are retained for seven days by default, and you need to download these files as soon as possible.
- The following conditions must be met for SQL flashback exact match:
 - The PolarDB-X instance version is 5.3.4-15378085 or later.
 - The version of the ApsaraDB RDS for MySQL instance used by the PolarDB-X database is 5.6 or later.
 - SQL flashback exact match is enabled before the error SQL statement is executed.

- The TRACE_ID information for the error SQL statement is provided.
- To ensure the precision of data recovery, the exact match feature is enabled by default for the database created in a PolarDB-X instance of 5.3.4-15378085 or later. After this feature is enabled, SQL execution information is included in the binary log file by default, which requires more storage space for ApsaraDB RDS for MySQL instances. If you need to use the exact match feature, we recommend that you upgrade PolarDB-X before enabling the feature. For more information, see [Enable exact match](#).

13.4.6.6.2. Generate a recovery file

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > SQL Flashback**. The **SQL Flashback** page appears.
5. On the **SQL Flashback** page, enter the basic information about a mistaken SQL statement, including Database, Time Range, Table Name, TRACE_ID, and SQL Statement Type. The following table describes the parameters.

Parameter	Description
Database	The database where the mistaken SQL statement was executed.
Time Range	The time range during which the mistaken SQL statement was executed. The start time is earlier than the start time when the mistaken SQL statement was executed, whereas the end time is later than the time when the execution of the mistaken SQL statement ended. To ensure efficient recovery, we recommend that you limit the time range to five minutes.
Table Name	The name of the table on which the mistaken SQL statement was executed. This parameter is optional.
TRACE_ID	The unique TRACE_ID that PolarDB-X allocates for each executed SQL statement. You can obtain the TRACE_ID of the mistaken SQL statement by using the SQL audit feature of PolarDB-X.
SQL Statement Type	The type of the mistaken SQL statement. Valid values: <ul style="list-style-type: none"> ◦ INSERT ◦ UPDATE ◦ DELETE

6. Click **Precheck**. The system checks whether a binary log file exists within the specified time range. For more information about binary log files, see [Configure local logs](#).

 **Note**

- If no binary log file exists within specified the time range, the precheck fails and the system cannot recover the data for you.
- If a binary log file exists within the specified time range, the precheck is successful and you can go to the next step.

7. Set **SQL Statement Type for Recovery** to **Rollback SQL** or **Original SQL Statement**. For more information about differences between the two methods, see [Rollback SQL statements and original SQL statements](#).
8. Click **Generate SQL** to generate an SQL flashback task. The statuses of the SQL flashback tasks that are running on the current instance appear at the bottom of the page.

What's next

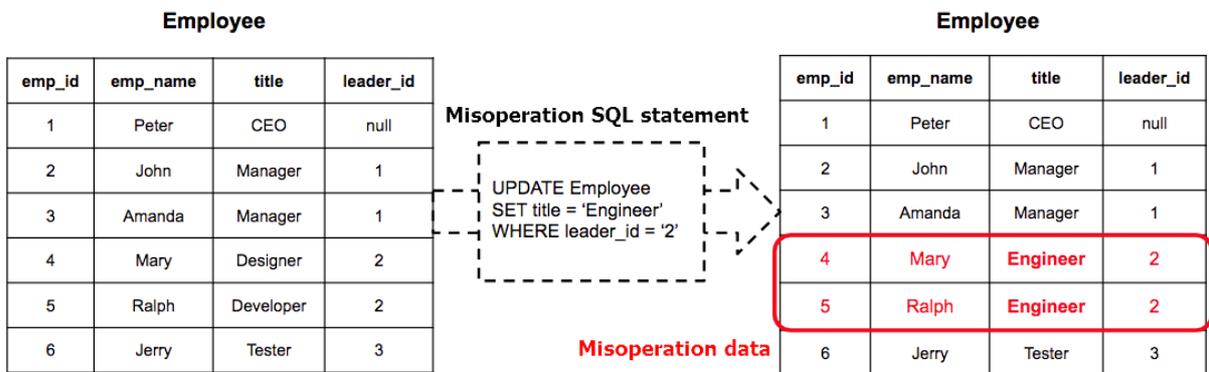
After an SQL flashback task is completed, the task information such as the exact match status and the number of recovered rows appears. You can click **Download** next to the target SQL flashback task to download the corresponding recovery file.

 **Notice** By default, the recovery file is retained for seven days. Download it as soon as possible.

13.4.6.6.3. Rollback SQL statements and original SQL statements

To support different business scenarios, PolarDB-X SQL flashback provides rollback SQL statements and original SQL statements. Before generating an SQL statement for recovering data, you must select a corresponding recovery method based on your scenario.

Recovery methods



Recovery method	Description	Example
Rollback SQL statement	<p>Traverses the events in the binary log file in reverse order to reverse the INSERT, UPDATE, and DELETE events.</p> <ul style="list-style-type: none"> The reverse of INSERT is equivalent to DELETE. The reverse of DELETE is equivalent to INSERT. The reverse of UPDATE is equivalent to the value before UPDATE. 	<pre>UPDATE Employee SET title = 'Developer' WHERE emp_id = '5' UPDATE Employee SET title = 'Designer' WHERE emp_id = '4'</pre>
Original SQL statement	<p>Traverses the events in the binary log file in order to mirror all records of the INSERT, UPDATE, and DELETE events.</p> <ul style="list-style-type: none"> An INSERT mirror is equivalent to INSERT. A DELETE mirror is equivalent to INSERT. An UPDATE mirror is equivalent to the value before INSERT. 	<pre>INSERT INTO Employee(emp_id,emp_name,title,leader_id) values('4','Mary','Designer','2') INSERT INTO Employee(emp_id,emp_name,title,leader_id) values('5','Ralph','Developer','2')</pre>

13.4.6.6.4. Exact match and fuzzy match

SQL flashback supports **exact match** and **fuzzy match** for binary log events. You do not need to select a match policy. SQL flashback automatically detects and selects the optimal match policy, and notifies you when the flashback task is completed.

Match mode	Description	Advantage	Disadvantage
Exact match	The system performs exact match on the event of a mistaken SQL statement in the binary log file and generates a recovery file.	The recovery file contains only data that is deleted or modified by the mistaken SQL statement. You can use the file directly to ensure the precision and efficiency of data recovery.	<p>The following requirements must be met:</p> <ul style="list-style-type: none"> The PolarDB-X instance is Version 5.3.4-15378085 or later. The version of the ApsaraDB RDS for MySQL instance used by the PolarDB-X database is Version 5.6 or later. You have enabled exact match of SQL flashback before the mistaken SQL statement is executed. You must provide the TRACE_ID of the mistaken SQL statement.
Fuzzy match	The system matches the information about the mistaken SQL statement in the binary log file, including the time range, table name, and SQL statement type. Then, the system generates a recovery file.	Fuzzy match is supported for all instances, regardless of the instance version or parameter settings.	Data that is deleted or modified by the mistaken SQL statement cannot be accurately matched. The recovery file contains data changes made by other business SQL operations. You must filter the required data.

Enable exact match

 **Note** Fuzzy match is enabled by default.

1. Log on to PolarDB-Xconsole, and go to the parameter settings page of the specified instance. For more information, see [Set parameters](#).
2. Change the value of `ENABLE_SQL_FLASHBACK_EXACT_MATCH` to `ON`.

13.4.6.7. Table recycle bin

13.4.6.7.1. Overview

The table recycle bin of PolarDB-X allows you to recover mistakenly deleted tables.

After the table recycle bin is enabled for your PolarDB-X database, the tables that are deleted by using the `DROP TABLE` statement are moved to the recycle bin and are no longer visible to you. After the tables are moved to the recycle bin for two hours, they are automatically cleared and cannot be recovered. You can view, recover, and clear the deleted tables in the recycle bin.

Limits and notes

- The table recycle bin feature is only supported by PolarDB-X 5.3.3-1670435 and later. For more information, see [View the instance version](#).
- The table recycle bin is disabled for your PolarDB-X database by default. For more information about how to enable it, see [Enable the table recycle bin](#).
- The table recycle bin of PolarDB-X does not support the recovery of tables deleted by the TRUNCATE TABLE command.
- Tables in the recycle bin still occupy the storage space of ApsaraDB RDS for MySQL before they are automatically cleared. To release the storage space as soon as possible, you can access the recycle bin to manually delete them.

13.4.6.7.2. Enable the table recycle bin

This topic describes how to enable the table recycle bin.

Prerequisites

An ApsaraDB RDS for MySQL database has been created in the PolarDB-X instance. For more information, see [Create a database](#).

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. At the top of the **Table Recycle Bin** page, click the tab of the database for which the table recycle bin needs to be enabled.
6. Click **Enabled**.
7. In the dialog box that appears, click **OK**.

13.4.6.7.3. Recover tables

This topic describes how to recover your tables from the table recycle bin.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. At the top of the **Table Recycle Bin** page, click the tab of the database for which you want to recover a table.
6. Click **Recover** in the Actions column of the target table.

13.4.6.7.4. Delete tables from the recycle bin

This topic describes how to delete unnecessary tables from the table recycle bin.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. At the top of the **Table Recycle Bin** page, click the tab of the database in which you want to clear a table.
6. Click **Delete** in the Actions column of the target table.

 **Note** To clear all tables from the table recycle bin, click **Empty Recycle Bin** on the tab of the corresponding database.

13.4.6.7.5. Disable the table recycle bin

If you no longer need the table recycle bin, you can disable it. This topic describes how to disable the table recycle bin.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Data Recovery > Table Recycle Bin**. The **Table Recycle Bin** page appears.
5. At the top of the **Table Recycle Bin** page, click the tab of the database for which the table recycle bin needs to be disabled.
6. Click **Disable** to disable the table recycle bin for the database.

13.4.7. Set parameters

PolarDB-X allows you to set parameters for instances and databases. You can view and modify parameter values in the PolarDB-X console as needed.

 **Note** You cannot set parameters for read-only instances.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. On the **DRDS Instance Management** page, find the target instance.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Parameter Settings**. Click the **Instance** or **Database** tab to view parameters that you can modify for instances and databases, respectively. For more information about the parameters, see [Description](#).
5. Click  next to the parameter you want to modify, enter the target value, and then click **OK**.
6. Click **Submit** in the upper-right corner to commit the new value.

 **Note** To undo parameter modification, click **Cancel** in the upper-right corner.

Description

Parameter	Level	Description
Slow SQL threshold	Instance	The threshold for slow SQL statements. SQL statements whose thresholds exceed this threshold are recorded in logical slow SQL logs.
Logical idle link timeout	Instance	The logical timeout period of the idle connection between user applications and PolarDB-X (unit: ms).
Maximum package size	Instance	The maximum network packet for the interaction between user applications and PolarDB-X (unit: byte).
Instance memory pool size limit	Instance	The maximum size of the memory pool for an instance. If the memory usage on an instance exceeds the value, an error is reported and the query ends.
Whether to prohibit all table deletion/update	Database	Specifies whether to disable all table deletion or update.
Whether to open the recycle bin	Database	Specifies whether to enable the recycle bin for storing deleted logical tables of PolarDB-X.
Temporary table size	Database	The size of the temporary table used during distributed queries in PolarDB-X (unit: row).
Number of join tables	Database	The maximum number of table shards that can be combined by JOIN when you query multiple table shards in a database.
Physical SQL timeout	Database	The timeout period of SQL statements for interaction between PolarDB-X and ApsaraDB RDS for MySQL (unit: ms). The value 0 indicates the timeout period is not limited.
SQL exact flashback switch	Database	Specifies whether to support SQL flashback exact match. It is disabled by default. After it is enabled, information about query execution is added to the binary log file used by the PolarDB-X database.
Whether to enable logical INFORMATION_SCHEMA query	Database	Specifies whether to enable logical INFORMATION_SCHEMA query (not relying on the shadow database but returning the aggregation results of logical databases and tables). When it is disabled, the original status is restored (relying on the shadow database and returning the physical database and table information).
Transaction log cleanup start time period	Database	The period during which transaction log cleanup starts at a random time.
Library-level memory pool size limit	Database	The maximum size of the database-level memory pool. When the memory usage of a PolarDB-X database exceeds this value, an error is reported and the query terminates. The value -1 indicates no limit.
Query-level memory pool size limit	Database	The maximum size of the query-level memory pool. When the memory usage of a query exceeds this value, an error is reported and the query terminates. The value -1 indicates no limit.
Whether CBO is enabled	Database	Specifies whether to enable the cost-based optimizer (CBO), including features such as Join Reorder and Hash Join.

Parameter	Level	Description
Default degree of parallelism	Database	It controls whether to start Parallel Query, and the degree of parallelism after Parallel Query is started. This parameter takes effect only when CBO is enabled. The value -1 indicates that the policy is automatically selected. The value 0 indicates that the policy is disabled.
Whether to enable the asynchronous DDL engine	Database	Specifies whether to enable the data definition language (DDL) engine. If you disable it, the execution logic of the original DDL engine remains.
Whether to enable pure asynchronous-mode under asynchronous DDL engine	Database	Specifies whether to enable the asynchronous-only mode when the asynchronous DDL engine is enabled. <ul style="list-style-type: none"> • Enabled: The execution status is returned immediately after the client connects to PolarDB-X and executes the DDL statement. Only asynchronous DDL management statements can be used to view the execution status. • Disabled: The synchronous mode remains. Specifically, the execution status is returned only after the client completes executing the DDL statement.
Maximum number of physical tables allowed to be created in a single physical database	Database	The maximum number of table shards that can be created in a database shard. If both database shards and table shards exist, the number of table shards cannot exceed this value.
INFORMATION_SCHEMA.TABLES queries whether statistics are aggregated	Database	Specifies whether to aggregate statistics of INFORMATION_SCHEMA.TABLES queries. To ensure the performance, the statistics are not aggregated by default.
Maximum number of physical sharding links	Database	The maximum number of connections between PolarDB-X and a single ApsaraDB RDS for MySQL database shard.
Minimum number of physical sharding links	Database	The minimum number of connections between PolarDB-X and a single ApsaraDB RDS for MySQL database shard.
Physical idle link timeout	Database	The idle time of the connection between PolarDB-X and ApsaraDB RDS for MySQL (unit: minute).

13.4.8. SQL audit and analysis

13.4.8.1. Description

Cloud Native Distributed Database PolarDB-X (PolarDB-X) combines the SQL audit and analysis feature with Log Service (SLS). This feature not only audits historical SQL records, but also provides real-time diagnosis and analysis of SQL execution status, performance metrics, and security risks. You can enable SQL audit and analysis in the PolarDB-X console.

Benefits

- **Easy operation:** SQL audit and analysis can be enabled with easy configuration to help you audit and analyze SQL logs in real time.
- **Lossless performance:** Pulling SQL log files from PolarDB-X nodes and uploading these logs to SLS in real time does not affect instance performance.

- **Trace to historical issues:** This feature supports importing historical SQL logs to trace issues.
- **Real-time analysis:** This feature provides real-time SQL analysis and an out-of-the-box report center based on SLS. This feature also supports custom reports and drill-down analysis, and helps you understand the execution status, performance, and security risks of databases.
- **Real-time alerts:** This feature supports real-time monitoring and alerts based on customized metrics to ensure timely response to critical business exceptions.

Limits and instructions

- You must activate Alibaba Cloud SLS to use the SQL audit and analysis feature.
- SQL audit logs are saved for 30 days by default. You can modify the log storage time as needed.
- Do not delete or modify the default settings for the project, Logstore, index, or dashboard that are created by SLS. SLS updates and upgrades the SQL log audit feature from time to time. The indexes and default reports of the exclusive Logstore are also automatically updated.
- The maximum length of a single SQL statement is 5 MB.

Scenarios

- **Troubleshoot SQL problems**

After the SQL audit and analysis feature is enabled, you can quickly search SQL logs to locate and troubleshoot problems. For example, to check whether a DROP operation is performed, you can perform the following query:

```
sql_type: Drop
```

The query result contains information such as the SQL execution time, user, and IP address of the client that runs the SQL statement.

- **Analyze costly SQL templates**

In most applications, SQL statements are dynamically generated based on several templates, with different parameters. The real-time analysis feature of SLS allows you to obtain the list of costly SQL statements in the current database.

For example, execute the following query:

```
| SELECT sql_code as "SQL template ID",
round(total_time * 1.0 /sum(total_time) over() * 100,2) as "execution time share (%)",
execute_times as "number of execution times",
round(avg_time) as "average execution time",
round(avg_rows) as "average number of affected rows",
CASE WHEN length(sql) > 200 THEN concat(substr(sql, 1, 200), '.....') ELSE trim(lpad(sql, 200,$) end as "sample SQL" F
ROM (SELECT sql_code, count(1) as execute_times,
sum(response_time) as total_time,
avg(response_time) as avg_time,
avg(affect_rows) as avg_rows,
arbitrary(sql) as sql FROM log GROUP BY sql_code) ORDER BY "execution time share (%)" desc limit 10
```

The search result contains the SQL template ID, ratio of response time of the statement generated from the template in the total response time of SQL statements, number of executions, average execution time, average number of affected rows, and sample SQL statement. You can find and optimize the most costly SQL templates in the application based on the analysis result.

- **Collect log statistics**

To help you analyze issues, PolarDB-X combines the SQL audit and analysis feature with SLS and provides out-of-the-box reports. You can diagnose and analyze the running status, performance, and potential security risks of databases in real time.

13.4.8.2. Enable SQL audit and analysis

The SQL audit and analysis feature is disabled by default. You can manually enable it in the PolarDB-X console. By default, you can perform only audit and analysis on the log data generated after the SQL audit and analysis feature is enabled. You can also import a portion of historical data.

Prerequisites

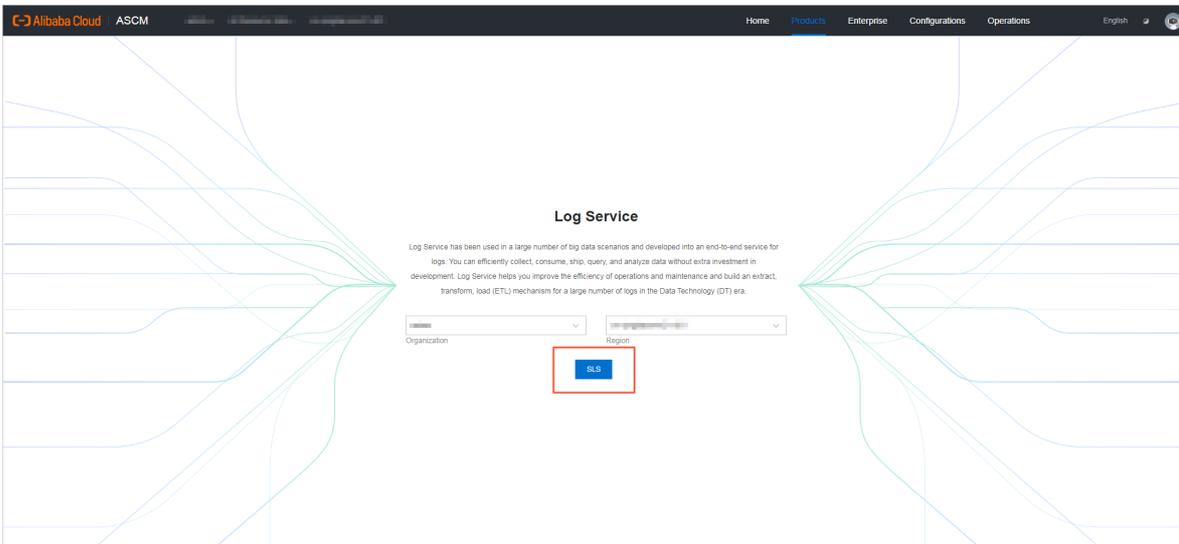
The SQL audit and analysis feature depends on Log Service (SLS). You must activate SLS before you use this feature.

Procedure

1. Log on to the SLS console. For more information, see *Log Service User Guide > Quick Start > Log on to the Log Service console*.
2. Select the organization to which the PolarDB-X instance belongs.

? **Note** The logon account must be consistent with the logon account of PolarDB-X.

3. Click SLS to go to the Log Service page.



4. Log on to the PolarDB-X console.
5. Find the target instance in the instance list.
6. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
7. In the left-side navigation pane, choose **Diagnostics and Optimization > SQL Audit and Analysis**.
8. In the left-side section, select the database for which you want to enable the SQL audit and analysis feature.
9. On the SQL Audit and Analysis page, turn on the switch next to **SQL Audit Log Status of Current Database** on the right.

? **Note** On the SQL Audit and Analysis page, you can also turn on the switch of **SQL Audit and Analysis** next to the target database in the left-side section.

10. Confirm whether to import historical data.

Note By default, you can analyze and audit only the logs that are generated after the SQL audit and analysis feature is enabled. If you find the historical data of the PolarDB-X database is modified but the SQL audit and analysis feature is not enabled, you can import historical data and include historical logs in the audit and analysis scope to trace data tampering. PolarDB-X dynamically checks the scope of historical data that can be imported based on the log storage on the PolarDB-X instance. Logs within seven days can be imported.

- If you need to import historical data, enable **Import Historical Data or Not**, specify the backtrace start time and end time, and then click **Enable**.
- If you do not need to import historical data, click **Enable**.

What's next

Every time you use the SQL audit and analysis feature, you must repeat the preceding steps.

13.4.8.3. Log fields

This topic describes the log fields in SQL audit and analysis.

Field	Description	Supported version
<code>__topic__</code>	The log topic in the format of <code>drds_audit_log_{instance_id name_{db_name}}</code> , such as <code>drds_audit_log_drdsxyzabcd_demo_drds_db</code> .	All versions
<code>instance_id</code>	The ID of the PolarDB-X instance.	All versions
<code>db_name</code>	The name of the PolarDB-X database	All versions
<code>user</code>	The user name used to run the SQL statement.	All versions
<code>client_ip</code>	The IP address of the client that accessed the PolarDB-X instance.	All versions
<code>client_port</code>	The port of the client that accessed the PolarDB-X instance.	All versions
<code>sql</code>	The executed SQL statement.	All versions
<code>trace_id</code>	The trace ID of the SQL statement when it was executed. If a transaction was executed, it is tracked by an ID that consists of the trace ID, a hyphen, and a number, for example, <code>drdsabcdxyz-1</code> and <code>drdsabcdxyz-2</code> .	All versions
<code>sql_code</code>	The hash value of the template SQL statement.	All versions
<code>hint</code>	The hint that was used to execute the SQL statement.	All versions

Field	Description	Supported version
table_name	The name of the table involved in the query. Separate multiple tables by commas (,).	All versions
sql_type	The type of the SQL statement. Valid values: SELECT, INSERT, UPDATE, DELETE, SET, ALTER, CREATE, DROP, TRUNCATE, REPLACE, and Other.	All versions
sql_type_detail	The name of the SQL parser.	All versions
sql_time	The start time for the execution of the SQL statement. The time follows the <code>yyyy-MM-dd HH:mm:ss.SSS</code> format.	All versions
response_time	The response time. Unit: milliseconds.	Version 5.3.4-15378085 and later
affect_rows	The number of rows returned when the SQL statement was executed. The number of rows affected when the INSERT, DELETE, or UPDATE statement was executed.	Version 5.3.4-15378085 and later
fail	Indicates whether an error occurred in the execution of the SQL statement. Valid values: <ul style="list-style-type: none"> • 0: successful • 1: failed 	Version 5.3.4-15378085 and later

13.4.8.4. Log analysis

The SQL audit and analysis feature is based on Log Service (SLS) and provides powerful log analytics capabilities. This topic describes SQL statements for log analysis in common scenarios and provides relevant examples.

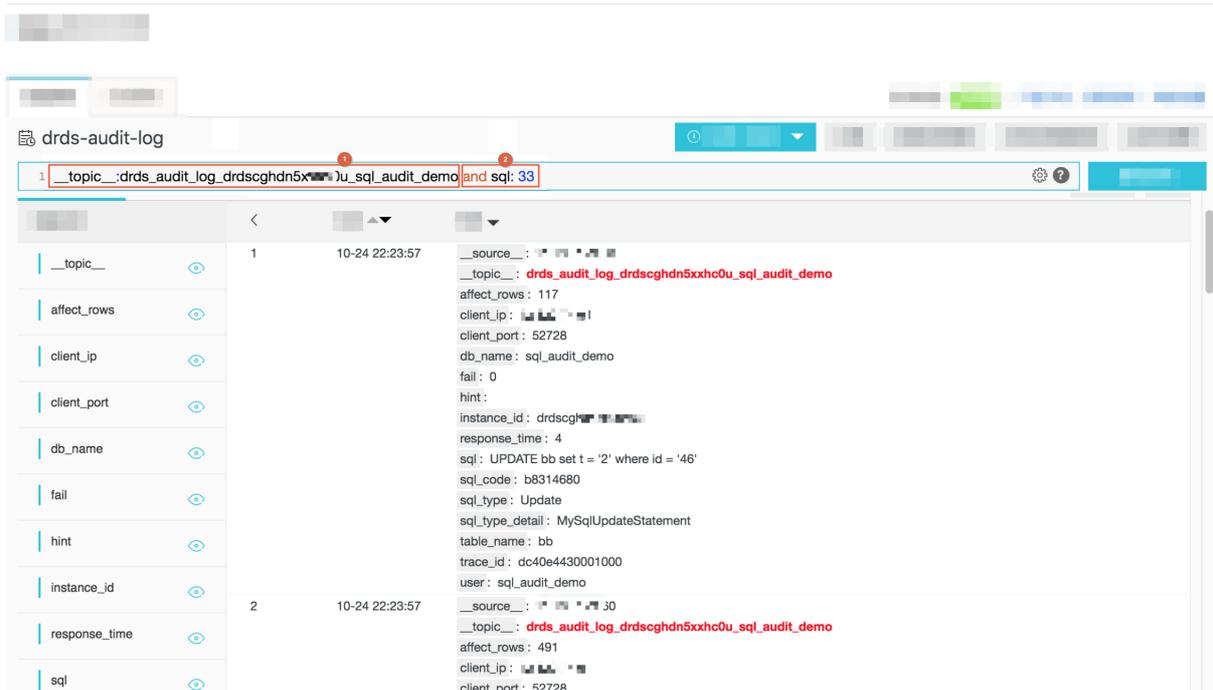
After the SQL audit and analysis feature is enabled, you can perform audit and analysis on SQL log files by using the query and analysis syntax of SLS on the SQL Audit and Analysis page. Based on the query and analysis syntax of SLS, you can find problematic SQL statements on the Log Analysis tab and analyze the SQL statement execution status, performance metrics, and security issues of PolarDB-X. For more information about the query and analysis syntax of SLS, see *Log Service User Guide > Query and Analysis > Query Syntax and Functions > Query Syntax*.

Precautions

All the audit logs of PolarDB-X databases in the same region are written to the same Logstore in SLS. Therefore, by default, the SQL Audit and Analysis page provides the filter conditions based on `__topic__`, to ensure that the searched SQL log files are from PolarDB-X. Therefore, all the statements provided in this topic must be used after the existing filter conditions.

An example is shown in the following figure:

- The ① part is the default filter condition.
- The ② part is the additional filter condition.



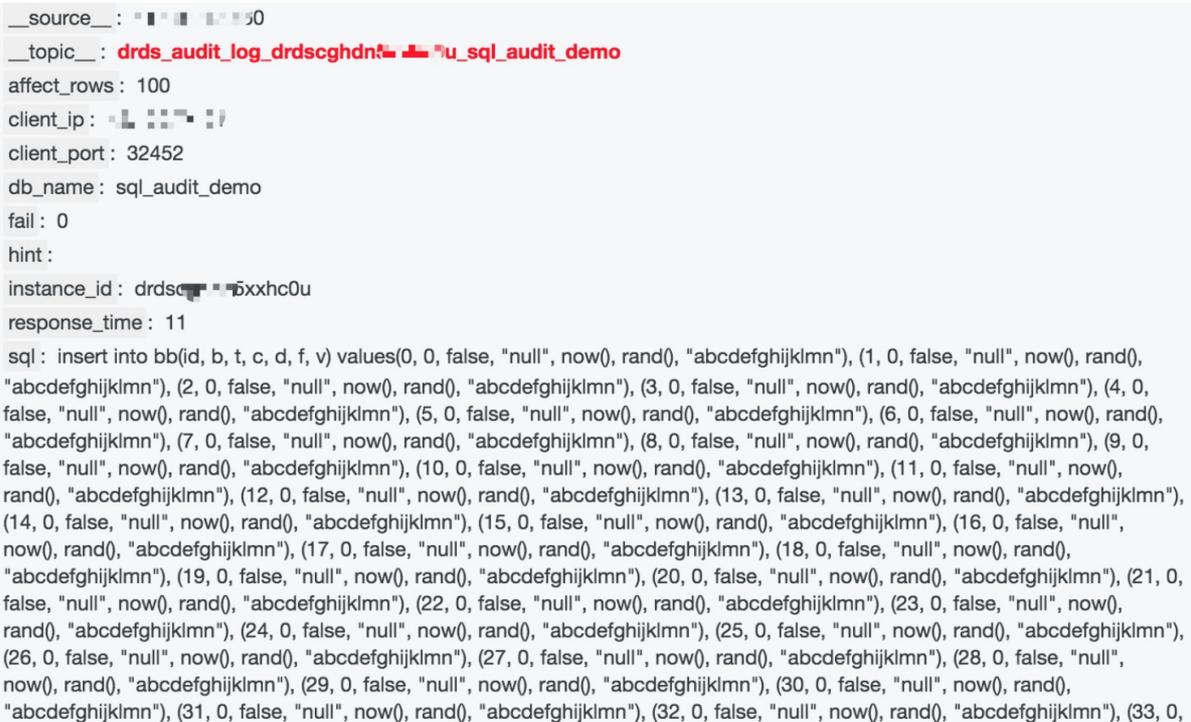
Find problematic SQL statements

- Fuzzy search

For example, to query SQL statements that contain the "34" keyword, enter the following content in the search box:

```
and sql: 34
```

The result is shown in the following figure.



- Field search

Based on built-in index fields, the SQL audit and analysis feature also supports field-based search.

For example, to query SQL statements of the Drop type, execute the following statement:

```
and sql_type:Drop
```

The result is shown in the following figure.

```
__source__: ...
__topic__: drds_audit_log_drdschdr...0u_sql_audit_demo
affect_rows: 0
client_ip: ...
client_port: 36085
db_name: sql_audit_demo
fail: 0
hint:
instance_id: drdschdr...0u
response_time: 3172
sql: drop table if exists bb
sql_code: 0cfc96e8
sql_type: Drop
sql_type_detail: SQLDropTableStatement
table_name: bb
trace_id: dc408feedc00000
user: sql_audit_demo
```

- Multi-condition search

You can use the "and" and "or" keywords to perform a multi-condition search.

For example, you can query the delete operation on the rows whose id is 34:

```
and sql:34 and sql_type: Delete
```

- Search based on numeric comparison

affect_rows and response_time in the index filed are numeric values and support comparison operators.

For example, you can query the SQL INSERT statements whose response_time is greater than 1s.

```
and response_time > 1507 and sql_type: Insert
```

For example, you can query the SQL statement that deletes more than 100 rows of data:

```
and affect_rows > 100 and sql_type: Delete
```

Analysis of the SQL statement execution status

This section introduces the statements used to query the SQL statement execution status in PolarDB-X.

- Failure rate of SQL statement execution

Execute the following statement to query the failure rate of SQL statement execution:

```
| SELECT sum(case when fail = 1 then 1 else 0 end) * 1.0 / count(1) as fail_ratio
```

The result is shown in the following figure.

fail_ratio +

0.0010322901477612633

If your business is sensitive to the error rate of SQL statement execution, you can customize the alert information based on the query result. Click **Save as Alert** in the upper-right corner of the page.

In the alert settings shown in the preceding figure, the number of log entries that have an error rate of SQL statement execution greater than 0.01 within 15 minutes is checked within every 15 minutes. You can also customize alerts as needed.

- Total number of rows returned by SELECT statements

Execute the following statement to query the cumulative number of rows queried by SELECT statements:

```
and sql_type: Select | SELECT sum(affect_rows)
```

- SQL statement type distribution

Execute the following statement to query the SQL statement type distribution:

```
| SELECT sql_type, count(sql) as times GROUP BY sql_type
```

- IP address distribution of SQL independent users

Execute the following statement to query the distribution of IP addresses of independent users who execute SQL statements:

```
| SELECT user, client_ip, count(sql) as times GROUP BY user, client_ip
```

SQL performance analysis

This section describes typical SQL statements for SQL performance analysis.

- Average response time of SELECT statements

Execute the following statement to query the average response time of SELECT statements:

```
and sql_type: Select | SELECT avg(response_time)
```

- Distribution of SQL statement response time

Execute the following statement to query the distribution of SQL statement response time:

```
and response_time > 0 | select case when response_time <=10 then '<=10 ms when response_time > 10 and response_time <= 100 then '10~100 ms when response_time > 100 and response_time <= 1000 then '100 ms ~ 1s 'When response_time > 1000 and response_time <= 10000 then '1s ~ 10s' when response_time > 10000 and response_time <= 60000 then '10s ~ 1 min'> 1 min' end as latency_type, count(1) as cnt group by latency_type order by latency_type DESC
```

The preceding query shows the distribution of SQL statement execution time based on a given time range. You can adjust the time range to obtain finer-grained results.

- Top 50 slow SQL statements

Execute the following statement to query slow SQL statements:

```
| SELECT date_format(from_unixtime(__time__), '%m/%d %H:%i:%s') as time, user, client_ip, client_port, sql_type, affect_rows, response_time, sql ORDER BY response_time desc LIMIT 50
```

The following figure shows the result, which includes the SQL statement execution time, user name, IP address, port number, SQL statement type, number of affected rows, response time, and text of SQL statements.

time	user	client_ip	client_port	sql_type	affect_rows	response_time	sql
09/28 14:04:05	sql_audit_demo	192.168.1.101	477	Drop	0	9583	drop table if exists bb
09/28 14:04:05	sql_audit_demo	192.168.1.101	477	Drop	0	9583	drop table if exists bb
09/28 14:04:05	sql_audit_demo	192.168.1.101	477	Drop	0	9583	drop table if exists bb
09/27 17:38:18	sql_audit_demo	192.168.1.101	473	Drop	0	7200	drop table if exists bb

• Top 10 costly SQL templates

In most applications, SQL statements are dynamically generated based on several templates, and only the parameters are different. You can find, analyze, and optimize the costly SQL templates based on template IDs. Enter the following query statement:

```
| SELECT sql_code as "SQL template ID", round(total_time * 1.0 /sum(total_time) over() * 100, 2) as "response time share (%)" ,execute_times as "number of executions", round(avg_time) as "average response time", round(avg_rows) as "average number of affected rows", CASE WHEN length(sql) > 200 THEN concat(substr(sql, 1, 200), '.....') ELSE trim(lpad(sql, 200,'hour') end as "sample SQL" FROM (SELECT sql_code, count(1) as execute_times, sum(response_time) as total_time, avg(response_time) as avg_time, avg(affect_rows) as avg_rows, arbitrary(sql) as sql FROM log GROUP BY sql_code) ORDER BY "execution time share (%)" desc limit 10
```

The statistics include the SQL template ID, percentage of response time of the statement generated from the template in the total response time of SQL statements, number of executions, average response time, average number of affected rows, and sample SQL statement. For better display effect, each page displays 200 entries. In the preceding query result, statements are ranked by the response time share. However, you can rank the statements by the average response time or the number of executions to troubleshoot relevant issues.

• Average transaction response time

For SQL statements within the same transaction, the preset trace_id field prefixes are the same, and the suffixes are '-' followed by sequence numbers. trace_id of non-transactional SQL statements does not contain '-'. Based on this, you can analyze the performance of transactions.

Note Transaction analysis is less efficient than other query operations because it involves prefix matching.

For example, execute the following statement to query the average response time of transactions:

```
| SELECT sum(response_time) / COUNT(DISTINCT substr(trace_id, 1, strpos(trace_id, '-') - 1)) where strpos(trace_id, '-') > 0
```

• Top 10 slow transactions

You can query the list of slow transactions by response time of transactions. Use the following statement:

```
| SELECT substr(trace_id, 1, strpos(trace_id, '-') - 1) as "transaction ID" , sum(response_time) as "response time" where strpos(trace_id, '-') > 0 GROUP BY substr(trace_id, 1, strpos(trace_id, '-') - 1) ORDER BY "response time" DESC LIMIT 10
```

Based on this, you can use the transaction ID to search for all the SQL statements under the transaction and analyze the specific causes of slow execution. Use the following statement:

```
and trace_id: db3226a20402000*
```

- Top 10 transactions with batch operations

Based on the number of rows affected by SQL statements in a transaction, you can obtain the list of transactions that contain batch operations. Use the following statement:

```
| SELECT substr(trace_id, 1, strpos(trace_id, '-') - 1) as "transaction ID" , sum(affect_rows) as "number of affected rows" where strpos(trace_id, '-') > 0 GROUP BY substr(trace_id, 1, strpos(trace_id, '-') - 1) ORDER BY "number of affected rows" DESC LIMIT 10
```

SQL security analysis

This section provides typical query statements for SQL security analysis.

- Distribution of types of failed SQL statements

```
and fail > 0 | select sql_type, count(1) as "number of errors" group by sql_type
```

- High-risk SQL statements

High-risk SQL statements are of the Drop or Truncate type. You can also add more conditions as needed.

```
and sql_type: Drop OR sql_type: Truncate
```

- SQL batch delete events

```
and affect_rows > 100 and sql_type: Delete | SELECT date_format(from_unixtime(__time__), '%m/%d %H:%i:%s') as time, user, client_ip, client_port, affect_rows, sql ORDER BY affect_rows desc LIMIT 50
```

13.4.8.5. Log reports

Based on Log Service (SLS), the SQL audit and analysis feature of PolarDB-X provides out-of-the-box report centers, including the Operation Center, Performance Center, and Security Center. This feature allows you to fully understand the performance status, performance metrics, and potential security risks of your PolarDB-X databases.

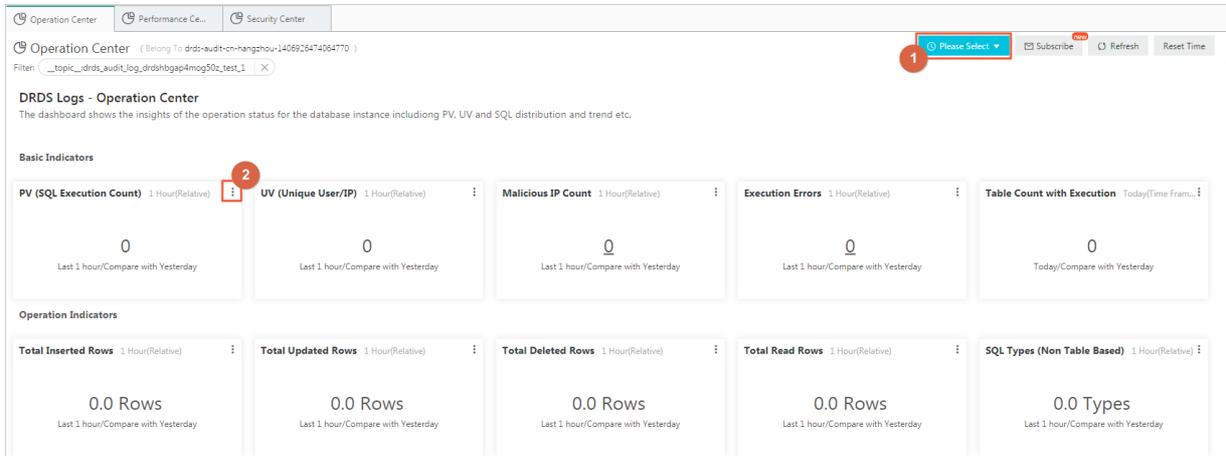
After [Enable SQL audit and analysis](#), click the **Log Reports** tab on the current page. You can view the reports pages provided by SLS, including **Operation Center**, **Performance Center**, and **Security Center**.

Note

- All the audit logs of PolarDB-X databases in the same region are written to the same Logstore in SLS. Therefore, when you view the reports of the current PolarDB-X database, filter conditions based on the `__topic__:drds_audit_log_instance id_database` name are added by default, which indicates that you are viewing the data of the current database. For example, `drds_audit_log_drdsxyzabcd_demo_drds_db`.
- If the version of the PolarDB-X instance is earlier than Version 5.3.4-15378085, the relevant fields are missing from SQL logs. For more information about log fields, see [Log fields](#). The Log Reports tab provides only a simplified version of Operation Center. To use a full version of reports, upgrade the instance to the latest version.

View reports

Statistics in the charts on the Log Reports tab are generated for different time periods. You can change the time range as needed. You can change the time range for all charts or a single chart.



- Click the time selector (position ① in the figure). In the dialog box that appears, you can change the time range of all the charts on the current page.
- Click the time selector of a chart (position ② in the figure) to modify the time range of the chart.

Operation Center

Operation Center shows the metrics, distribution, and trends of SQL statement execution in PolarDB-X databases.

Item	Type	Default time range	Description
PV (SQL Execution Count)	Single value	1 Hour (Relative)	The number of SQL statement executions
UV (Unique User/IP)	Single value	1 Hour (Relative)	The number of unique user-IP groups
Malicious IP Count	Single value	1 Hour (Relative)	The number of malicious IP addresses. For more information about the definition of malicious IP addresses, see security detection functions.
Execution Errors	Single value	1 Hour (Relative)	The number of SQL statements with execution errors
Table Count with Execution	Single value	1 Hour (Relative)	The total number of tables operated by SQL statements
Total Inserted Rows	Single value	1 Hour (Relative)	The total number of rows inserted by INSERT statements
Total Updated Rows	Single value	1 Hour (Relative)	The total number of rows updated by UPDATE statements

Item	Type	Default time range	Description
Total Deleted Rows	Single value	1 Hour (Relative)	The total number of rows deleted by DELETE statements
Total Read Rows	Single value	1 Hour (Relative)	The total number of rows returned by SELECT statements
SQL Types (Non Table Based)	Single value	1 Hour (Relative)	The types of SQL statements used for non-table operations, such as SHOW VARIABLES LIKE
SQL Execution Trend	Column chart	1 Hour (Relative)	The distribution trend of SQL statement executions and the distribution trend of failed SQL statements
Operated Tables	Flow diagram	1 Hour (Relative)	The distribution of tables operated by SQL statements
SQL Type	Flow diagram	1 Hour (Relative)	The distribution of SQL statement types by time
User Distribution	Pie chart	1 Hour (Relative)	The distribution of users who execute SQL statements
SQL Type Distribution	Area chart	1 Hour (Relative)	The percentage of SQL statement types in the current time range
Tables with Most Operations (Top 50)	Table	1 Hour (Relative)	The list of top tables by the number of operations, including table names and the number of operations such as read, delete, update, and insert
SQL Type (World)	Map	1 Hour (Relative)	The distribution of IP addresses of clients that execute the SQL statements, on the world map
SQL Type (China)	Map	1 Hour (Relative)	The distribution of IP addresses of clients that execute the SQL statements, on the map of China

Performance Center

Performance Center shows performance metrics, the distribution of slow and fast SQL statements, and the distribution and sources of costly SQL statements in PolarDB-X databases.

Item	Data type	Default time range	Description
------	-----------	--------------------	-------------

Item	Data type	Default time range	Description
Peak SQL Execution Traffic	Single value	1 Hour (Relative)	The maximum number of SQL statements executed per second
Peak Select Traffic	Single value	1 Hour (Relative)	The maximum number of rows returned by SELECT statements per second
Peak Insert Traffic	Single value	1 Hour (Relative)	The maximum number of rows inserted by INSERT statements per second
Peak Update Traffic	Single value	1 Hour (Relative)	The maximum number of rows updated by UPDATE statements per second
Peak Delete Traffic	Single value	1 Hour (Relative)	The maximum number of rows deleted by DELETE statements per second
Average Response Time	Single value	1 Hour (Relative)	The average response time of SQL statements
Select SQL	Single value	1 Hour (Relative)	The average number of SELECT statements executed per second
Insert SQL	Single value	1 Hour (Relative)	The average number of INSERT statements executed per second
Update SQL	Single value	1 Hour (Relative)	The average number of UPDATE statements executed per second
Delete SQL	Single value	1 Hour (Relative)	The average number of DELETE statements executed per second
Select/Update Traffic Trend	Line chart	1 Hour (Relative)	The distribution of rows affected by the SELECT and UPDATE statements over time
SQL Execution Time Distribution	Pie chart	1 Hour (Relative)	The distribution of execution time of SQL statements
Slow SQL Table Distribution	Pie chart	1 Hour (Relative)	The distribution of tables targeted by slow SQL statements whose response time exceeds 1s
Slow SQL User Distribution	Pie chart	1 Hour (Relative)	The distribution of users who execute slow SQL statements with response time of more than 1s

Item	Data type	Default time range	Description
Slow SQL Type Distribution	Pie chart	1 Hour (Relative)	The distribution of types of slow SQL statements whose response time exceeds 1s
Slow SQL (Top 50)	Table	1 Hour (Relative)	The table of slow SQL statements whose response time exceeds 1s, including the time, client, response time, PolarDB-X instance, database, table, user, affected rows, SQL type, and SQL text
SQL Template Execution Time Top 20	Table	1 Hour (Relative)	Statistics of the execution status of the SQL statements in the template based on the specified SQL template, including the SQL template ID, response time share, number of executions, average response time, average number of affected rows, and sample SQL statement
Transaction Affected Rows Top 20	Table	1 Hour (Relative)	The table of top 20 transaction-by the number of affected rows, including the transaction ID and the number of affected rows
Transaction Executed Time Top 20	Table	1 Hour (Relative)	The table of top 20 transactions by response time, including the transaction ID and the number of affected rows

Security Center

Security Center shows failed and malicious SQL statement executions in PolarDB-X databases, and the details, distribution, and trends of malicious SQL batch delete and update events.

Item	Type	Default time range	Description
Error Count	Single value	1 Hour (Relative)	The number of failed SQL statement executions
Batch Delete Events	Single value	1 Hour (Relative)	The number of SQL statements for batch delete events (more than 100 rows)
Batch Update Events	Single value	1 Hour (Relative)	The number of SQL statements for batch update events (more than 100 rows)

Item	Type	Default time range	Description
Malicious SQL Executions	Single value	1 Hour (Relative)	The number of malicious SQL statement executions (Drop and Truncate)
Malicious IP Count	Single value	1 Hour (Relative)	The number of malicious IP addresses. For more information about the definition of malicious IP addresses, see security detection functions.
Error Distribution	Area chart	1 Hour (Relative)	The distribution of types of failed SQL statements
Distribution of Client with Errors	Map	1 Hour (Relative)	The distribution of clients for failed SQL statements on the map of China
Client with Most Errors	Table	1 Hour (Relative)	The table of clients on which the execution of SQL statements failed, including the IP address, number of errors, type of failed SQL statement, and sample failed SQL statement
Malicious SQL Executions	Table	1 Hour (Relative)	The table of malicious SQL statement executions, including the time, IP address, SQL, PolarDB-X instance ID, database, table, and user
Batch Delete Events (Top 50)	Table	1 Hour (Relative)	The table of top SQL batch delete events, including the earliest execution time, most recent execution time, PolarDB-X instance ID, database, table, number of executions, average number of deleted rows, average response time, and sample SQL statement
Batch Update Events (Top 50)	Table	1 Hour (Relative)	The table of top SQL batch update events, including the earliest execution time, most recent execution time, PolarDB-X instance ID, database, table, number of executions, average number of updated rows, average response time, and sample SQL statement

13.4.9. Monitor PolarDB-X instances

13.4.9.1. View monitoring information

PolarDB-X provides multi-dimensional monitoring. This topic describes how to view monitoring information in the PolarDB-X console.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the **Basic Information** page, choose **Monitoring and Alerts > Instance Monitoring** in the left-side navigation pane.
5. On the **Instance Monitoring** page, select a monitoring dimension and the corresponding metrics to view details. For more information about metrics, see [Monitoring metrics](#).

13.4.9.2. Monitoring metrics

Instance monitoring is divided into resource monitoring and engine monitoring. Engine monitoring metrics are classified into metrics at the PolarDB-X instance level and metrics at the PolarDB-X database level. When some engine monitoring metrics are abnormal, you can directly check the metrics of each database to locate the database with performance problems. The following table describes the metrics of these two types in details.

Monitoring item	Category	Description	Data collection cycle	Data retention period	Description
CPU Utilization (%)	Resource monitoring	The average CPU utilization of PolarDB-X server nodes.	1 minute	3 days	-
Memory Usage (%)	Resource monitoring	The memory usage of JVM Old Generation on PolarDB-X server nodes.	1 minute	3 days	Memory usage fluctuations are normal.
Inbound Traffic (Kbps)	Resource monitoring	The total inbound network traffic of PolarDB-X server nodes.	1 minute	3 days	Inbound network traffic is generated when ApsaraDB RDS for MySQL returns data to PolarDB-X.

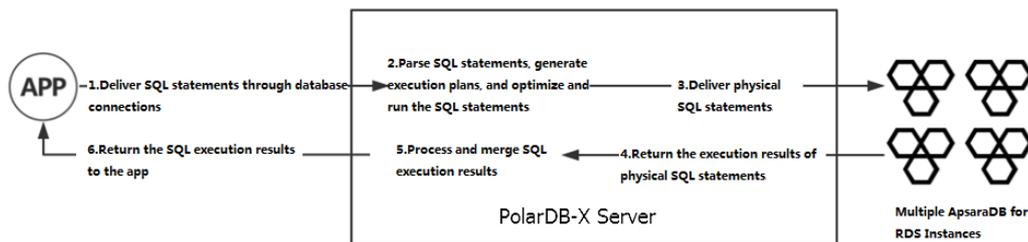
Monitoring item	Category	Description	Data collection cycle	Data retention period	Description
Outbound Traffic (Kbps)	Resource monitoring	The total outbound network traffic of PolarDB-X server nodes.	1 minute	3 days	Outbound network traffic is generated when a PolarDB-X instance sends a physical SQL statement to an ApsaraDB RDS for MySQL instance or a PolarDB-X instance returns data to an application.
Logical QPS	Engine monitoring	The total number of SQL statements processed per second on PolarDB-X server nodes.	5 seconds	7 days	-
Physical QPS	Engine monitoring	The total number of SQL operations sent from PolarDB-X server nodes to ApsaraDB RDS for MySQL per second.	5 seconds	7 days	One logical SQL statement can be partitioned into multiple physical SQL statements.
Logical RT (ms)	Engine monitoring	The average response time (RT) for processing each SQL statement by PolarDB-X.	5 seconds	7 days	If a logical SQL statement is partitioned into physical SQL statements for delivery, the logical RT of the SQL statement contains the RT of the physical SQL statements.
Physical RT (ms)	Engine monitoring	The average RT for transmitting SQL statements from PolarDB-X to ApsaraDB RDS for MySQL.	5 seconds	7 days	-
Connections	Engine monitoring	The total number of connections established between an application and PolarDB-X.	5 seconds	7 days	The connections from PolarDB-X to ApsaraDB RDS for MySQL are not included.

Monitoring item	Category	Description	Data collection cycle	Data retention period	Description
Active Threads	Engine monitoring	The number of threads that are used by PolarDB-X to run SQL statements.	5 seconds	7 days	-

13.4.9.3. How metrics work

Before analyzing metrics, you need to understand the execution process of SQL statements on PolarDB-X.

PolarDB-X SQL execution flowchart



In the entire SQL execution process, the execution status of steps 2 through 4 is reflected in various metrics of PolarDB-X.

- In step 2, SQL parsing, optimization, and execution consume CPU resources. A more complex SQL statement (with a complex structure or ultra-long length) consumes more CPU resources. You can run the **TRACE** command to trace the SQL execution process. You can see the time consumed by an SQL statement during optimization. The longer time consumed indicates a higher CPU utilization.
- In step 3, the delivery and execution of physical SQL statements consume I/O resources. You can analyze the execution status of physical SQL statements based on metrics such as logical queries per second (QPS), physical QPS, logical response time (RT), and physical RT. For example, if the physical QPS is low and the physical RT is high, the current ApsaraDB RDS for MySQL instance is processing SQL statements very slowly. You need to check the performance of the ApsaraDB RDS for MySQL instance.
- In step 5, the SQL execution results are processed and integrated. These operations convert the execution results of physical SQL statements. In most cases, only SQL metadata is converted, which consumes few resources. However, the CPU utilization is high for steps such as `heap sort`. For more information about how to determine the consumption of SQL statements at this stage, see [Details about a low SQL statement](#).

13.4.9.4. Prevent performance problems

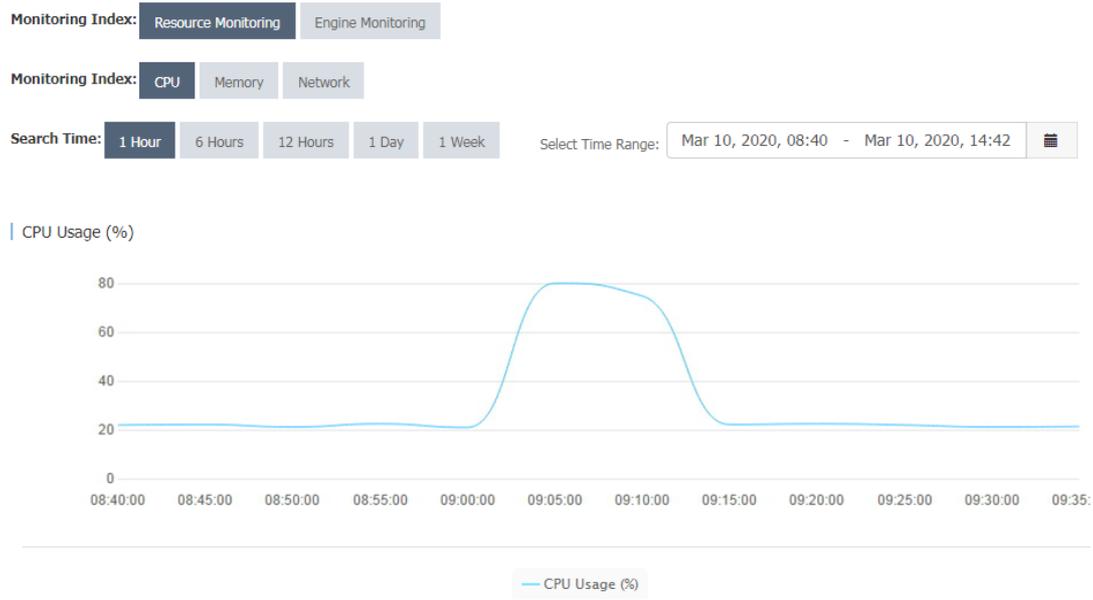
13.4.9.4.1. Example 1: PolarDB-X CPU utilization

Performance metrics change with the system business traffic.

The following describes the CPU utilization in two common cases:

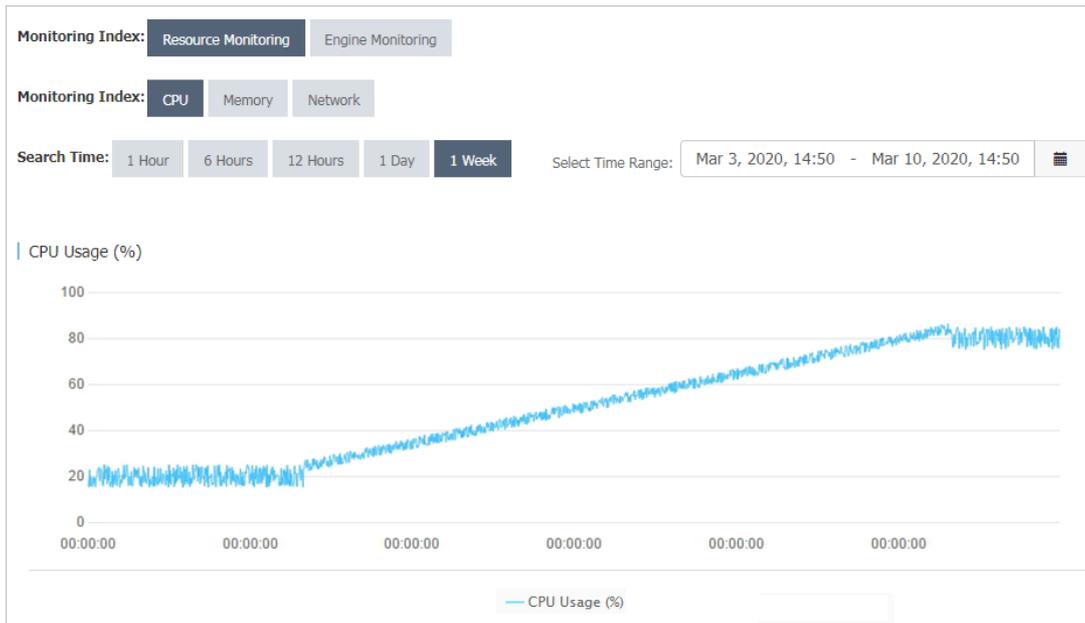
- An application has a shopping spree activity at 09:00 every morning. Therefore, the traffic of the system increases significantly at this time point. According to the monitoring data, the CPU utilization of the PolarDB-X instance increased from 20% to about 80% from about 09:00, with the traffic peak lasting about 10 minutes.

CPU utilization-1



- The system traffic keeps increasing with an application until it reaches a plateau. The monitored CPU utilization of the PolarDB-X instance also reflects this change.

CPU utilization-2



When the load on the PolarDB-X instance changes with the business, you must pay close attention to the changes in metrics. If the CPU utilization exceeds the threshold, you must upgrade the PolarDB-X specifications to alleviate the performance pressure.

You can set alert rules for instances in the PolarDB-X console. When the average CPU utilization exceeds the preset threshold, the system sends short messages to the corresponding contacts. You can set the CPU utilization threshold as needed. We recommend that you set it to 80%.

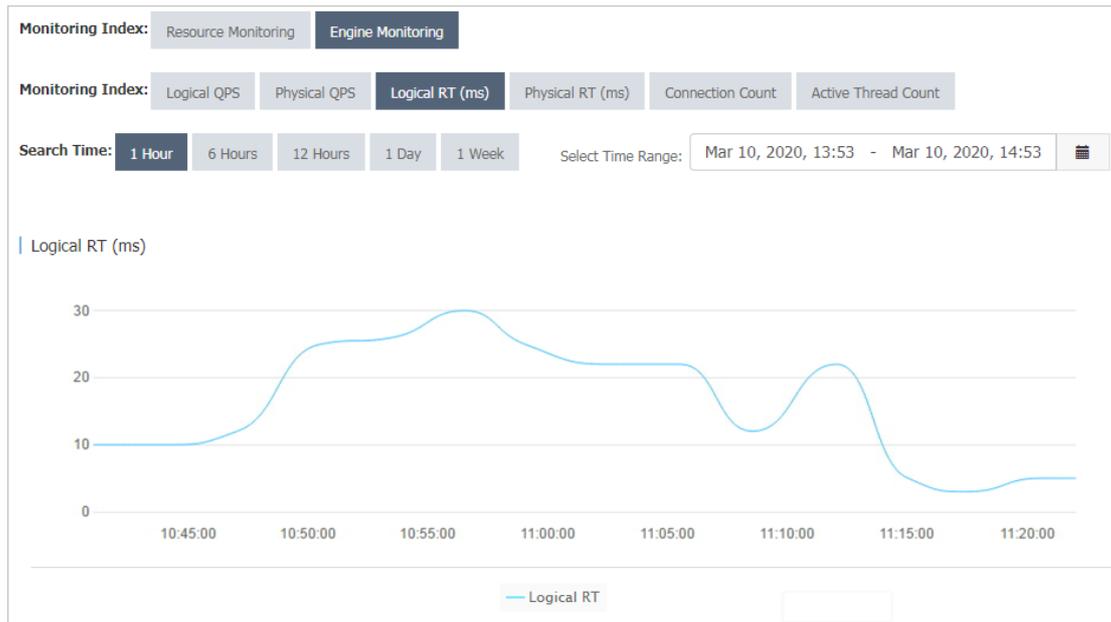
13.4.9.4.2. Example 2: Logical RT and physical RT

You can observe the difference between the logical response time (RT) and physical RT.

Logical RT refers to the time from when a PolarDB-X instance receives a logical SQL statement to when it returns data to an application. Physical RT refers to the time from when a PolarDB-X instance sends a physical SQL statement to an ApsaraDB RDS for MySQL instance to when it receives the data returned by the ApsaraDB RDS for MySQL instance.

If a logical SQL statement is partitioned into one or more physical SQL statements, the logical RT is greater than or equal to the physical RT. Ideally, PolarDB-X performs only a few operations on the data returned by ApsaraDB RDS for MySQL. Therefore, logical RT is slightly longer than physical RT. Under special circumstances, physical SQL queries are run fast, while logical SQL queries take a long time to run. In this case, the logical RT and physical RT are as follows.

Logical RT



Physical RT



As shown in the preceding figures, the change trends of logical RT and physical RT in the two monitoring charts are basically the same, while logical RT fluctuates between 10 ms and 20 ms and physical RT fluctuates between 2 ms and 5 ms. This means that PolarDB-X has a heavy load, which can be solved by upgrading the PolarDB-X configuration. If both the logical RT and physical RT are high, you can upgrade the ApsaraDB RDS for MySQL configuration or optimize SQL statements on the ApsaraDB RDS for MySQL instance.

13.4.9.4.3. Example 3: Logical QPS and physical QPS

You can observe the difference between the logical queries per second (QPS) and physical QPS.

According to the monitoring data, the logical QPS and physical QPS have the same trends, but the difference between the two is relatively large and in a certain proportion.

Logical QPS



Physical QPS



As shown in the preceding figures, logical QPS fluctuates between 80 and 150, and physical QPS fluctuates between 700 and 1,200.

The reason is that PolarDB-X generates physical SQL statements based on logical SQL statements. The ratio of logical SQL statements to physical SQL statements is not necessarily 1:1. For example, a PolarDB-X logical table is created by using the following statement:

```
CREATE TABLE drds_user
(id int,
name varchar(30))
dbpartition by hash(id);
```

When the query condition contains the database shard key, PolarDB-X pushes the logical SQL statement down to the ApsaraDB RDS for MySQL instance for execution. According to the execution plan, the number of physical SQL statements is 1:

```
mysql> explain select name from drds_user where id = 1;
+-----+-----+-----+
| GROUP_NAME | SQL | PARAMS |
+-----+-----+-----+
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`id` = 1) | {} |
+-----+-----+-----+
```

When the query does not contain the database shard key, PolarDB-X partitions the logical SQL statement into multiple physical SQL statements. The following execution plan shows that there are eight physical SQL statements:

```
mysql> explain select name from drds_user where name = 'LiLei';
+-----+-----+-----+
| GROUP_NAME | SQL | PARAMS |
+-----+-----+-----+
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
| SANGUAN_BSQT_0001_RDS | select `drds_user`.`name` from `drds_user` where (`drds_user`.`name` = 'LiLei') | {} |
+-----+-----+-----+
8 rows in set (0.06 sec)
```

Logical or physical QPS indicates the total number of logical or physical SQL statements processed per unit of time. When most SQL statements in the system contain the shard key, the ratio of logical QPS to physical QPS is close to 1:1. If the difference between the logical and physical QPS is too large, many SQL statements of the current application do not contain the shard key. In this case, check the SQL statements of the application to improve performance.

13.4.9.4.4. Example 4: High memory usage

The overly high memory usage of the PolarDB-X instance is mostly caused by the large number of SQL queries in your application and the overlarge result set that is returned. If the memory usage of your PolarDB-X instance remains at about 100%, perform the [Restart a PolarDB-X instance](#) operations to locate and optimize the slow SQL queries of your application.

13.4.10. View the instance version

This topic describes two ways that you can use to view the version of a PolarDB-X instance.

View the instance version in the console

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the **Configuration Information** section, view the value of **Current Version**.

View the instance version by using the `version()` function

Connect to the PolarDB-X instance by using the MySQL command line and execute the `SELECT version()` statement to view the version of the PolarDB-X instance.

```
mysql> select version();
+-----+
| VERSION()          |
+-----+
| 5.6.29-TDDL-5.1.28-1320920 |
+-----+
1 row in set (0.00 sec)
```

In the preceding statement, 5.1.28-1320920 indicates the version of the PolarDB-X instance.

13.5. Account management

13.5.1. Basic concepts

This topic describes the terms of the PolarDB-X account and permission system.

The usage of the account and permission system in PolarDB-X is the same as in MySQL. PolarDB-X supports statements such as `GRANT`, `REVOKE`, `SHOW GRANTS`, `CREATE USER`, `DROP USER`, and `SET PASSWORD`. PolarDB-X allows you to grant permissions at the database and table levels, but does not allow you to grant permissions at the global or column level.

For more information about the MySQL account and permission system, see [MySQL Documentation](#).

Notice Accounts created by using `CREATE USER` in PolarDB-X exist only in the PolarDB-X instance and will not be synchronized to the backend ApsaraDB RDS for MySQL instances.

Account

An account is specified by the user name and hostname in the `username@'host'` format. Accounts with the same user name but different hostnames are different accounts. For example, `lily@30.9.73.96` and `lily@30.9.73.100` are two different accounts, and their passwords and permissions may be different.

After a database is created in the PolarDB-X console, the system automatically creates two system accounts for the database: the administrator account and read-only account. These two accounts are built-in accounts. You cannot delete them or modify their permissions.

- The administrator account name is the same as the database name. For example, if the database name is `easydb`, the administrator account name is also `easydb`.
- The read-only account name is the database name suffixed with `_RO`. For example, if the database name is `easydb`, the read-only account name is `easydb_RO`.

Assume that the `dreamdb` and `andoradb` databases are available. Based on the preceding rules, the `dreamdb` database contains the administrator account named `dreamdb` and the read-only account named `dreamdb_RO`, while the `andoradb` database contains the administrator account named `andoradb` and the read-only account named `andoradb_RO`.

Account rules

- Each administrator account has all permissions.
- Only the administrator account can create accounts and grant permissions. Other accounts can only be created and granted permissions by the administrator account.
- The administrator account is bound to a database and does not have permissions on other databases. It can access only the bound database, and cannot grant permissions of other databases to an account. For example, the `easydb` administrator account can connect only to the `easydb` database, and can grant only the permissions of the `easydb` database or tables in the `easydb` database to an account.
- A read-only account has only the `SELECT` permission.

User name rules

- User names are case-insensitive.
- A user name must be 4 to 20 characters in length.
- A user name must start with a letter.
- A user name can contain uppercase letters, lowercase letters, and digits.

Password rules

- A password must be 6 to 20 characters in length.
- A password can contain letters, digits, and special characters that include the at sign (`@`), number sign (`#`), dollar sign (`$`), percent sign (`%`), caret (`^`), ampersand (`&`), plus sign (`+`), equal sign (`=`).

Hostname matching rules

- A hostname must be an IP address. It can contain underscores (`_`) and wildcards (`%`). An underscore (`_`) indicates a character and a wildcard (`%`) indicates zero or more characters. Hostnames that contain wildcards must be quoted with single quotation marks (`'`), for example, `'lily@'30.9.%.%'` and `'david@'%'`.
- If two user names in the system can be used to log on to the database, the user name with the longest prefix (the longest IP segment excluding wildcards) prevails. For example, if the `'david@'30.9.12_.234'` and `'david@'30.9.1%.234'` user names are available in the system, use `'david@'30.9.12_.234'` to log on from the `30.9.127.234` host as `david`.
- When you enable the virtual private cloud (VPC) access feature for a host, the IP address of the host changes. To avoid invalid configurations in the account and permission system, set the hostname to `'%'` to match an IP address.

Permissions

Permission support by level

- Global permission (not supported)
- Database-level permission (supported)
- Table-level permission (supported)
- Column-level permission (not supported)

- Subprogram-level permission (not supported)

Permissions

Eight table-associated basic permissions are supported: `CREATE` , `DROP` , `ALTER` , `INDEX` , `INSERT` , `DELETE` , `UPDATE` , and `SELECT` .

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

Permission rules

- Permissions are bound to an account (`username@'host'`) rather than a user name (`username`).
- An error occurs if the table does not exist during authorization.
- The database permissions are listed by level in descending order: global permissions (not supported), database-level permissions, table-level permissions, and column-level permissions (not supported). A granted higher-level permission overwrites a lower-level permission. If you remove the higher-level permission, the lower-level permission is also removed.
- `USAGE` authorization is not supported.

13.5.2. Create an account

This topic describes how to create a PolarDB-X account in the PolarDB-X console and by using SQL statements.

Prerequisites

You have created or added a database. For more information, see [Create a database](#).

Create an account in the console

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Account Management**.
5. On the **Account Management** page, click **Create Account** in the upper-right corner.
6. Set the following parameters.

Parameter	Description
Database Account	Enter a name for the account. The account name must meet the following requirements: <ul style="list-style-type: none"> ◦ The name must be 4 to 20 characters in length. ◦ The name must start with a letter and end with a letter or digit. ◦ The name can contain uppercase letters, lowercase letters, digits, and underscores (_).

Parameter	Description
New Password	<p>Enter a password for the account. The password must meet the following requirements:</p> <ul style="list-style-type: none"> ○ The password must be 8 to 32 characters in length. ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ○ Special characters include exclamation marks (!). @#\$%^&*()_+ -=
Confirm New Password	Enter the password again.
Authorize Databases	<p>You can grant permissions on one or more databases to the account.</p> <ol style="list-style-type: none"> From the left-side section, select one or more databases. Then click Add to add them to the right-side section. In the right-side section, select Read/Write, Read-only, DDL Only, or DML Only for the specified database. <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note You can also grant permissions on multiple added databases by clicking Set All to Read-only, Set All to DDL Only, Set All to DML Only, or Set All to Read/Write in the upper-right corner of the right-side section.</p> <p>The buttons in the upper-right corner change after you click them. For example, after you click Set All to Read-only, this button is changed to Set All to DDL Only.</p> </div>

Create an account by using the command line

Use the following syntax rule:

```
CREATE USER user_specification [, user_specification] ...
user_specification: user [ auth_option ]
auth_option: IDENTIFIED BY 'auth_string'
```

For example:

Create an account whose name is lily and password is 123456 and which can be used to log on only from 30.xx.xx.96.

```
CREATE USER lily@30.xx.xx.96 IDENTIFIED BY '123456';
```

Create an account that is named david, has no password, and can be used to log on from all hosts.

```
CREATE USER david@'%';
```

13.5.3. Reset password

When you use PolarDB-X, you can reset the password of your database account in the PolarDB-X console or by using the command line.

Note

- Accounts that have root permissions cannot be deleted or modified.
- For data security, we recommend that you change your password on a regular basis.

Reset the password in the console

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Account Management**.
5. Find the target account and click **Reset Password**.
6. In the **Reset Account Password** dialog box, set **New Password** and **Confirm New Password**.

 **Note** The password must meet the following requirements:

- The password must be 8 to 32 characters in length.
- The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- Special characters include the following characters:
!@#%&^*()_+~

7. After you confirm that the password is correct, click **OK**.

Reset the password by using the command line

Use the following syntax rule:

```
SET PASSWORD FOR user = password_option
password_option: {
  PASSWORD('auth_string')
}
```

For example:

Change the password of the lily@30.xx.xx.96 account to 123456.

```
SET PASSWORD FOR lily@30.xx.xx.96 = PASSWORD('123456')
```

13.5.4. Modify account permissions

You can modify the account permissions of your instances at any time when you use PolarDB-X.

Precautions

- Privileged accounts cannot be modified.
- In the console, you can grant only DML, DDL, read-only, and read/write permissions to standard accounts. If you need more fine-grained authorization, use the command line.

Modify account permissions in the console

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Account Management**.
5. Find the target account and click **Modify Permission**.

6. In the **Modify Permissions** dialog box, grant or remove the permissions on one or more databases to or from the account.

- Add databases:

From the left-side section, select one or more databases. Then click **Add** to add them to the right-side section.

- Remove databases:

From the right-side section, select one or more databases. Then click **Remove** to move the databases to the left-side section.

- Modify permissions on added databases:

In the right-side section, select **Read/Write**, **Read-only**, **DDL Only** or **DML Only** for the specified database.

 **Note** You can also grant permissions on multiple added databases by clicking **Set All to Read-only**, **Set All to DDL Only**, **Set All to DML Only**, or **Set All to Read/Write** in the upper-right corner of the right-side section.

The buttons in the upper-right corner change after you click them. For example, after you click **Set All to Read-only**, this button is changed to **Set All to DDL Only**.

7. After the configuration is complete, click **OK**.

GRANT statement

Use the following syntax rule:

```
GRANT
  priv_type[, priv_type] ...
  ON priv_level
  TO user_specification [, user_specification] ...
  [WITH GRANT OPTION]
priv_level: {
  | db_name.*
  | db_name.tbl_name
  | tbl_name
}
user_specification:
  user [ auth_option ]
auth_option: {
  IDENTIFIED BY 'auth_string'
}
```

Notice

- If the account in the GRANT statement does not exist and no IDENTIFIED BY information is provided, an error message is returned, which indicates that the account does not exist.
- If the account specified in the GRANT statement does not exist but the IDENTIFIED BY information is provided, the account is created and granted with the specified permission.

For example, in the easydb database, create an account named david, which can be used to log on from all hosts and has all the permissions on easydb.

Method 1: Create an account and then grant permissions to the account.

```
CREATE USER david@%' IDENTIFIED BY 'your#password';
GRANT ALL PRIVILEGES ON easydb.* to david@%';
```

Method 2: Create an account and grant permissions to the account by executing a statement.

```
GRANT ALL PRIVILEGES ON easydb.* to david@%' IDENTIFIED BY 'your#password';
```

In the easydb database, create an account named hanson, which can be used to log on from all hosts and has all the permissions on the easydb.employees table.

```
GRANT ALL PRIVILEGES ON easydb.employees to hanson@%' IDENTIFIED BY 'your#password';
```

In the easydb database, create an account named hanson, which can be used to log on only from 192.xx.xx.10 and has the INSERT and SELECT permissions on the easydb.emp table.

```
GRANT INSERT,SELECT ON easydb.emp to hanson@'192.xx.xx.10' IDENTIFIED BY 'your#password';
```

In the easydb database, create a read-only account named actro, which can be used to log on from all hosts.

```
GRANT SELECT ON easydb.* to actro@%' IDENTIFIED BY 'your#password';
```

REVOKE statement

Use the following syntax rule:

- Delete specific permissions from an account: Delete the permissions at a certain level from an account. The permission level is specified by `priv_level`.

```
REVOKE
  priv_type
  [, priv_type] ...
  ON priv_level
  FROM user [, user] ...
```

- Delete all permissions from an account: Delete all permissions from the account at the database and table levels.

```
REVOKE ALL PRIVILEGES, GRANT OPTION
  FROM user [, user] ...
```

For example:

Delete the CREATE, DROP, and INDEX permissions from `hanson@'%'` on the `easydb.emp` table.

```
REVOKE CREATE,DROP,INDEX ON easydb.emp FROM hanson@%';
```

Delete all permissions from the `lily@30.xx.xx.96` account.

```
REVOKE ALL PRIVILEGES,GRANT OPTION FROM lily@30.xx.xx.96;
```



Notice GRANT OPTION must be added to the statement for compatibility with MySQL.

SHOW GRANTS statement

Use the following syntax rule:

```
SHOW GRANTS[FOR user@host];
```

Query all permissions:

```
SHOW GRANTS;
```

Query the permissions of an account:

```
SHOW GRANTS FOR user@host;
```

13.5.5. Delete an account

You can delete an account in the Cloud Native Distributed Database PolarDB-X (PolarDB-X) console or by using the command line.

Delete an account in the PolarDB-X console

 **Note** You can delete only standard accounts that are created in the console.

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Account Management**.
5. Find the target account and click **Delete**.
6. In the **Delete Account** dialog box, click **OK**.

Delete an account by using the command line

Use the following syntax rule:

```
DROP USER user [, user] ...
```

For example:

Delete the lily@30.xx.xx.96 account.

```
DROP USER lily@30.xx.xx.96;
```

13.6. Database management

13.6.1. Create a database

After you create a PolarDB-X instance, you must create a database that contains one or more ApsaraDB RDS for MySQL instances. You can create a database in four steps.

Prerequisites

- You have created an ApsaraDB RDS for MySQL instance in the same department of PolarDB-X.
- You have granted the Resource Access Management (RAM) permission. For more information, see *ASCM Console User Guide* for the *RAM* topic.

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. On the **Basic Information** page, click **Create Database** in the upper-right corner.
5. In the **Enter Basic Information** step, enter the following information.

Parameter	Description
Partition Mode	Select Horizontal Partitioning .
Database Name	Specify a name for the PolarDB-X database. The name must meet the following requirements: <ul style="list-style-type: none"> ◦ It must be 2 to 24 characters in length. ◦ It must start with a letter and end with a letter or digit. ◦ It can contain lowercase letters, digits, and underscores (_). ◦ It must be unique on the PolarDB-X instance.
Character Set	Select utf8, gbk, latin1, or utf8mb4 .
DRDS Link Password	Set the connection password for the PolarDB-X database. The rules are listed in the following content: <ul style="list-style-type: none"> ◦ The password must be 8 to 30 characters in length. ◦ The password must contain at least three of the following types: uppercase letters, lowercase letters, numbers, and underscores (_).
Confirm Password	Enter the password again.

6. Click **Next**.
7. In the **Select RDS Instance** step, you can click the **Buy New RDS Instance** or **Use Existing RDS Instance** tab.
 - i. **Buy New RDS Instance:** Click the **Buy New RDS Instance** tab.
 - ii. Select **Storage Type, Edition, Instance Specifications, Storage Capacity, Zone, and Quantity**.
 - iii. Click **Next**.
 - i. **Use Existing RDS Instance:** Click the **Use Existing RDS Instance** tab.
 - ii. In the left-side section, click the **ApsaraDB RDS for MySQL** instances to be added.
 - iii. Click to move the selected instances to the **Selected RDS Instances** section on the right.
 - iv. Click **Next**.
8. After all prechecks are passed in the **Precheck** step, click **Next**.

 **Note** If a precheck fails, rectify the configuration as prompted.

9. In the **Preview** step, click **Next**. Wait until the database is created.

13.6.2. View a database

After the database is created, you can view the basic information of the database on the **Database Management** page in the console.

Procedure

1. [Log on to the PolarDB-X console.](#)
2. Find the target instance in the instance list.

3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Database Management**.
5. Find the target database, and click **Manage** in the Actions column. The **Basic Information** page of the database appears.

 **Note** On the **Basic Information** page, you can delete the database or reset the password.

What's next

PolarDB-X is fully compatible with the MySQL protocol. You can use **Command Line URL** on the MySQL client to connect to the PolarDB-X instance and enter the user name and password to log on to the PolarDB-X database. When you use the MySQL client, note the following points:

Note

- MySQL clients of some earlier versions have limits on the user name length, which cannot exceed 16 characters. The PolarDB-X database name and user name are the same. If the database name exceeds 16 characters in length, an error is reported.
- When you use the MySQL client, you must add the `-c` parameter to the HINT command. In PolarDB-X, an annotation is used to implement HINT. If the `-c` parameter is not added, the annotation is lost and the HINT of PolarDB-X is lost.

13.6.3. Perform smooth scale-out

When the underlying storage of the logical database reaches the physical bottleneck, for example, when the remaining disk space is about 30%, you can smoothly scale it out to improve the performance. The smooth scale-out process is divided into four steps: configuration > migration > switchover > cleanup.

Configuration

 **Note** In smooth scale-out, ApsaraDB RDS for MySQL instances are added, and some source database shards are migrated to the new ApsaraDB RDS for MySQL instances. In this way, the overall data storage capacity is increased and the number of requests that a single ApsaraDB RDS for MySQL instance needs to process is reduced.

1. **Log on to the PolarDB-X console.**
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Database Management**.
5. Find the target database, and click **Manage**. in the Actions column The **Basic Information** page of the database appears.
6. In the left-side navigation pane, choose **Configuration and Management > Scale-out Management**.
7. In the upper-right corner of the **Scale-out Management** page, click **Scale Out**.
8. Select **Smooth Scale-out**, and then click **Next**.
9. After all prechecks are passed in the **Precheck** step, click **Next**.

 **Note** If a precheck fails, rectify the configuration as prompted.

10. In the **Select RDS Instance** step, you can click the **Buy New RDS Instance** or **Use Existing RDS Instance** tab.
 - i. **Buy New RDS Instance:** Click the **Buy New RDS Instance** tab.

- ii. Select **Storage Type, Edition, Instance Specifications, Storage Capacity, Zone, and Quantity**.
 - iii. Click **Next**.
 - i. **Use Existing RDS Instance:** Click the **Use Existing RDS Instance** tab.
 - ii. In the left-side section, click the **ApsaraDB RDS for MySQL** instances to be added.
 - iii. Click  to move the selected instances to the **Selected RDS Instances** section on the right.
 - iv. Click **Next**.
11. In the **Preview** step, click **Start Scale-out**.

 **Note** By default, the console evenly distributes the physical database shards to the ApsaraDB RDS for MySQL instances you added. You can also manually add or delete physical database shards to or from the new ApsaraDB RDS for MySQL instances.

12. Click the  icon in the upper-right corner to view the progress of the scale-out task.

Migration

Some physical database shards are migrated during smooth scale-out.

The migration does not change the data in the source database and therefore does not affect online services. Before the switchover, you can cancel the smooth scale-out operation by using a rollback.

Note

- This is because before the switchover, the current scale-out operation does not have a real impact on the data in the source database.
- During scale-out, the binary log files of the source RDS instance are not cleaned, which may result in insufficient disk space. Therefore, you must reserve sufficient disk space on the source ApsaraDB RDS for MySQL instance. Generally, the remaining disk space needs to be more than 30%. If the disk space cannot be guaranteed, you can submit a ticket to expand the storage space of the ApsaraDB RDS for MySQL instance.
- To reduce the pressure of read operations on the source RDS instance, perform scale-out when the load on the source ApsaraDB RDS for MySQL instance is low.
- During the scale-out, do not submit data description language (DDL) tasks in the console or connect to the PolarDB-X instance to directly run DDL statements. Otherwise, the scale-out task may fail.
- Make sure that all tables in the source database have primary keys before scale-out.

After historical data and incremental data are migrated, the migration progress reaches 100%. Then, you can switch the read/write traffic to the new ApsaraDB RDS for MySQL instance or roll back to cancel this scale-out.

Switchover

The switchover task switches the read/write traffic to the new ApsaraDB RDS for MySQL instance. The whole process takes three to five minutes. During the switchover process, the service is not affected except for one or two transient disconnections. Perform the switchover during off-peak hours.

1. In the upper-right corner of the **Basic Information** page, click the  icon. The **Task Progress** dialog box appears.
2. In the **Task Progress** dialog box, click **Switch Over** and then click **OK**.
During the switchover process, a switchover task is generated and appears in the task progress.
3. After the switchover is completed, the **Clean Up** button appears in the **Task Progress** dialog box.

Cleanup

In this step, the migrated database shards are deleted from the source ApsaraDB RDS for MySQL instance.

1. After switchover is completed, click **Clean Up** next to the target task.
2. Click **OK**. A cleanup task appears in the Task Progress dialog box.

 **Note**

- The cleanup task is an asynchronous task. You can view the execution status in the Task Progress dialog box.
- After the cleanup task is completed, the smooth scale-out process ends. The new ApsaraDB RDS for MySQL instance becomes the storage node of the PolarDB-X logical database.
- You can implement smooth scale-out by migrating physical database shards. If no further scale-out is allowed after the number of database shards exceeds the capacity of a single ApsaraDB RDS for MySQL instance, you can submit a ticket to apply for increasing the number of database shards and scaling out the database. In this case, calculation based on the hash algorithm is performed again to reallocate data.
- The cleanup task deletes database shards that are no longer used after the current scale-out. You can back up the database shards before you run the cleanup task.
- The cleanup operation brings pressure to databases. We recommend that you perform this operation during off-peak hours.

13.6.4. View database monitoring information

PolarDB-X displays the historical monitoring information of a PolarDB-X database in two dimensions: data metrics and query time.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Monitoring and Alerts > Database Monitoring**.
5. Select the target database, and then set **Data Indexes** and **Query Time**. You can see the corresponding monitoring information.

 **Note** For more information about instance-level monitoring, see [View monitoring information](#).

13.6.5. Set the IP address whitelist

PolarDB-X provides the access control feature. Only IP addresses in the whitelist of a database can be used to access the database.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Database Management**.
5. Find the target database, and click **Manage** in the Actions column. The **Basic Information** page of the database appears.
6. On the **Basic Information** page of the database, choose **Data Security > Whitelist Settings** in the left-side navigation pane.

7. On the Whitelist Settings page, click **Manually Modify**.
8. Enter the IP addresses that are allowed to access the database, and click **OK**.

 **Note**

- The following formats are supported in the whitelist:
 - Single IP addresses, for example, 192.168.1.1.
 - IP addresses in CIDR format, for example, 192.168.1.1/24.
 - IP addresses that includes an asterisk (*) as a wildcard, for example, 192.168.1.*. This example indicates that access is allowed from a host with an IP address in the range from 192.168.1.1 to 192.168.1.254.
 - IP range, for example, 192.168.1.1-192.168.1.254.
- If you need to add multiple IP addresses or IP ranges, separate them with commas (,) and do not use spaces before and after the commas, for example, 192.168.0.1,172.16.213.9.

13.6.6. Delete a database

This topic describes how to delete a database in the Cloud Native Distributed Database PolarDB-X (PolarDB-X) console.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Database Management**.
5. Find the target database and click **Delete**.

 **Warning** You cannot recover databases that have been deleted. Exercise caution when you perform this operation.

6. In the **Delete Database** dialog box, click **OK**.

13.6.7. Fix database shard connections

Context

When you use a PolarDB-X instance, you must access ApsaraDB RDS for MySQL. If the network configuration of the connected ApsaraDB RDS for MySQL instance changes, for example, if the zone is switched or the network type is changed from classic network to VPC, the network connection between the PolarDB-X instance and the ApsaraDB RDS for MySQL instance is broken. As a result, the PolarDB-X instance cannot access the ApsaraDB RDS for MySQL instance. In this case, you must manually fix the database shard link in the PolarDB-X console to restore the network connection from the PolarDB-X instance to the ApsaraDB RDS for MySQL instance.

Procedure

1. [Log on to the PolarDB-X console](#).
2. Find the target instance in the instance list.
3. Click the target instance ID or choose **More > Manage** from the Actions column of the instance to access the **Basic Information** page.
4. In the left-side navigation pane, choose **Configuration and Management > Database Management**.
5. Find the target database, and click **Manage** in the Actions column. The **Basic Information** page of the

database appears.

6. In the **Shortcuts** section, click **Fix Database Shard Connections**.

7. In the dialog box that appears, click **OK**.

13.7. Custom control commands

PolarDB-X provides a series of auxiliary SQL commands to help you conveniently use PolarDB-X.

13.7.1. Overview

PolarDB-X provides unique auxiliary statements for you to use and maintain PolarDB-X.

Syntax description: The identifier provided by the user is in [] and optional content is in (). In addition, this document applies to the current version. If some statements are unavailable, the version is earlier than required.

13.7.2. Help statements

This topic describes all the auxiliary SQL commands of PolarDB-X and their descriptions.

SHOW HELP statements:

```
mysql> show help;
+-----+
| STATEMENT          | DESCRIPTION                               | EXAMPLE                               |
+-----+
| show rule          | Report all table rule                     |                                       |
| show rule from TABLE | Report table rule                         | show rule from user                   |
| show full rule from TABLE | Report table full rule                   | show full rule from user             |
| show topology from TABLE | Report table physical topology           | show topology from user              |
|
| show partitions from TABLE | Report table dbPartition or tbPartition columns | show partitions from user            |
|
| show broadcasts    | Report all broadcast tables               |                                       |
| show datasources   | Report all partition db threadPool info  |                                       |
| show node          | Report master/slave read status          |                                       |
| show slow          | Report top 100 slow sql                  |                                       |
| show physical_slow | Report top 100 physical slow sql         |                                       |
| clear slow         | Clear slow data                          |                                       |
| trace SQL          | Start trace sql, use show trace to print profiling data | trace select count(*) from user; show trace |
| show trace         | Report sql execute profiling info        |                                       |
| explain SQL        | Report sql plan info                     | explain select count(*) from user    |
| explain detail SQL | Report sql detail plan info              | explain detail select count(*) from user |
| explain execute SQL | Report sql on physical db plan info      | explain execute select count(*) from user |
| show sequences     | Report all sequences status              |                                       |
| create sequence NAME [start with COUNT] | Create sequence                               | create sequence test start with 0    |
| alter sequence NAME [start with COUNT] | Alter sequence                               | alter sequence test start with 1000  |
| drop sequence NAME | Drop sequence                             | drop sequence test                   |
+-----+
-----+
20 rows in set (0.00 sec)
```

13.7.3. Statements for viewing rules and node topologies

SHOW RULE [FROM tablename]

Usage notes:

- `show rule` : shows the partitioning status of each logical table in a database.
- `show rule from tablename` : shows the partitioning status of a specified logical table in a database.

The following describes the meanings of important columns:

- **BROADCAST**: indicates whether the table is a broadcast table. 0 indicates "No" and 1 indicates "Yes".

- **DB_PARTITION_KEY**: indicates the database shard key. If no database shards exist, the parameter value is NULL.
- **DB_PARTITION_POLICY**: indicates the database sharding policy. Options are Hash and date policies such as YYYYMM, YYYYDD, and YYYYWEEK.
- **DB_PARTITION_COUNT**: indicates the number of database shards.
- **TB_PARTITION_KEY**: indicates the table shard key. If no table shards exist, the parameter value is NULL.
- **TB_PARTITION_POLICY**: indicates the table sharding policy. Options are Hash and date policies such as MM, DD, MMDD, and WEEK.
- **TB_PARTITION_COUNT**: indicates the number of table shards.

```
mysql> show rule;
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_
KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| 0 | dept_manager | 0 | NULL | 1 | NULL | 1 | | |
| 1 | emp | 0 | emp_no | hash | 8 | id | hash | 2 |
| 2 | example | 0 | shard_key | hash | 8 | NULL | NULL | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
3 rows in set (0.01 sec)
```

SHOW FULL RULE [FROM tablename]

You can run this SQL statement to view the sharding rules of logical tables in a database. It displays more detailed information than the SHOW RULE command.

The following describes the meanings of important columns:

- **BROADCAST**: indicates whether the table is a broadcast table. 0 indicates "No" and 1 indicates "Yes".
- **JOIN_GROUP**: a reserved field.
- **ALLOW_FULL_TABLE_SCAN**: indicates whether to allow data querying when no table shard key is specified for database or table sharding. If this parameter is set to True, each physical table is scanned to find data that meets the condition, which is a full table scan.
- **DB_NAME_PATTERN**: The value 0 between {} in DB_NAME_PATTERN is a placeholder. When the SQL statement is run, this value is replaced by the value of DB_RULES_STR, with the number of digits unchanged. For example, if the value of DB_NAME_PATTERN is SEQ_{0000}_RDS and the value of DB_RULES_STR is [1,2,3,4], four DB_NAME values are generated: SEQ_0001_RDS, SEQ_0002_RDS, SEQ_0003_RDS, and SEQ_0004_RDS.
- **DB_RULES_STR**: indicates the database sharding rule.
- **TB_NAME_PATTERN**: The value 0 between {} in TB_NAME_PATTERN is a placeholder. When the SQL statement is run, this value is replaced by the value of TB_RULES_STR, with the number of digits unchanged. For example, if the value of TB_NAME_PATTERN is table_{00} and the value of TB_RULES_STR is [1,2,3,4,5,6,7,8], eight tables are generated: table_01, table_02, table_03, table_04, table_05, table_06, table_07, and table_08.
- **TB_RULES_STR**: indicates the table sharding rule.
- **PARTITION_KEYS**: indicates a set of database and table shard keys. When database sharding and table sharding coexist, the database shard key is placed before the table shard key.
- **DEFAULT_DB_INDEX**: indicates the database shard in which a single database and a single table are stored.

```
mysql> show full rule;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | TABLE_NAME | BROADCAST | JOIN_GROUP | ALLOW_FULL_TABLE_SCAN | DB_NAME_PATTERN | D
B_RULES_STR | TB_NAME_PATTERN | TB_RULES_STR | PARTITION_KEYS | DEFAULT_DB_I
NDEX |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | dept_manager | 0 | NULL | 0 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS | NULL
| dept_manager | NULL | NULL | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |
| 1 | emp | 0 | NULL | 1 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_{0000}_RDS | ((#emp_no
,1,8#).longValue().abs() % 8) | emp_{0} | ((#id,1,2#).longValue().abs() % 2) | emp_no id | SEQ_TEST_148776
7780814RGKKSEQ_TEST_WNJG_0000_RDS |
| 2 | example | 0 | NULL | 1 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_{0000}_RDS | ((#shard
_key,1,8#).longValue().abs() % 8).intdiv(1) | example | NULL | shard_key | SEQ_TEST_1487767780
814RGKKSEQ_TEST_WNJG_0000_RDS |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

SHOW TOPOLOGY FROM tablename

You can run this SQL statement to view the topology of a specified logical table, that is, the database shards to which data in the logical table is partitioned and the table shards in each database shard.

```
mysql> show topology from emp;
+-----+-----+
| ID | GROUP_NAME          | TABLE_NAME |
+-----+-----+
| 0 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS | emp_0 |
| 1 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS | emp_1 |
| 2 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS | emp_0 |
| 3 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS | emp_1 |
| 4 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS | emp_0 |
| 5 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS | emp_1 |
| 6 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS | emp_0 |
| 7 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS | emp_1 |
| 8 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS | emp_0 |
| 9 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS | emp_1 |
| 10 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS | emp_0 |
| 11 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS | emp_1 |
| 12 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS | emp_0 |
| 13 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS | emp_1 |
| 14 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS | emp_0 |
| 15 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS | emp_1 |
+-----+-----+
16 rows in set (0.01 sec)
```

SHOW PARTITIONS FROM tablename

You can run this SQL statement to view the set of database shard keys and table shard keys, which are separated by commas (.). If the final result contains two values, both database sharding and table sharding are performed. The first value is the database shard key and the second value is the table shard key. If only one value is returned, only database sharding is performed. This value is the database shard key.

```
mysql> show partitions from emp;
+-----+
| KEYS |
+-----+
| emp_no,id |
+-----+
1 row in set (0.00 sec)
```

SHOW BROADCASTS

You can run this SQL statement to view the list of broadcast tables.

```
mysql> show broadcasts;
+-----+-----+
| ID | TABLE_NAME |
+-----+-----+
| 0 | brd2 |
| 1 | brd_tbl |
+-----+-----+
2 rows in set (0.01 sec)
```

SHOW DATASOURCES

You can run this SQL statement to view the information about the underlying storage, including the database name, database group name, connection URL, user name, storage type, read/write weight, and connection pool information.

The following describes the meanings of important columns:

- **SCHEMA:** indicates the database name.
- **GROUP:** indicates the database group name. Grouping aims to manage multiple groups of databases that have identical data, such as the primary and secondary databases after data replication through ApsaraDB RDS for MySQL. It is mainly used for read/write splitting and primary/secondary switchovers.
- **URL:** indicates the connection information of the underlying ApsaraDB RDS for MySQL instance.
- **TYPE:** indicates the type of the underlying storage. Currently, only ApsaraDB RDS for MySQL instances are supported.
- **READ_WEIGHT:** indicates the read weight of the database. When the primary ApsaraDB RDS for MySQL instance is under a heavy load of many read requests, you can use the read/write splitting function of PolarDB-X to distribute the read traffic. PolarDB-X automatically identifies the read and write traffic. It directs the write traffic to the primary ApsaraDB RDS for MySQL instance and the read traffic to all ApsaraDB RDS for MySQL instances based on the configured weight.
- **WRITE_WEIGHT:** indicates the write weight. For more information, see **READ_WEIGHT**.

```
mysql> show datasources;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | SCHEMA          | NAME                               | GROUP          | URL
| USER | TYPE | INIT | MIN | MAX | IDLE_TIMEOUT | MAX_WAIT | ACTIVE_COUNT | POOLING_COUNT | ATOM
| READ_WEIGHT | WRITE_WEIGHT |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0000_iiab_1 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0000_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0000 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0000_iiab | 10
| 10 |
| 1 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0001_iiab_2 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0001_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0001 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0001_iiab | 10
| 10 |
| 2 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0002_iiab_3 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0002_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0002 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0002_iiab | 10
| 10 |
| 3 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0003_iiab_4 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0003_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0003 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0003_iiab | 10
| 10 |
| 4 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0004_iiab_5 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0004_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0004 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0004_iiab | 10
| 10 |
| 5 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0005_iiab_6 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0005_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0005 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0005_iiab | 10
| 10 |
| 6 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0006_iiab_7 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0006_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0006 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0006_iiab | 10
| 10 |
| 7 | seq_test_1487767780814rgkk | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0007_iiab_8 | SEQ_TEST_1487767780814RGKKS
EQ_TEST_WNJG_0007_RDS | jdbc:mysql://rds1ur80kcv8g3t6p3ol.mysql.rds.aliyuncs.com:3306/seq_test_wnjg_0007 | jnkins
ea0 | mysql | 0 | 24 | 72 | 15 | 5000 | 0 | 1 | rds1ur80kcv8g3t6p3ol_seq_test_wnjg_0007_iiab | 10
| 10 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

SHOW NODE

You can run this SQL statement to view the accumulative number of read and write operations and accumulative read/write weights of a physical database.

The following describes the meanings of important columns:

- **MASTER_READ_COUNT**: indicates the accumulative number of read-only queries processed by the primary ApsaraDB RDS for MySQL instance.
- **SLAVE_READ_COUNT**: indicates the accumulative number of read-only queries processed by the secondary ApsaraDB RDS for MySQL instances.
- **MASTER_READ_PERCENT**: indicates the actual percentage of read-only queries processed by the primary ApsaraDB RDS for MySQL instance, not the configured percentage.
- **SLAVE_READ_PERCENT**: indicates the actual percentage of read-only queries processed by the secondary ApsaraDB RDS for MySQL instances, not the configured percentage.

Note

- Read-only queries in transactions are sent to the primary ApsaraDB RDS for MySQL instance.
- The `MASTER_READ_PERCENT` and `SLAVE_READ_PERCENT` fields indicate the accumulative historical data. After the read/write weight ratio has been changed, these values do not immediately reflect the latest read/write weight ratio, which appears after a long period of time has passed.

```
mysql> show node;
+-----+-----+-----+-----+-----+-----+
--+
| ID | NAME                               | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_READ_PERCENT |
+-----+-----+-----+-----+-----+-----+
--+
| 0 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0000_RDS |          12 |           0 | 100%                | 0%                 |
| 1 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0001_RDS |           0 |           0 | 0%                   | 0%                 |
| 2 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0002_RDS |           0 |           0 | 0%                   | 0%                 |
| 3 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0003_RDS |           0 |           0 | 0%                   | 0%                 |
| 4 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0004_RDS |           0 |           0 | 0%                   | 0%                 |
| 5 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0005_RDS |           0 |           0 | 0%                   | 0%                 |
| 6 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0006_RDS |           0 |           0 | 0%                   | 0%                 |
| 7 | SEQ_TEST_1487767780814RGKKSEQ_TEST_WNJG_0007_RDS |           0 |           0 | 0%                   | 0%                 |
+-----+-----+-----+-----+-----+-----+
--+
8 rows in set (0.01 sec)
```

13.7.4. SQL tuning statements

SHOW [FULL] SLOW [WHERE expr] [limit expr]

SQL statements that take more than 1 second to execute are slow SQL statements. Slow logical SQL statements are the slow SQL statements sent from an application to a PolarDB-X instance.

- **SHOW SLOW**: You can run this SQL statement to view the 100 slowest logical SQL queries that are recorded since the PolarDB-X instance is started or the last time when `CLEAR SLOW` is executed.

 **Note** The recorded 100 slowest SQL queries are stored in the PolarDB-X system. When the PolarDB-X instance is restarted or executes `CLEAR SLOW`, these queries will be discarded.

- **SHOW FULL SLOW** : You can run this SQL statement to view all the slow logical SQL queries that are recorded and persisted to the built-in database of the PolarDB-X instance since the PolarDB-X instance is started. The upper limit for the number of records is specified in the specifications of the PolarDB-X instance. The PolarDB-X instance scrolls to delete the earliest slow SQL statements. If the specifications of the PolarDB-X instance is 4-core 4 GB, a maximum of 10,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. If the specifications of the PolarDB-X instance is 8-core 8 GB, a maximum of 20,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. The same rule applies to other specifications.

The following describes the meanings of important columns:

- **HOST**: the IP address of the host from which the SQL statement is sent.
- **START_TIME**: the time when the SQL statement starts to be executed.
- **EXECUTE_TIME**: the time when the SQL statement is executed.
- **AFFECT_ROW**: For data manipulation language (DML) statements, this parameter indicates the number of affected rows. For query statements, this parameter indicates the number of returned records.

```
mysql> show slow where execute_time > 1000 limit 1;
+-----+-----+-----+-----+-----+
| HOST   | START_TIME   | EXECUTE_TIME | AFFECT_ROW | SQL   |
+-----+-----+-----+-----+-----+
| 127.0.0.1 | 2016-03-16 13:02:57 | 2785 | 7 | show rule |
+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

SHOW [FULL] PHYSICAL_SLOW [WHERE expr] [limit expr]

SQL statements that take more than 1 second to execute are slow SQL statements. Slow logical SQL statements are the slow SQL statements sent from an application to a PolarDB-X instance.

- **SHOW SLOW** : You can run this SQL statement to view the 100 slowest logical SQL queries that are recorded since the PolarDB-X instance is started or the last time when `CLEAR SLOW` is executed.

 **Note** The recorded 100 slowest SQL queries are stored in the PolarDB-X system. When the PolarDB-X instance is restarted or executes `CLEAR SLOW`, these queries will be discarded.

- **SHOW FULL SLOW** : You can run this SQL statement to view all the slow logical SQL queries that are recorded and persisted to the built-in database of the PolarDB-X instance since the PolarDB-X instance is started. The upper limit for the number of records is specified in the specifications of the PolarDB-X instance. The PolarDB-X instance scrolls to delete the earliest slow SQL statements. If the specifications of the PolarDB-X instance is 4-core 4 GB, a maximum of 10,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. If the specifications of the PolarDB-X instance is 8-core 8 GB, a maximum of 20,000 slow SQL statements can be recorded, including slow logical and physical SQL statements. The same rule applies to other specifications.

The following describes the meanings of important columns:

- **GROUP_NAME**: the name of the group to which the database that executes the SQL statement belongs.
- **START_TIME**: the time when the SQL statement starts to be executed.
- **EXECUTE_TIME**: the time when the SQL statement is executed.
- **AFFECT_ROW**: For data manipulation language (DML) statements, this parameter indicates the number of affected rows. For query statements, this parameter indicates the number of returned records.

```
mysql> show physical_slow;
+-----+-----+-----+-----+-----+-----+
| GROUP_NAME | DBKEY_NAME          | START_TIME      | EXECUTE_TIME | SQL_EXECUTE_TIME | GETLOCK_CONNE  
CTION_TIME | CREATE_CONNECTION_TIME | AFFECT_ROW | SQL      |
+-----+-----+-----+-----+-----+-----+
| TDDL5_00_GROUP | db218249098_sqa_zmf_tddl5_00_3309 | 2016-03-16 13:05:38 | 1057 | 1011 | 0 |
0 | 1 | select sleep(1) |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

CLEAR SLOW

You can run this SQL statement to clear the 100 slowest logical SQL queries and the 100 slowest physical SQL queries that are recorded since the PolarDB-X instance is started or the last time when `CLEAR SLOW` is executed.

 **Note** Both `SHOW SLOW` and `SHOW PHYSICAL_SLOW` can be executed to display the 100 slowest SQL statements. If `CLEAR SLOW` has not been executed for a long time, these SQL statements may have been recorded a long time ago. Therefore, after SQL tuning statements are executed, we recommend that you execute `CLEAR SLOW`. After the system runs for a while, check the tuning results of slow SQL statements.

```
mysql> clear slow;
Query OK, 0 rows affected (0.00 sec)
```

EXPLAIN SQL

You can run this SQL statement to view the execution plan of a specified SQL statement in the PolarDB-X. Note that this SQL statement is not truly executed.

Example:

You can run this SQL statement to view the execution plan of the SQL `select * from doctest` statement. The `doctest` table is stored in database shards according to values in the `id` column. According to the execution plan, the SQL statement will be routed to each database shard for execution, and the execution results will be aggregated.

```
mysql> explain select * from doctest;
+-----+-----+-----+
| GROUP_NAME          | SQL                                | PARAMS |
+-----+-----+-----+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0000_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0002_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0003_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0004_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0005_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0006_RDS | select `doctest`.`id` from `doctest` | {} |
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0007_RDS | select `doctest`.`id` from `doctest` | {} |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

You can run this SQL statement to view the execution plan of the SQL `select * from doctest where id = 1` statement. The doctest table is stored in database shards according to values in the id column. According to the execution plan, the PolarDB-X instance will calculate a specified database shard based on the shard key, which is id, directly route the SQL statement to the database shard, and aggregate the execution results.

```
mysql> explain select * from doctest where id = 1;
+-----+-----+-----+
| GROUP_NAME          | SQL                                | PARAMS |
+-----+-----+-----+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | select `doctest`.`id` from `doctest` where (`doctest`.`id` = 1) | {} |
+-----+-----+-----+
1 row in set (0.01 sec)
```

EXPLAIN DETAIL SQL

You can run this SQL statement to view the execution plan of a specified SQL statement in the PolarDB-X. Note that this SQL statement is not truly executed.

```
mysql> explain detail select * from doctest where id = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+
| GROUP_NAME          | SQL                                     |
| PARAMS              |
+-----+-----+-----+-----+-----+-----+-----+
| DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS | Query from doctest as doctest
  keyFilter:doctest.id = 1
  queryConcurrency:SEQUENTIAL
  columns:[doctest.id]
  tableName:doctest
  executeOn:DOCTEST_1488704345426RCUPDOCTEST_CAET_0001_RDS
| NULL |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

EXPLAIN EXECUTE SQL

You can run this SQL statement to view the execution plan of underlying storage. This statement is equivalent to the MySQL EXPLAIN statement.

```
mysql> explain execute select * from tddl_mgr_log limit 1;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table      | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | tddl_mgr_log | ALL | NULL          | NULL | NULL    | NULL | 1 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.07 sec)
```

TRACE SQL and SHOW TRACE

You can run these SQL statements to view the execution results of an SQL statement. Note that you must use TRACE SQL and SHOW TRACE together. The difference between TRACE SQL and EXPLAIN SQL is that TRACE SQL is truly executed.

For example, you can run these statements to view the execution results of the select 1 statement.

```
mysql> trace select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.03 sec)
mysql> show trace;
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| ID | TYPE | GROUP_NAME | DBKEY_NAME | TIME_COST(MS) | CONNECTION_TIME_COST(MS) | ROWS | STATEMENT | PARAMS |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| 0 | Optimize | DRDS | DRDS | 3 | 0.00 | 0 | select 1 | NULL |
| 1 | Query | TDDL5_00_GROUP | db218249098_sqa_zmf_tddl5_00_3309 | 7 | 0.15 | 1 | select 1 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
2 rows in set (0.01 sec)
```

CHECK TABLE tablename

You can run this SQL statement to check a data table. This SQL statement is mainly used when a table failed to be created by using a data definition language (DDL) statement.

- If the data table is a table shard, this SQL statement allows you to check whether any underlying physical table shard is missing and whether the columns and indexes of the underlying physical table are consistent.
- If the data table is a single-database non-partition table, this SQL statement allows you to check whether this table exists.

```
mysql> check table tddl_mgr_log;
+-----+-----+-----+-----+
| TABLE | OP | MSG_TYPE | MSG_TEXT |
+-----+-----+-----+-----+
| TDDL5_APP.tddl_mgr_log | check | status | OK |
+-----+-----+-----+-----+
1 row in set (0.56 sec)
mysql> check table tddl_mg;
+-----+-----+-----+-----+
| TABLE | OP | MSG_TYPE | MSG_TEXT |
+-----+-----+-----+-----+
| TDDL5_APP.tddl_mg | check | Error | Table 'tddl5_00.tddl_mg' doesn't exist |
+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

SHOW TABLE STATUS [LIKE 'pattern' | WHERE expr]

You can run this SQL statement to obtain the information about a table. This command aggregates the data of all underlying physical table shards.

The following describes the meanings of important columns:

- **NAME:** indicates the name of the table.
- **ENGINE:** indicates the storage engine of the table.
- **VERSION:** indicates the version of the storage engine of the table.
- **ROW_FORMAT:** indicates the format of the rows in the table. Valid values include Dynamic, Fixed, and Compressed. The value Dynamic indicates that the row length is variable, for example, is a VARCHAR or BLOB field. The value Fixed indicates that the row length is constant, for example, is a CHAR or INTEGER field.
- **ROWS:** indicates the number of rows in the table.
- **AVG_ROW_LENGTH:** indicates the average number of bytes in each row.
- **DATA_LENGTH:** indicates the data volume of the entire table, in bytes.
- **MAX_DATA_LENGTH:** indicates the maximum volume of data that can be stored in the table.
- **INDEX_LENGTH:** indicates the size of the disk space occupied by indexes.
- **CREATE_TIME:** indicates the time when the table was created.
- **UPDATE_TIME:** indicates the time when the table was last updated.
- **COLLATION:** indicates the default character set and character sorting rule of the table.
- **CREATE_OPTIONS:** indicates all the other options specified when the table was created.

```
mysql> show table status like 'multi_db_multi_tbl';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME          | ENGINE | VERSION | ROW_FORMAT | ROWS | AVG_ROW_LENGTH | DATA_LENGTH | MAX_DATA_LENGTH | INDEX_LENGTH | DATA_FREE | AUTO_INCREMENT | CREATE_TIME          | UPDATE_TIME | CHECK_TIME | COLLATION      | CHECKS |
UM | CREATE_OPTIONS | COMMENT |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| multi_db_multi_tbl | InnoDB | 10 | Compact | 2 | 16384 | 16384 | 0 | 16384 | 0 | 100000 | 2017-03-27 17:43:57.0 | NULL | NULL | utf8_general_ci | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

The combination of the SHOW TABLE STATUS statement and the PolarDB-X SCAN hint allows you to view the data volume of each physical table shard.

```
mysql> /*! TDDL:SCAN='multi_db_multi_tbl'*/show table status like 'multi_db_multi_tbl';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
---+
| Name          | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length
| Data_free | Auto_increment | Create_time   | Update_time | Check_time | Collation   | Checksum | Create_options |
Comment | Block_format |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
---+
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 1 | 16384 | 16384 | 0 | 16384 | 0 | 2 | 2
017-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_1 | InnoDB | 10 | Compact | 0 | 0 | 16384 | 0 | 16384 | 0 | 1 | 201
7-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
| multi_db_multi_tbl_0 | InnoDB | 10 | Compact | 1 | 16384 | 16384 | 0 | 16384 | 0 | 3 | 2
017-03-27 17:43:57 | NULL | NULL | utf8_general_ci | NULL | | | Original |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
---+
16 rows in set (0.04 sec)
```

13.7.5. Statistics query statements

SHOW [FULL] STATS

You can run this SQL statement to view the overall statistics. The statistics are instantaneous values. Note that the results of `SHOW FULL STATS` vary with the version.

The following describes the meanings of important columns:

- **QPS**: the number of queries per second (QPS) sent from an application to the PolarDB-X instance. These queries are usually called logical QPS.
- **RDS_QPS**: the number of QPS sent from the PolarDB-X instance to an ApsaraDB RDS for MySQL instance. These queries are usually called physical QPS.
- **ERROR_PER_SECOND**: the total number of errors that occur on the PolarDB-X instance per second. These errors include SQL syntax errors, primary key conflicts, system errors, and connectivity errors.
- **VIOLATION_PER_SECOND**: the number of conflicts that occur on primary keys or unique keys per second.
- **MERGE_QUERY_PER_SECOND**: the number of queries processed on multiple tables through database sharding and table sharding per second.
- **ACTIVE_CONNECTIONS**: the number of active connections to the PolarDB-X instance.
- **CONNECTION_CREATE_PER_SECOND**: the number of connections that are created for the PolarDB-X instance per second.
- **RT(MS)**: the time to respond to an SQL query sent from an application to the PolarDB-X instance. This response time (RT) is usually called logical RT.
- **RDS_RT(MS)**: the time to respond to an SQL query sent from the PolarDB-X instance to an ApsaraDB RDS for MySQL instance. This RT is usually called physical RT.
- **NET_IN(KB/S)**: the amount of inbound traffic of the PolarDB-X instance per second.
- **NET_OUT(KB/S)**: the amount of outbound traffic of the PolarDB-X instance per second.
- **THREAD_RUNNING**: the number of threads that are running in the PolarDB-X instance.
- **HINT_USED_PER_SECOND**: the number of SQL queries that contain hints and are processed by the PolarDB-X instance per second.
- **HINT_USED_COUNT**: the total number of SQL queries that contain hints and have been processed by the PolarDB-X instance since startup.
- **AGGREGATE_QUERY_PER_SECOND**: the number of aggregate SQL queries processed by the PolarDB-X instance per second.
- **AGGREGATE_QUERY_COUNT**: the total number of aggregate SQL queries that have been processed by the PolarDB-X instance.
- **TEMP_TABLE_CREATE_PER_SECOND**: the number of temporary tables created in the PolarDB-X instance per second.
- **TEMP_TABLE_CREATE_COUNT**: the total number of temporary tables that have been created in the PolarDB-X instance since startup.
- **MULTI_DB_JOIN_PER_SECOND**: the number of multi-database JOIN queries processed by the PolarDB-X instance per second.
- **MULTI_DB_JOIN_COUNT**: the number of multi-database JOIN queries that have been processed by the PolarDB-X instance since startup.

```
mysql> show stats;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| QPS | RDS_QPS | SLOW_QPS | PHYSICAL_SLOW_QPS | ERROR_PER_SECOND | MERGE_QUERY_PER_SECOND | ACTIVE_CONNECTIONS | RT(MS) | RDS_RT(MS) | NET_IN(KB/S) | NET_OUT(KB/S) | THREAD_RUNNING |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1.77 | 1.68 | 0.03 | 0.03 | 0.02 | 0.00 | 7 | 157.13 | 51.14 | 134.49 | 1.48 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> show full stats;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| QPS | RDS_QPS | SLOW_QPS | PHYSICAL_SLOW_QPS | ERROR_PER_SECOND | VIOLATION_PER_SECOND | MERGE_QUERY_PER_SECOND | ACTIVE_CONNECTIONS | CONNECTION_CREATE_PER_SECOND | RT(MS) | RDS_RT(MS) | NET_IN(KB/S) | NET_OUT(KB/S) | THREAD_RUNNING | HINT_USED_PER_SECOND | HINT_USED_COUNT | AGGREGATE_QUERY_PER_SECOND | AGGREGATE_QUERY_COUNT | TEMP_TABLE_CREATE_PER_SECOND | TEMP_TABLE_CREATE_COUNT | MULTI_DB_JOIN_PER_SECOND | MULTI_DB_JOIN_COUNT | CPU | FREEMEM | FULLGCCOUNT | FULLGCTIME |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1.63 | 1.68 | 0.03 | 0.03 | 0.02 | 0.00 | 0.00 | 6 | 0.01 | 157.13 | 51.14 | 134.33 | 1.21 | 1 | 0.00 | 54 | 0.00 | 663 | 0.00 | 512 | 0.00 | 516 | 0.09% | 6.96% | 76446 | 21326906 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

SHOW DB STATUS

You can run this SQL statement to view the capacity and performance information of a physical database. All the returned values indicate the real-time information. The capacity information is obtained from the ApsaraDB RDS for MySQL system table, and therefore may be different from the actual capacity information.

The following describes the meanings of important columns:

- **NAME:** the PolarDB-X internal tag that represents a logical PolarDB-X database corresponding to the database shard. The value is different from the name of the logical PolarDB-X database.
- **CONNECTION_STRING:** the information about a connection from the PolarDB-X instance to the database shard.
- **PHYSICAL_DB:** the name of the database shard. The **TOTAL** row indicates the total amount of capacity of all the database shards corresponding to the logical PolarDB-X database.

- **SIZE_IN_MB**: the size of the space occupied by the data in the database shard. Unit: MB.
- **RATIO**: the ratio of the data volume of the database shard to the total data volume of the current logical PolarDB-X database.
- **THREAD_RUNNING**: the number of threads that are running in the ApsaraDB RDS for MySQL instance to which the physical database belongs. The meaning of this parameter is the same as that of the **THREAD_RUNNING** parameter returned by the MySQL `SHOW GLOBAL STATUS` command. For more information, see [MySQL official documentation](#).

```
mysql> show db status;
+-----+-----+-----+-----+-----+-----+-----+
| ID | NAME                | CONNECTION_STRING | PHYSICAL_DB | SIZE_IN_MB | RATIO | THREAD_RUNNING |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | drds_db_1516187088365dau | 100.100.64.1:59077 | TOTAL      | 13.109375 | 100% | 3              |
| 2 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0000 | 1.578125 | 12.04% |                |
| 3 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0001 | 1.4375 | 10.97% |                |
| 4 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0002 | 1.4375 | 10.97% |                |
| 5 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0003 | 1.4375 | 10.97% |                |
| 6 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0004 | 1.734375 | 13.23% |                |
| 7 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0005 | 1.734375 | 13.23% |                |
| 8 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0006 | 2.015625 | 15.38% |                |
| 9 | drds_db_1516187088365dau | 100.100.64.1:59077 | drds_db_xzip_0007 | 1.734375 | 13.23% |                |
+-----+-----+-----+-----+-----+-----+-----+
```

SHOW FULL DB STATUS [LIKE {tablename}]

You can run this SQL statement to view the capacity and performance information of a table shard in a physical database, which is also called a database shard. All the returned values indicate the real-time information. The capacity information is obtained from the ApsaraDB RDS for MySQL system table, and therefore may be different from the actual capacity information.

The following describes the meanings of important columns:

- **NAME**: the PolarDB-X internal tag that represents a logical PolarDB-X database corresponding to the database shard. The value is different from the name of the logical PolarDB-X database.
- **CONNECTION_STRING**: the information about a connection from the PolarDB-X instance to the database shard.
- **PHYSICAL_DB**: the name of the database shard. If the **LIKE** keyword is used for filtering in the statement, the **TOTAL** row indicates the total amount of capacity of the database shard. If the **LIKE** keyword is not used for filtering in the statement, the **TOTAL** row indicates the total amount of capacity of all database shards.
- **PHYSICAL_TABLE**: the name of the table shard in the database shard. If the **LIKE** keyword is used for filtering in the statement, the **TOTAL** row indicates the total amount of capacity of the table shard. If the **LIKE** keyword is not used for filtering in the statement, the **TOTAL** row indicates the total amount of capacity of all table shards in the database shard.
- **SIZE_IN_MB**: the size of the space occupied by the data in the database shard. Unit: MB.
- **RATIO**: the ratio of the data volume of the table shard to the total data volume of all the table shards obtained through filtering.
- **THREAD_RUNNING**: the number of threads that are running in the ApsaraDB RDS for MySQL instance to which the physical database belongs. The meaning of this parameter is the same as that of the **THREAD_RUNNING** parameter returned by the MySQL `SHOW GLOBAL STATUS` command. For more information, see [MySQL official documentation](#).

```
mysql> show full db status like hash_tb;
```

ID	NAME	CONNECTION_STRING	PHYSICAL_DB	PHYSICAL_TABLE	SIZE_IN_MB	RATIO	THREAD_RUNNING
1	drds_db_1516187088365dai	100.100.64.1:59077	TOTAL		19.875	100%	3
2	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0000	TOTAL	3.03125	15.25%	
3	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0000	hash_tb_00	1.515625	7.63%	
4	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0000	hash_tb_01	1.515625	7.63%	
5	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0001	TOTAL	2.0	10.06%	
6	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0001	hash_tb_02	1.515625	7.63%	
7	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0001	hash_tb_03	0.484375	2.44%	
8	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0002	TOTAL	3.03125	15.25%	
9	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0002	hash_tb_04	1.515625	7.63%	
10	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0002	hash_tb_05	1.515625	7.63%	
11	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0003	TOTAL	1.953125	9.83%	
12	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0003	hash_tb_06	1.515625	7.63%	
13	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0003	hash_tb_07	0.4375	2.2%	
14	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0004	TOTAL	3.03125	15.25%	
15	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0004	hash_tb_08	1.515625	7.63%	
16	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0004	hash_tb_09	1.515625	7.63%	
17	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0005	TOTAL	1.921875	9.67%	
18	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0005	hash_tb_11	1.515625	7.63%	
19	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0005	hash_tb_10	0.40625	2.04%	
20	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0006	TOTAL	3.03125	15.25%	
21	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0006	hash_tb_12	1.515625	7.63%	
22	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0006	hash_tb_13	1.515625	7.63%	
23	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0007	TOTAL	1.875	9.43%	
24	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0007	hash_tb_14	1.515625	7.63%	
25	drds_db_1516187088365dai	100.100.64.1:59077	drds_db_xzip_0007	hash_tb_15	0.359375	1.81%	

13.7.6. SHOW PROCESSLIST and KILL commands

Note

- If the version of PolarDB-X is 5.1.28-1408022 or later, PolarDB-X support the SHOW PROCESSLIST and KILL commands for both logical and physical connections. For more information, see this topic.
- If the version of PolarDB-X is earlier than 5.1.28-1408022, PolarDB-X support the SHOW PROCESSLIST and KILL commands only for physical connections. For more information, see [SHOW PROCESSLIST and KILL commands in earlier versions](#).

SHOW PROCESSLIST command

In a PolarDB-X instance, you can run the `SHOW PROCESSLIST` command to view information such as connections to the PolarDB-X instance and SQL statements that are being executed in the PolarDB-X instance.

Syntax

```
SHOW [FULL] PROCESSLIST
```

Examples

```
mysql> SHOW PROCESSLIST\G
  ID: 1971050
  USER: admin
  HOST: 111.111.111.111:4303
  DB: drds_test
  COMMAND: Query
  TIME: 0
  STATE:
  INFO: show processlist
1 row in set (0.01 sec)
```

The following describes the meanings of the fields in the result set:

- **ID:** the ID of the connection. The value is a long-type number.
- **USER:** the name of the user who sets up the connection.
- **HOST:** the IP address and port number of the host that sets up the connection.
- **DB:** the name of the database accessed by the connection.
- **COMMAND:** the usage state of the connection. Currently, this field can be set to the following values:
 - **Query:** the current connection is executing an SQL statement.
 - **Sleep:** the current connection is idle.
- **TIME:** the duration when the connection is in the current state:
 - When the value of **COMMAND** is **Query**, this field indicates how long the SQL statement has been being executed on the connection.
 - When the value of **COMMAND** is **Sleep**, this field indicates how long the connection has been in the idle state.
- **STATE:** currently, no meaning has been assigned for this field. The value is constantly empty.
- **INFO:**
 - When the value of **COMMAND** is **Query**, this field indicates the content of the SQL statement that is being executed on the connection. If the **FULL** parameter is not specified, a maximum of the first 30 characters of the SQL statement are returned. If the **FULL** parameter is specified, a maximum of the first 1000 characters of the SQL statement are returned.
 - When the value of **COMMAND** is other values, this field is meaningless and left empty.

SHOW PHYSICAL_PROCESSLIST command

In a PolarDB-X instance, you can run the `SHOW PHYSICAL_PROCESSLIST` command to view information about all the SQL statements that are being executed on underlying ApsaraDB RDS for MySQL instances.

Syntax

```
SHOW [FULL] PHYSICAL_PROCESSLIST
```

When an SQL statement is excessively long, the responses of the `SHOW PHYSICAL_PROCESSLIST` command may be truncated. In this case, you can run the `SHOW FULL PHYSICAL_PROCESSLIST` command to obtain the complete SQL statement.

The meaning of each column in the responses is equivalent to that in the responses of the `SHOW PROCESSLIST` command. For more information, see [SHOW PROCESSLIST Syntax](#).

 **Note** Different from ApsaraDB RDS for MySQL, the PolarDB-X instance returns a string instead of a number in the ID column of a physical connection.

```
mysql> SHOW PHYSICAL_PROCESSLIST\G
***** 1. row *****
      ID: 0-0-521414
      USER: tddl5
      DB: tddl5_00
      COMMAND: Query
      TIME: 0
      STATE: init
      INFO: show processlist
***** 2. row *****
      ID: 0-0-521570
      USER: tddl5
      DB: tddl5_00
      COMMAND: Query
      TIME: 0
      STATE: User sleep
      INFO: /*DRDS /88.88.88.88/b67a0e4d880000/ */ select sleep(1000)
2 rows in set (0.01 sec)
```

KILL command

The `KILL` command is used to terminate an SQL statement that is being executed.

The PolarDB-X instance connects to an ApsaraDB RDS for MySQL instance by using the username created by the PolarDB-X instance on the ApsaraDB RDS for MySQL instance. Therefore, if you directly connect to the ApsaraDB RDS for MySQL instance, you do not have the permission to run the `KILL` command on a request initiated by the PolarDB-X instance.

To terminate an SQL statement that is being executed on the PolarDB-X instance, you must use tools such as the MySQL command line and to connect to the PolarDB-X instance, and then run the `KILL` command on the PolarDB-X instance.

Syntax

```
KILL PROCESS_ID | 'PHYSICAL_PROCESS_ID' | 'ALL'
```

The KILL command can be used in the following ways:

- Run `KILL PROCESS_ID` to terminate a specified logical SQL statement.

The `PROCESS_ID` parameter is obtained from the `ID` column in the responses of the `SHOW [FULL] PROCESSLIST` command.

Running the `KILL PROCESS_ID` command in the PolarDB-X instance will terminate both logical and physical SQL statements that are being executed on this connection, and disconnect this connection.

The PolarDB-X instance does not support the `KILL QUERY` command.

- Run `KILL 'PHYSICAL_PROCESS_ID'` to terminate a specified physical SQL statement.

The `PHYSICAL_PROCESS_ID` parameter is obtained from the `ID` column in the responses of the `SHOW PHYSICAL_PROCESS_ID` command.

 **Note** The `PHYSICAL_PROCESS_ID` column is a string instead of a number. Therefore, the `PHYSICAL_PROCESS_ID` parameter must be enclosed in single quotation marks (' ') in the KILL command.

Examples:

```
mysql> KILL '0-0-521570';
Query OK, 0 rows affected (0.01 sec)
```

- Run `KILL 'ALL'` to terminate all the physical SQL statements that are executed by the PolarDB-X instance in the current logical database.

When the underlying ApsaraDB RDS for MySQL instance is overloaded due to some SQL statements, you can use the `KILL 'ALL'` command to terminate all the physical SQL statements that are being executed in the current logical PolarDB-X database.

All physical SQL statements indicated by `PROCESS` that meet the following conditions can be terminated by running `KILL 'ALL'` :

- The value of the `User` parameter for the physical SQL statement indicated by `PROCESS` is a username created by the PolarDB-X instance in the ApsaraDB RDS for MySQL instance.
- The physical SQL statement indicated by `PROCESS` is executing a query. In other words, the value of `COMMAND` is `Query`.

13.7.7. SHOW PROCESSLIST and KILL commands in earlier versions

 **Note**

- If the version of PolarDB-X is 5.1.28-1408022 or later, PolarDB-X support the `SHOW PROCESSLIST` and `KILL` commands for both logical and physical connections. For more information, see [SHOW PROCESSLIST and KILL commands](#).
- If the version of PolarDB-X is earlier than 5.1.28-1408022, PolarDB-X only supports the `SHOW PROCESSLIST` and `KILL` commands for physical connections. For more information, see this topic.

SHOW PROCESSLIST command

In a PolarDB-X instance, you can run the `SHOW PROCESSLIST` command to view information about all the SQL statements that are being executed on the ApsaraDB RDS for MySQL instances.

Syntax

```
SHOW [FULL] PROCESSLIST
```

When an SQL statement is excessively long, the responses of the `SHOW PROCESSLIST` command may be truncated. In this case, you can run the `SHOW FULL PROCESSLIST` command to obtain the complete SQL statement.

The meaning of each column in the responses is equivalent to that in the responses of the `SHOW PROCESSLIST` command. For more information, see [SHOW PROCESSLIST Syntax](#).

```
mysql> SHOW PROCESSLIST\G
***** 1. row *****
      ID: 0-0-521414
      USER: tddl5
      DB: tddl5_00
      COMMAND: Query
      TIME: 0
      STATE: init
      INFO: show processlist
      ROWS_SENT: NULL
      ROWS_EXAMINED: NULL
      ROWS_READ: NULL
***** 2. row *****
      ID: 0-0-521570
      USER: tddl5
      DB: tddl5_00
      COMMAND: Query
      TIME: 0
      STATE: User sleep
      INFO: /*DRDS /88.88.88.88/b67a0e4d8800000/ */ select sleep(1000)
      ROWS_SENT: NULL
      ROWS_EXAMINED: NULL
      ROWS_READ: NULL
2 rows in set (0.01 sec)
```

KILL command

The KILL command is used to terminate an SQL statement that is being executed.

The PolarDB-X instance connects to an ApsaraDB RDS for MySQL instance by using the username created by the PolarDB-X instance on the ApsaraDB RDS for MySQL instance. Therefore, if you directly connect to the ApsaraDB RDS for MySQL instance, you do not have the permission to run the KILL command on a request initiated by the PolarDB-X instance.

To terminate an SQL statement that is being executed on the PolarDB-X instance, you must use tools such as the MySQL command line and to connect to the PolarDB-X instance, and then run the KILL command on the PolarDB-X instance.

Syntax

```
KILL 'PROCESS_ID' | 'ALL'
```

The KILL command can be used in the following ways:

- Run `KILL 'PROCESS_ID'` to terminate a specified SQL statement.

The `PROCESS_ID` parameter is obtained from the `ID` column in the responses of the `SHOW PROCESSLIST` command.

 **Note** Different from ApsaraDB RDS for MySQL, the PolarDB-X instance returns a string instead of a number in the `ID` column. Therefore, the `PROCESS_ID` parameter must be enclosed in single quotation marks (`'`) in the `KILL` command.

Examples

```
mysql> KILL '0-0-521570';
Query OK, 0 rows affected (0.01 sec)
```

- Run `KILL 'ALL'` to terminate all the SQL statements executed by the PolarDB-X instance in the current logical database.

When the underlying ApsaraDB RDS for MySQL instance is overloaded due to several SQL statements, you can use the `KILL 'ALL'` command to terminate all the SQL statements that are being executed in the current logical PolarDB-X database.

All SQL statements indicated by `PROCESS` that meet the following conditions can be terminated by running `KILL 'ALL'` :

- The value of the `User` parameter for the physical SQL statement indicated by `PROCESS` is a username created by the PolarDB-X instance in the ApsaraDB RDS for MySQL instance.
- The physical SQL statement indicated by `PROCESS` is executing a query. In other words, the value of `COMMAND` is `Query`.

PolarDB-X instances in earlier versions do not support the `KILL 'ALL'` command. An error will be reported if this command is being executed in these instances. To resolve this problem, you can upgrade the version of the PolarDB-X instance.

13.8. Custom hints

 **Note** This topic is applicable to PolarDB-X 5.3 and later. For earlier versions, see [PolarDB-X 5.2 hints](#).

13.8.1. Introduction to hints

As a supplement to the SQL syntax, hints play a critical role in relational databases. They allow you to influence execution plans of SQL statements by using relevant syntax, to specially optimize the SQL statements. PolarDB-X also provides special hint syntax.

For example, if you know the target data is stored in table shards in certain database shards and you need to route the SQL statement directly to the database shards for execution, you can use custom hints provided by PolarDB-X.

```
SELECT /*+TDDL:node('node_name')*/ * FROM table_name;
```

In the preceding SQL statement, the part between `/*` and `*/`, namely, `+TDDL:node('node_name')`, is a PolarDB-X hint. The hint specifies the ApsaraDB RDS for MySQL database shard where the SQL statement is to be executed.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

PolarDB-X hint syntax

Basic syntax

```
/*+TDDL: hint_command [hint_command ...] */
/*!+TDDL: hint_command [hint_command ...] */
```

PolarDB-X hints are based on the [MySQL Comment Syntax](#). The hint statements are located between `/*` and `*/` or between `/*!` and `*/`, and must begin with `+TDDL:`. The `hint_command` parameter indicates a PolarDB-X hint command related to the specific operation. Multiple `hint_command` parameters are separated by spaces.

Examples

```
# Query the names of physical tables in each database shard.
/*+TDDL:scan()*/SHOW TABLES;
# Route the query to database shard 0000 of a read-only ApsaraDB RDS for MySQL instance.
/*+TDDL:node(0) slave()*/SELECT * FROM t1;
```

In the example, `/*+TDDL:scan()*/` and `/*+TDDL:node(0) slave()*/` are PolarDB-X hints that begin with `+TDDL:`. The `scan()`, `node(0)`, and `slave()` functions are PolarDB-X hint commands. Hint commands are separated by spaces.

- Use one hint in an SQL statement:

PolarDB-X allows you to use hints in data manipulation language (DML), data definition language (DDL), and data access language (DAL) statements. The following describes the syntax in detail.

- For all statements that support hints, you can specify a hint at the beginning of the statements, for example:

```
/*+TDDL: ... */ SELECT ...
/*+TDDL: ... */ INSERT ...
/*+TDDL: ... */ REPLACE ...
/*+TDDL: ... */ UPDATE ...
/*+TDDL: ... */ DELETE ...
/*+TDDL: ... */ CREATE TABLE ...
/*+TDDL: ... */ ALTER TABLE ...
/*+TDDL: ... */ DROP TABLE ...
/*+TDDL: ... */ SHOW ...
...
```

- For DML statements, you can specify a hint behind the first keyword of the statements, for example:

```
SELECT /*+TDDL: ... */ ...
INSERT /*+TDDL: ... */ ...
REPLACE /*+TDDL: ... */ ...
UPDATE /*+TDDL: ... */ ...
DELETE /*+TDDL: ... */ ...
...
```

 **Note** Different hints may be applicable to different syntaxes. For more information about the applicable syntaxes, see the documentation of hint commands.

- Use multiple hint commands in an SQL statement:

PolarDB-X allows you to use multiple hint commands in SQL statements that contain hints.

```
SELECT /*+TDDL:node(0) slave()*/ ... ;
```

PolarDB-X has the following limitations on the use of multiple hint commands:

```
# A single SQL statement cannot contain multiple hint statements.
SELECT /*+TDDL:node(0)*/ /*+TDDL:slave()*/ ... ;
# An SQL statement that contains a hint cannot contain duplicate hint commands.
SELECT /*+TDDL:node(0) node(1)*/ ... ;
```

PolarDB-X hint classification

PolarDB-X hints are classified into the following major categories according to operation types:

- [Read/write splitting](#)
- [Specify a timeout period for an SQL statement](#)
- [Specify a database shard to run an SQL statement](#)
- [Scan all or some of database shards and table shards](#)

13.8.2. Read/write splitting

PolarDB-X provides transparent read/write splitting at the application layer. Data synchronization between primary and read-only ApsaraDB RDS for MySQL instances has a delay of several milliseconds. If you need to read changed data immediately after the primary ApsaraDB RDS for MySQL instance is changed, you must ensure that the SQL statement for reading data is routed to the primary ApsaraDB RDS for MySQL instance. To meet this demand, PolarDB-X provides custom hints for read/write splitting, to route SQL statements to a specified primary or read-only ApsaraDB RDS for MySQL instance.

 **Note** This topic is applicable to PolarDB-X 5.3 and later. For earlier versions, see [Read/write splitting](#).

Syntax

```
/*+TDDL:
  master()
  | slave()
*/
```

With this custom hint, you can specify whether to run an SQL statement on a primary or read-only ApsaraDB RDS for MySQL instance. With the custom hint `/*+TDDL:slave()*/`, if a primary ApsaraDB RDS for MySQL instance is configured with multiple read-only ApsaraDB RDS for MySQL instances, the PolarDB-X instance randomly selects a read-only ApsaraDB RDS for MySQL instance based on its weight, to run the SQL statement.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

- Specify a primary ApsaraDB RDS for MySQL instance to run an SQL statement:

```
SELECT /*+TDDL:master()*/ * FROM table_name;
```

After the custom hint `/*+TDDL:master()*/` is added behind the first keyword in the SQL statement, this SQL statement is routed to the primary ApsaraDB RDS for MySQL instance for execution.

- Specify a read-only ApsaraDB RDS for MySQL instance to run an SQL statement:

```
SELECT /*+TDDL:slave()*/ * FROM table_name;
```

After the custom hint `/*+TDDL:slave()*/` is added behind the first keyword in the SQL statement, this SQL statement is randomly routed to a read-only ApsaraDB RDS for MySQL instance based on the allocated weight.

Note

- The custom hints for read-write splitting are only applicable to read SQL statements for non-transactional data. SQL statements for transactional data and write SQL statements are still routed to the primary ApsaraDB RDS for MySQL instance for execution.
- The PolarDB-X hint `/*+TDDL:slave()*/` allows you to route the SQL statement randomly to a read-only ApsaraDB RDS for MySQL instance based on the configured weight for execution. If no read-only ApsaraDB RDS for MySQL instance is available, no error is reported. Instead, the primary ApsaraDB RDS for MySQL instance is selected to run the SQL statement.

13.8.3. Specify a timeout period for an SQL statement

In PolarDB-X, the SQL statements for PolarDB-X instances and ApsaraDB RDS for MySQL instances are timed out after 900 seconds (which can be adjusted) by default. However, for some slow SQL statements, the execution duration may exceed 900 seconds. For these slow SQL statements, PolarDB-X provides a custom hint to adjust their timeout periods. You can use this custom hint to adjust the SQL execution duration as needed.

 Note This topic is applicable to PolarDB-X 5.3 and later. For earlier versions, see [Specify a timeout period for an SQL statement](#).

Syntax

The syntax of the PolarDB-X hint for specifying a timeout period for an SQL statement is as follows:

```
/*+TDDL:SOCKET_TIMEOUT(time)*/
```

The `SOCKET_TIMEOUT` parameter is measured in milliseconds. With this custom hint, you can adjust the timeout period for the SQL statement based on business requirements.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/` .
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/` , add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

Set the timeout period of an SQL statement to 40 seconds:

```
/*+TDDL:SOCKET_TIMEOUT(40000)*/SELECT * FROM t_item;
```

Note A longer timeout period causes database resources to be occupied for a longer period of time. If excessive SQL statements are executed over a long time within the same period, a large number of database resources may be consumed. This will make users unable to use PolarDB-X properly. In this case, we need to use this custom hint to optimize the SQL statements that take a long time to execute.

13.8.4. Specify a database shard to run an SQL statement

When running SQL commands in a PolarDB-X instance, you may find that some SQL statements are not supported by the PolarDB-X instance. In this case, you can use the `NODE HINT` provided by PolarDB-X, to route the SQL statements to one or more database shards for execution. In addition, if you need to query the data in a specified database shard or the data in a specified table shard in a known database shard, you can use the `NODE HINT` to directly route the SQL statement to the database shard for execution.

Note This topic is applicable to PolarDB-X 5.3 and later. For earlier versions, see [Specify a database shard to run an SQL statement](#).

Syntax

The `NODE HINT` allows you to specify a database shard by using a shard name, to run the SQL statement in the database shard. A shard name uniquely identifies a database shard in a PolarDB-X instance. You can run the `SHOW NODE` statement to obtain the shard name.

Specify a database shard by using a shard name, to run an SQL statement

This custom hint allows you to specify one or more database shards to run an SQL statement.

Note If the hint for specifying a database shard is used in an `INSERT` statement that contains a sequence for the target table, the sequence will not take effect. For more information, see [Limits and precautions for sequences](#).

- Specify one database shard to run an SQL statement:

```
/*+TDDL:node('node_name')*/
```

Specifically, `node_name` indicates the shard name. This PolarDB-X hint enables you to route the SQL statement to the database shard specified by `node_name` .

- Specify multiple database shards to run an SQL statement:

```
/*+TDDL:node('node_name','node_name1','node_name2')*/
```

You can specify multiple shard names in the parameters and route the SQL statement to multiple database shards for execution. Separate multiple shard names with commas (,).

Note

- When this custom hint is used, the PolarDB-X instance directly routes the SQL statement to the specified database shards for execution. Therefore, the specified shard names in the SQL statement must correspond to existing database shards.
- The `NODE HINT` can be used in data manipulation language (DML), data definition language (DDL), and data access language (DAL) statements.
- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

The following shows the responses of the `SHOW NODE` statement for a logical database named `drds_test` in a PolarDB-X instance.

```
mysql> SHOW NODE\G
***** 1. row *****
      ID: 0
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS
      MASTER_READ_COUNT: 212
      SLAVE_READ_COUNT: 0
      MASTER_READ_PERCENT: 100%
      SLAVE_READ_PERCENT: 0%
***** 2. row *****
      ID: 1
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0001_RDS
      MASTER_READ_COUNT: 29
      SLAVE_READ_COUNT: 0
      MASTER_READ_PERCENT: 100%
      SLAVE_READ_PERCENT: 0%
***** 3. row *****
      ID: 2
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0002_RDS
      MASTER_READ_COUNT: 29
      SLAVE_READ_COUNT: 0
      MASTER_READ_PERCENT: 100%
      SLAVE_READ_PERCENT: 0%
***** 4. row *****
      ID: 3
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS
```

```

MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 5. row *****
      ID: 4
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0004_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 6. row *****
      ID: 5
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0005_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 7. row *****
      ID: 6
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 8. row *****
      ID: 7
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0007_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
8 rows in set (0.02 sec)

```

As you can see, each database shard has the `NAME` attribute, which indicates the shard name corresponding to the database shard. Each shard name uniquely corresponds to one database shard name. For example, the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS` corresponds to the database shard name `drds_test_vtla_0003`. Therefore, after obtaining the shard name, you can use the PolarDB-X hint to specify the corresponding database shard to run the SQL statement.

- Specify database shard 0 to run an SQL statement:

```
SELECT /*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS')*/ * FROM table_name;
```

- Specify multiple database shards to run an SQL statement:

```
SELECT /*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS','DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS')*/ * FROM table_name;
```

This SQL statement will be executed in the database shards whose shard names are `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` and `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS` .

- View the execution plan of an SQL statement in database shard 0:

```
/*TDDL:node('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS')*/ EXPLAIN SELECT * FROM table_name;
```

13.8.5. Scan all or some of database shards and table shards

In addition to routing an SQL statement to one or more database shards for execution, PolarDB-X provides the `SCAN HINT` to scan all or some of database shards and table shards. With the `SCAN HINT` , you can route an SQL statement to each database shard at a time. For example, you can view all the table shards in a specified database shard or view the data volume of each physical table of a specified logical table.

 **Note** This topic is applicable to PolarDB-X 5.3 and later. For earlier versions, see [Scan all database shards and table shards](#).

With the `SCAN HINT`, you can specify the following SQL execution manners:

- Run an SQL statement in all table shards in all database shards.
- Run an SQL statement in all table shards in a specified database shard.
- Run an SQL statement in the specified table shard in the specified database shard by calculating the name of the physical table based on conditions.
- Run an SQL statement in the specified table shard in the specified database shard by explicitly specifying the name of the physical table.

The `SCAN HINT` can be used in data manipulation language (DML) statements, data definition language (DDL) statements, and some data access language (DAL) statements.

 **Note**

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/` .
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/` , add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Syntax

```

# SCAN HINT
# Route an SQL statement to all table shards in all database shards.
SCAN()
# Route an SQL statement to all table shards in a specified database shard.
SCAN(NODE="node_list")      # Specify the database shard.
# Route an SQL statement to the specified table shard in the specified database shard by calculating the name of the
physical table based on conditions.
SCAN(
  [TABLE="table_name_list"    # Specify the name of the logical table.
  , CONDITION="condition_string" # Calculate the names of physical databases based on the content of TABLE and C
ONDITION.
  [, NODE="node_list"])      # Filter the results obtained based on the content of CONDITION, to retain only the resul
ts of the specified physical database.
# Route an SQL statement to the specified table shard in the specified database shard by explicitly specifying the nam
e of the physical table.
SCAN(
  [TABLE="table_name_list"    # Specify the name of the logical table.
  , REAL_TABLE=("table_name_list") # Specify the name of the physical table. The same physical table names are appli
ed to all physical databases.
  [, NODE="node_list"])      # Filter the results obtained based on the content of CONDITION, to retain only the resul
ts of the specified physical database.
# Specify physical table names or logical table names.
table_name_list:
  table_name [, table_name]...
# Specify physical databases by using GROUP_KEY and GROUP_INDEX, which can be obtained by running the SHOW NO
DE statement.
node_list:
  {group_key | group_index} [, {group_key | group_index}]...
# Run an SQL WHERE statement. When using this syntax, you must specify conditions for each table, for example, t1.id
= 2 and t2.id = 2.
condition_string:
  where_condition

```

Examples

- Run the following SQL statement in all table shards in all database shards:

```
SELECT /*+TDDL:scan()*/ COUNT(1) FROM t1
```

After this statement is executed, the SQL statement is routed to all the physical tables corresponding to the logical table `t1`, and the result sets are merged and returned.

- Run the following SQL statement in all table shards in specified database shards:

```
SELECT /*+TDDL:scan(node='0,1,2')*/ COUNT(1) FROM t1
```

After this statement is executed, all physical tables corresponding to the logical table `t1` in database shards 0000, 0001, and 0002 are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

- Run the following SQL statement in specified table shards based on conditions:

```
SELECT /*+TDDL:scan('t1', condition='t1.id = 2')*/ COUNT(1) FROM t1
```

After this statement is executed, all physical tables that correspond to the logical table `t1` and meet the conditions are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

- Run the following SQL JOIN statement in the specified table shards based on conditions:

```
SELECT /*+TDDL:scan('t1, t2', condition='t1.id = 2 and t2.id = 2')*/ * FROM t1 a JOIN t2 b ON a.id = b.id WHERE b.name = "test"
```

After this statement is executed, all physical tables that correspond to the logical tables `t1` and `t2` and meet the conditions are calculated, the SQL statement is routed to the physical tables, and the result sets are merged and returned.

 **Notice** Before using this custom hint, you must ensure that the logical tables `t1` and `t2` are partitioned into the same number of database shards and the same number of table shards. Otherwise, the database shards calculated by the PolarDB-X instance based on the conditions are different, and an error will be returned.

- Run the following SQL statement in the specified table shards in database shards by explicitly specifying the names of the physical tables:

```
SELECT /*+TDDL:scan('t1', real_table=('t1_00', 't1_01'))*/ COUNT(1) FROM t1
```

After this statement is executed, the SQL statement is routed to the table shards `t1_00`t1_01` in all database shards, and the result sets are merged and returned.

- Run the following SQL JOIN statement in the specified table shards in database shards by explicitly specifying the names of the physical tables:

```
SELECT /*+TDDL:scan('t1, t2', real_table=('t1_00,t2_00', 't1_01,t2_01'))*/ * FROM t1 a JOIN t2 b ON a.id = b.id WHERE b.name = "test";
```

After this statement is executed, the SQL statement is routed to the table shards `t1_00` , `t2_00` , `t1_01` , and `t2_01` in all database shards, and the result sets are merged and returned.

13.8.6. INDEX HINT

- PolarDB-X supports global secondary indexes. The `INDEX` hint allows you to obtain query results from a specified GSI.
- The `INDEX` hint takes effect only for SQL `SELECT` statements.

 **Note** This custom hint is applicable to only MySQL 5.7 and later and PolarDB-X 5.4.1 and later.

Syntax

```
# FORCE INDEX
tbl_name [[AS] alias] [index_hint]
index_hint:
    FORCE INDEX({index_name})
# INDEX()
/*+TDDL:
    INDEX({table_name | table_alias}, {index_name})
*/
```

PolarDB-X INDEX hint can be used in two ways:

- `FORCE INDEX()` : This syntax is the same as that of [MySQL FORCE INDEX](#).
- `INDEX()` : In this syntax, a global secondary index is specified using a table name (or alias) and an index name. This hint does not take effect in the following cases:
 - The query does not contain the specified table name or alias.
 - The specified global secondary index is not in the specified table.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/` .
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/` , add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

```
CREATE TABLE t_order (
  `id` bigint(11) NOT NULL AUTO_INCREMENT,
  `order_id` varchar(20) DEFAULT NULL,
  `buyer_id` varchar(20) DEFAULT NULL,
  `seller_id` varchar(20) DEFAULT NULL,
  `order_snapshot` longtext DEFAULT NULL,
  `order_detail` longtext DEFAULT NULL,
  PRIMARY KEY (`id`),
  GLOBAL INDEX `g_i_seller` (`seller_id`) dbpartition by hash(`seller_id`),
  UNIQUE GLOBAL INDEX `g_i_buyer` (`buyer_id`) COVERING(`seller_id`, `order_snapshot`)
  dbpartition by hash(`buyer_id`) tpartition by hash(`buyer_id`) tpartitions 3
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`order_id`);
```

Specify the global secondary index `g_i_seller` by using `FORCE INDEX` in the `FROM` clause:

```
SELECT a.*, b.order_id
FROM t_seller a
JOIN t_order b FORCE INDEX(g_i_seller) ON a.seller_id = b.seller_id
WHERE a.seller_nick="abc";
```

Specify the global secondary index `g_i_buyer` by using `INDEX+table alias`:

```
/*+TDDL:index(a, g_i_buyer)*/ SELECT * FROM t_order a WHERE a.buyer_id = 123
```

13.9. PolarDB-X 5.2 hints

13.9.1. Introduction to hints

As a supplement to the SQL syntax, hints play a critical role in relational databases. They allow you to affect execution plans of SQL statements by using relevant syntax, to specially optimize the SQL statements.

Overview of PolarDB-X hints

PolarDB-X provides special hint syntax.

For example, if you know the target data is stored in table shards in certain database shards and you need to route the SQL statement directly to the database shards for execution, you can use custom hints provided by PolarDB-X.

```
/*! TDDL:NODE IN('node_name', ...) */SELECT * FROM table_name;
```

In the preceding SQL statement, the part between `/*!` and `*/`, namely, `TDDL:node in('node_name', ...)`, is a PolarDB-X hint. The hint specifies the ApsaraDB RDS for MySQL database shard where the SQL statement is to be executed.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*! +TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

PolarDB-X hint syntax

Basic syntax:

```
/*! TDDL:hint command*/
```

PolarDB-X hints are based on [MySQL Comment Syntax](#). Therefore, an SQL statement that contains a PolarDB-X hint is located between `/*!` and `*/`, and must begin with `TDDL: .` The `hint command` indicates a PolarDB-X hint command related to the specific operation. For example, a PolarDB-X hint is added to the following SQL statement to display the name of each database shard.

```
/*! TDDL:SCAN*/SHOW TABLES;
```

In this SQL statement, `/*! TDDL:SCAN*/` is the PolarDB-X hint that begins with `TDDL: .`, and `SCAN` is a PolarDB-X hint command.

13.9.2. Read/write splitting

PolarDB-X provides transparent read/write splitting at the application layer. Data synchronization between primary and read-only ApsaraDB RDS for MySQL instances has a delay of several milliseconds. If you need to read changed data immediately after the primary ApsaraDB RDS for MySQL instance is changed, you must ensure that the SQL statement for reading data is routed to the primary ApsaraDB RDS for MySQL instance. To meet this demand, PolarDB-X provides custom hints for read/write splitting, to route SQL statements to a specified primary or read-only ApsaraDB RDS for MySQL instance.

Syntax

```
/*! TDDL:MASTER|SLAVE*/
```

With this custom hint, you can specify whether to run an SQL statement on a primary or read-only ApsaraDB RDS for MySQL instance. With the custom hint `/*!TDDL:SLAVE*/`, if a primary ApsaraDB RDS for MySQL instance is configured with multiple read-only ApsaraDB RDS for MySQL instances, the PolarDB-X instance randomly selects a read-only ApsaraDB RDS for MySQL instance based on its weight, to run the SQL statement.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

- Specify a primary ApsaraDB RDS for MySQL instance to run an SQL statement:

```
/*! TDDL:MASTER*/SELECT * FROM table_name;
```

After the custom hint `/*! TDDL:MASTER*/` is added at the beginning of the SQL statement, this SQL statement is routed to the primary ApsaraDB RDS for MySQL instance for execution.

- Specify a read-only ApsaraDB RDS for MySQL instance to run an SQL statement:

```
/*! TDDL:SLAVE*/SELECT * FROM table_name;
```

After the custom hint `/*! TDDL:SLAVE*/` is added at the beginning of the SQL statement, this SQL statement is randomly routed to a read-only ApsaraDB RDS for MySQL instance based on the allocated weight.

Note

- The custom hints for read-write splitting are only applicable to read SQL statements for non-transactional data. SQL statements for transactional data and write SQL statements are still routed to the primary ApsaraDB RDS for MySQL instance for execution.
- The PolarDB-X hint `/*+TDDL:slave()*/` allows you to route the SQL statement randomly to a read-only ApsaraDB RDS for MySQL instance based on the configured weight for execution. If no read-only ApsaraDB RDS for MySQL instance is available, no error is reported. Instead, the primary ApsaraDB RDS for MySQL instance is selected to run the SQL statement.

13.9.3. Prevent the delay from a read-only ApsaraDB RDS for MySQL instance

Normally, if you have configured a read-only ApsaraDB for RDS instance for the primary ApsaraDB RDS for MySQL instance of a logical database in a PolarDB-X instance and set read traffic for both the primary and read-only ApsaraDB RDS for MySQL instances, PolarDB-X routes SQL statements to the primary and read-only ApsaraDB RDS for MySQL instances based on the read/write ratio. However, if asynchronous data replication between the primary and read-only ApsaraDB RDS for MySQL instances has a high delay, an error is reported or error results are returned when PolarDB-X routes the SQL statements to the read-only ApsaraDB RDS for MySQL instance.

To address this issue, the PolarDB-X instance provides a custom hint to cut off the delay of the read-only instance. Specifically, based on the maximum delay of primary/secondary replication, PolarDB-X determines whether to route the SQL statement to the primary or the read-only ApsaraDB RDS for MySQL instance.

Syntax

```
/*! TDDL:SQL_DELAY_CUTOFF=time*/
```

With this custom hint, you can specify the value of `SQL_DELAY_CUTOFF`. When the value of `SQL_DELAY` (primary/secondary replication delay of ApsaraDB RDS for MySQL) for the read-only ApsaraDB RDS for MySQL instance reaches or exceeds the value of `time` (which is measured in seconds), the SQL statement is routed to the primary ApsaraDB RDS for MySQL instance.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

- Set the primary/secondary replication delay to 5 seconds:

```
/*! TDDL:SQL_DELAY_CUTOFF=5*/SELECT * FROM table_name;
```

In this SQL statement, the value of `SQL_DELAY_CUTOFF` is set to 5. Therefore, when the value of `SQL_DELAY` for the read-only ApsaraDB RDS for MySQL instance reaches or exceeds 5 seconds, the SQL statement is routed to the primary ApsaraDB RDS for MySQL instance.

- Use the custom hint for delay cutoff with other custom hints:

```
/*! TDDL:SLAVE AND SQL_DELAY_CUTOFF=5*/SELECT * FROM table_name;
```

The custom hint for cutting off the delay of the read-only ApsaraDB RDS for MySQL instance can be used with other hints. By default, the SQL query request is routed to a read-only ApsaraDB RDS for MySQL instance. However, when the primary/secondary replication delay reaches or exceeds 5 seconds, the SQL query request is routed to the primary ApsaraDB RDS for MySQL instance.

13.9.4. Specify a timeout period for an SQL statement

In PolarDB-X, the SQL statements for PolarDB-X instances and ApsaraDB RDS for MySQL instances are timed out after 900 seconds (which can be adjusted) by default. However, for some slow SQL statements, the execution duration may exceed 900 seconds. For these slow SQL statements, PolarDB-X provides a custom hint to adjust their timeout periods. You can use this custom hint to adjust the SQL execution duration as needed.

Syntax

The syntax of the PolarDB-X hint for specifying a timeout period for an SQL statement is as follows:

```
/*! TDDL:SOCKET_TIMEOUT=time*/
```

The `SOCKET_TIMEOUT` parameter is measured in milliseconds. With this custom hint, you can adjust the timeout period for the SQL statement based on business requirements.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*! +TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

Set the timeout period of an SQL statement to 40 seconds:

```
/*! TDDL:SOCKET_TIMEOUT=40000*/SELECT * FROM t_item;
```



Note A longer timeout period causes database resources to be occupied for a longer period of time. If excessive SQL statements are executed over a long time within the same period, a large number of database resources may be consumed. This will make users unable to use PolarDB-X properly. In this case, we need to use this custom hint to optimize the SQL statements that take a long time to execute.

13.9.5. Specify a database shard to run an SQL statement

When running SQL commands in a PolarDB-X instance, you may find that some SQL statements are not supported by the PolarDB-X instance. In this case, you can use the custom hint provided by PolarDB-X to route the SQL statements to one or more database shards for execution. In addition, if you need to query the data in a specified database shard or the data in a specified table shard, you can use the custom hint to directly route the SQL statement to the database shard for execution.

Syntax

This custom hint allows you to specify a database shard by using a shard name or the value of the database shard key, to run an SQL statement in the database shard. A shard name uniquely identifies a database shard in a PolarDB-X instance. You can run the `SHOW NODE` command to obtain the shard name.



Note If the hint for specifying a database shard is used in an INSERT statement that contains a sequence for the target table, the sequence will not take effect. For more information, see [Limits and precautions for sequences](#).

- Specify a database shard by using a shard name, to run an SQL statement

This custom hint allows you to specify one or more database shards to run an SQL statement.

- Specify one database shard to run an SQL statement:

```
/*! TDDL:NODE='node_name'*/
```

Specifically, `node_name` indicates the shard name. This PolarDB-X hint enables you to route the SQL statement to the database shard specified by `node_name`.

- Specify multiple database shards to run an SQL statement:

```
/*! TDDL:NODE IN ('node_name','node_name1','node_name2')*/
```

The `IN` keyword is used to specify multiple shard names. This custom hint allows you to route the SQL statement to multiple database shards. Separate multiple shard names with commas (,).

Note When this custom hint is used, the PolarDB-X instance directly routes the SQL statement to the specified database shards for execution. Therefore, the specified shard names in the SQL statement must correspond to existing database shards.

- Specify a database shard by using the value of the database shard key, to run an SQL statement

```
/*! TDDL:table_name.partition_key=value [and table_name1.partition_key=value1]*/
```

In this PolarDB-X hint, `table_name` indicates the name of a logical table, and this table is a partitioned table. In addition, `partition_key` indicates a shard key, and `value` indicates the value specified for the shard key. In this custom hint, you can use the `and` keyword to specify the shard keys of multiple partitioned tables. When this PolarDB-X hint is used, the PolarDB-X instance calculates the database shards and table shards where the SQL statement is to be executed, and routes the SQL statement to the corresponding database shards.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*!+TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

The following shows the responses of the `SHOW NODE` statement for a logical database named `drds_test` in a PolarDB-X instance.

```
mysql> SHOW NODE\G
***** 1. row *****
      ID: 0
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS
      MASTER_READ_COUNT: 212
      SLAVE_READ_COUNT: 0
      MASTER_READ_PERCENT: 100%
      SLAVE_READ_PERCENT: 0%
***** 2. row *****
      ID: 1
      NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0001_RDS
      MASTER_READ_COUNT: 29
      SLAVE_READ_COUNT: 0
      MASTER_READ_PERCENT: 100%
      SLAVE_READ_PERCENT: 0%
***** 3. row *****
      ID: 2
```

```
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0002_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 4. row *****
ID: 3
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 5. row *****
ID: 4
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0004_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 6. row *****
ID: 5
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0005_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 7. row *****
ID: 6
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
***** 8. row *****
ID: 7
NAME: DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0007_RDS
MASTER_READ_COUNT: 29
SLAVE_READ_COUNT: 0
MASTER_READ_PERCENT: 100%
SLAVE_READ_PERCENT: 0%
8 rows in set (0.02 sec)
```

As you can see, each database shard has the `NAME` attribute, which indicates the shard name corresponding to the database shard. Each shard name uniquely corresponds to one database shard name. For example, the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0003_RDS` corresponds to the database shard name `drds_test_vtla_0003`. Therefore, after obtaining the shard name, you can use the PolarDB-X hint to specify the corresponding database shard to run the SQL statement.

- Specify database shard 0 to run an SQL statement:

```
/*! TDDL:NODE='DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS'*/SELECT * FROM table_name;
```

- Specify multiple database shards to run an SQL statement:

```
/*! TDDL:NODE IN('DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS','DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS')*/SELECT * FROM table_name;
```

This SQL statement will be executed in the database shards whose shard names are `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` and `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0006_RDS`.

- View the execution plan of an SQL statement in a specified database shard:

```
/*! TDDL:NODE='DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS'*/EXPLAIN SELECT * FROM table_name;
```

After this SQL statement is executed, the execution plan of the `SELECT` statement in the database shard corresponding to the shard name `DRDS_TEST_1473471355140LRPRDRDS_TEST_VTLA_0000_RDS` will be returned.

- Specify a database shard by using the value of the database shard key, to run an SQL statement:

PolarDB-X does not support subqueries in the `SET` clause of an `UPDATE` statement, because a shard key must be specified for `UPDATE` statements in PolarDB-X. To address this issue, PolarDB-X provides a custom hint to route the statement to a database shard for execution.

For example, the following shows the `CREATE TABLE` statement for creating two logical tables `t1` and `t2`, which are partitioned into table shards in database shards:

```
CREATE TABLE `t1` (
  `id` bigint(20) NOT NULL,
  `name` varchar(20) NOT NULL,
  `val` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tpartition by hash(`name`) tpartitions 3
CREATE TABLE `t2` (
  `id` bigint(20) NOT NULL,
  `name` varchar(20) NOT NULL,
  `val` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tpartition by hash(`name`) tpartitions 3
```

The following SQL statement is to be executed for the two tables:

```
UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WHERE id=1;
```

If this statement is directly executed in a PolarDB-X instance, an error will be returned indicating that this statement is not supported. In this case, you can add the PolarDB-X hint to this SQL statement before submitting it to the PolarDB-X instance for execution. The SQL statements are as follows:

```
/*! TDDL:t1.id=1 and t2.id=1*/UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WHERE id=1;
```

This statement will be routed to database shards of `t1`, with the `id` of the database shards being 1. You can run the following `EXPLAIN` command to view the execution plan of this SQL statement:

```
mysql> explain /*! TDDL:t1.id=1 and t2.id=1*/UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) WHERE id=1\G
***** 1. row *****
GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
  SQL: UPDATE `t1_2` AS `t1` SET `val` = (SELECT val FROM `t2_2` AS `t2` WHERE `id` = 1) WHERE `id` = 1
  PARAMS: {}
***** 2. row *****
GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
  SQL: UPDATE `t1_1` AS `t1` SET `val` = (SELECT val FROM `t2_1` AS `t2` WHERE `id` = 1) WHERE `id` = 1
  PARAMS: {}
***** 3. row *****
GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
  SQL: UPDATE `t1_0` AS `t1` SET `val` = (SELECT val FROM `t2_0` AS `t2` WHERE `id` = 1) WHERE `id` = 1
  PARAMS: {}
3 rows in set (0.00 sec)
```

According to the result set of the `EXPLAIN` command, the SQL statement is rewritten into three statements, which are then routed to the database shards for execution. You can further specify a table shard by using the value of the table shard key, to narrow the execution scope of the SQL statement to a specified table shard.

```
mysql> explain /*! TDDL:t1.id=1 and t2.id=1 and t1.name='1'*/UPDATE t1 SET val=(SELECT val FROM t2 WHERE id=1) W
HERE id=1\G
***** 1. row *****
GROUP_NAME: TEST_DRDS_1485327111630IXLWTEST_DRDS_IGHF_0001_RDS
  SQL: UPDATE `t1_1` AS `t1` SET `val` = (SELECT val FROM `t2_1` AS `t2` WHERE `id` = 1) WHERE `id` = 1
  PARAMS: {}
1 row in set (0.00 sec)
```

Note Before using this custom hint, you must ensure that the logical tables `t1` and `t2` are partitioned into the same number of database shards and the same number of table shards. Otherwise, the database shards calculated by the PolarDB-X instance based on the conditions are different, and an error will be returned.

13.9.6. Scan all database shards and table shards

In addition to routing an SQL statement to one or more database shards for execution, PolarDB-X provides a custom hint to allow you to scan all database shards and table shards. With this custom hint, you can route an SQL statement to each database shard at a time. For example, you can use this custom hint to view all the table shards in a specified database shard. In addition, you can use this custom hint to view the data volume of table shards in each database shard corresponding to a specified logical table.

Syntax

With this PolarDB-X hint, you can route an SQL statement to all database shards for execution and route an SQL statement to all database shards to perform an operation on a specified logical table.

- Route an SQL statement to all database shards for execution:

```
/*! TDDL:SCAN*/
```

- Perform an operation on a specified logical table:

```
/*! TDDL:SCAN='table_name'*/
```

The `table_name` parameter indicates the name of a logical table in the logical database of a PolarDB-X instance. This custom hint is provided for table shards in database shards. Ensure that the value of `table_name` is the name of a table shard in database shards.

Note

- PolarDB-X hints can be in the formats of `/*+TDDL:hint_command*/` and `/*! +TDDL:hint_command*/`.
- In the MySQL command-line client, if you need to run an SQL statement that contains a PolarDB-X hint in the format of `/*+TDDL:hint_command*/`, add the `-c` parameter to the logon command, because PolarDB-X hints are based on the [MySQL Comment Syntax](#). Otherwise, the client deletes the PolarDB-X hint and then sends the SQL statement to the server for execution, which causes the hint to fail to take effect. For more information, see [MySQL Client Options](#).

Examples

- View the data volume of a specified broadcast table in each database shard:

```
/*! TDDL:SCAN*/SELECT COUNT(1) FROM table_name
```

In this SQL statement, `table_name` indicates a broadcast table. This hint causes the PolarDB-X instance to route the SQL statement to each database shard for execution. Therefore, the result sets include the total data volume of the broadcast table `table_name` in all database shards. This statement allows you to conveniently check whether the data of a broadcast table is normal.

- Scan a single-database non-partition logical table:

```
/*! TDDL:SCAN*/SELECT COUNT(1) FROM table_name
```

This hint causes the PolarDB-X instance to route the SQL `select count(1) from table_name` statement to each database shard for execution. The `table_name` parameter indicates a logical table in a logical database of a PolarDB-X instance. Before using this hint, ensure that each database shard contains the table shard `table_name`. In other words, the table shard `table_name` is a logical table that is only partitioned into database shards, but not partitioned into table shards. Otherwise, an error that indicates that the table is not found will be returned.

- Scan a partitioned logical table in database shards:

```
/*! TDDL:SCAN='table_name'*/SELECT COUNT(1) FROM table_name
```

When executing this statement, the PolarDB-X instance first calculates all the database shards and table shards corresponding to the logical table `table_name`, and then generates a COUNT clause for each table shard in each database shard.

- View the execution plans of all database shards:

```
/*! TDDL:SCAN='table_name'*/EXPLAIN SELECT * FROM table_name;
```

13.10. Distributed transactions

13.10.1. Distributed transactions based on MySQL 5.7

Note

- When you use MySQL 5.7 or later and PolarDB-X 5.3.4 or later, XA distributed transactions are automatically enabled. The user experience of the XA distributed transactions is the same as that of single-database transactions in MySQL. No special commands are required to enable XA distributed transactions.
- When you use MySQL and a PolarDB-X instance in other versions, see [Distributed transactions based on MySQL 5.6](#).

How it works

When you use MySQL 5.7 or later, the PolarDB-X instance processes distributed transactions based on the XA protocol by default.

Use method

The user experience of distributed transactions in a PolarDB-X instance is the same as that of single-database transactions in MySQL, for example, in terms of the following commands:

- `SET AUTOCOMMIT=0` : Start a transaction.
- `COMMIT` : Commit the current transaction.
- `ROLLBACK` : Roll back the current transaction.

If the SQL statement in a transaction involves only a single shard, the PolarDB-X instance routes the transaction directly to the ApsaraDB RDS for MySQL instance as a single-database transaction. If the SQL statement in the transaction is to modify the data of multiple shards, the PolarDB-X instance automatically upgrades the current transaction to a distributed transaction.

13.10.2. Distributed transactions based on MySQL 5.6

How it works

The XA protocol for MySQL 5.6 is not mature. Therefore, the PolarDB-X instance independently implements two-phase commit (2PC) transaction policies for distributed transactions. When you use MySQL 5.7 or later, we recommend that you use XA transaction policies.

Note The distributed transactions described in this topic are intended for users who use MySQL 5.6 or PolarDB-X earlier than 5.3.4. When you use MySQL 5.7 or later and a PolarDB-X instance in 5.3.4 or later, see [Distributed transactions based on MySQL 5.7](#).

Use method

If a transaction involves multiple database shards, you must declare the current transaction as a distributed transaction. If a transaction involves only a single database shard, you do not need to enable distributed transactions, but can process the transaction as a single-database transaction in MySQL. No additional operations are required.

To enable distributed transactions, do as follows:

After transactions are enabled, run `SET drds_transaction_policy = '...'`.

To enable 2PC transactions in the MySQL command-line client, run the following statements:

```
SET AUTOCOMMIT=0;
SET drds_transaction_policy = '2PC'; -- We recommend that you use MySQL 5.6 to run this command.
.... -- Here, you can run your business SQL statement.
COMMIT; -- You can alternatively write ROLLBACK.
```

To enable 2PC transactions by using the Java database connectivity (JDBC) API, write the code as follows:

```
conn.setAutoCommit(false);
try (Statement stmt = conn.createStatement()) {
    stmt.execute("SET drds_transaction_policy = '2PC'");
}
// ... Here, you can run your business SQL statement.
conn.commit(); // You can alternatively write rollback().
```

FAQ

Q: How can I use PolarDB-X distributed transactions in the Spring framework?

A: If you enable transactions by using the Spring `@Transactional` annotation, you can enable PolarDB-X distributed transactions by extending the transaction manager.

Sample code:

```
import org.springframework.jdbc.datasource.DataSourceTransactionManager;
import org.springframework.transaction.TransactionDefinition;
import javax.sql.DataSource;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;

public class DrdsTransactionManager extends DataSourceTransactionManager {
    public DrdsTransactionManager(DataSource dataSource) {
        super(dataSource);
    }
    @Override
    protected void prepareTransactionalConnection(Connection con, TransactionDefinition definition) throws SQLException {
        try (Statement stmt = con.createStatement()) {
            stmt.executeUpdate("SET drds_transaction_policy = '2PC'"); // A 2PC transaction is used as an example.
        }
    }
}
```

After that, instantiate the preceding class in the Spring configuration. For example, you can write the code as follows:

```
<bean id="drdsTransactionManager" class="my.app.DrdsTransactionManager">
    <property name="dataSource" ref="yourDataSource" />
</bean>
```

To enable PolarDB-X distributed transactions for a class, you can add the `@Transactional("drdsTransactionManager")` annotation.

13.11. DDL operations

13.11.1. DDL statements

The data definition language (DDL) statement `CREATE TABLE` in a PolarDB-X instance is similar to that in a MySQL database, and is extended based on the syntax in a MySQL database. To create a table shard in a PolarDB-X instance, you must specify the table sharding manner and the database sharding manner in the `drds_partition_options` parameter. The valid values include `DBPARTITION BY`, `TBPARTITION BY`, `TBPARTITIONS`, and `BROADCAST`.

Currently, you can run a DDL statement in the following ways:

- Run the DDL statement through the MySQL command-line client, for example, by using MySQL command lines, Navicat, or MySQL Workbench.
- Connect to the specified PolarDB-X instance by using program code and then call the DDL statement for execution.

For the syntax of the `CREATE TABLE` statement in a MySQL database, see [MySQL CREATE TABLE Statement](#).

13.11.2. CREATE TABLE statement

13.11.2.1. Overview

This topic describes the syntax, clauses, parameters, and basic methods for creating a table by using a data definition language (DDL) statement.

 **Note** PolarDB-X instances do not allow you to directly create a database by using a DDL statement. To create a database, you can [Log on to the PolarDB-X console](#). For the information about how to create a database, see [Create a database](#).

Syntax

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
  (create_definition,...)
  [table_options]
  [drds_partition_options]
  [partition_options]
drds_partition_options:
  DBPARTITION BY
    HASH([column])
  [TBPARTITION BY
    { HASH(column)
    | {MM|DD|WEEK|MMDD}(column)}
  [TBPARTITIONS num]
]
```

Clauses and parameters for database and table sharding

- `DBPARTITION BY hash(partition_key)` : This parameter specifies the shard key and the sharding algorithm for database sharding. Database sharding by time is not supported.
- `TBPARTITION BY { HASH(column) | {MM|DD|WEEK|MMDD}(column)}` : (Optional) This parameter specifies the method of mapping data to a physical table. The value is the same as that of `DBPARTITION BY` by default.
- `TBPARTITIONS num` : (Optional) This parameter specifies the number of physical tables to be created in each database shard. The default value is 1. If no table sharding is required, you do not need to specify this parameter.

13.11.2.2. Create a single-database non-partition table

This topic describes how to create a single-database non-partition table.

Create a single-database non-partition table

```
CREATE TABLE single_tbl(
  id int,
  name varchar(30),
  primary key(id)
);
```

According to the node topology of the logical table, you can see that a single-database non-partition logical table is created in database 0.

```
mysql> show topology from single_tbl;
+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | single_tbl |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Specify parameters

You can also specify the `select_statement` parameter when creating a single-database non-partition table. If you need to create table shards, you cannot specify this parameter.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
  [(create_definition,...)]
  [table_options]
  [partition_options]
  select_statement
```

For example, you can run the following statement to create a single-database non-partition table `single_tbl2` to store the data from the `single_tbl` table. In this case, no sharding is required.

```
CREATE TABLE single_tbl2(
  id int,
  name varchar(30),
  primary key(id)
) select * from single_tbl;
```

13.11.2.3. Create a non-partition table in database shards

This topic describes how to create a non-partition table in database shards.

Assume that eight database shards have been created. You can run the following command to create a non-partition table in the database shards by calculating the hash function based on the `userId` shard key.

```
CREATE TABLE multi_db_single_tbl(
  id int,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id);
```

According to the node topology of the logical table, you can see that a table shard is created in each database shard. In other words, the table is only distributed to database shards.

```
mysql> show topology from multi_db_single_tbl;
+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_single_tbl |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_single_tbl |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_single_tbl |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_single_tbl |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_single_tbl |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_single_tbl |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_single_tbl |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_single_tbl |
+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

13.11.2.4. Create table shards in database shards

This topic describes how to create table shards in database shards in different sharding manners.

- Use HASH for sharding
- Use RANGE_HASH for sharding
- Use date functions for sharding

In the following examples, it is assumed that eight database shards have been created.

Use HASH for sharding

Create a table that is partitioned into table shards in database shards, with each database shard containing three physical tables. The database sharding process calculates the hash by using id as the shard key, and the table sharding process calculates the hash by using bid as the shard key. Specifically, a hash operation is performed on the data of the table based on the id column, to distribute the data to multiple database shards. Then, a hash operation is performed on the data in each database shard based on the bid column, to distribute the data to the three physical tables.

```
CREATE TABLE multi_db_multi_tbl(
  id int auto_increment,
  bid int,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id) tpartition by hash(bid) tpartitions 3;
```

According to the node topology of the logical table, you can see that three table shards are created in each database shard.

```
mysql> show topology from multi_db_multi_tbl;
```

```
+-----+-----+-----+-----+
| ID | GROUP_NAME          | TABLE_NAME      |
+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_00 |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_01 |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | multi_db_multi_tbl_02 |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_03 |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_04 |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | multi_db_multi_tbl_05 |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_06 |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_07 |
| 8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | multi_db_multi_tbl_08 |
| 9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_09 |
| 10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_10 |
| 11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | multi_db_multi_tbl_11 |
| 12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_12 |
| 13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_13 |
| 14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | multi_db_multi_tbl_14 |
| 15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_15 |
| 16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_16 |
| 17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | multi_db_multi_tbl_17 |
| 18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_18 |
| 19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_19 |
| 20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | multi_db_multi_tbl_20 |
| 21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_21 |
| 22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_22 |
| 23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | multi_db_multi_tbl_23 |
+-----+-----+-----+-----+
24 rows in set (0.01 sec)
```

According to the sharding rule of the logical table, you can see that both database sharding and table sharding use the hash function, except that the database shard key is id and the table shard key is bid.

```
mysql> show rule from multi_db_multi_tbl;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | multi_db_multi_tbl | 0 | id | hash | 8 | bid | hash | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Use RANGE_HASH for sharding

- Requirements

The shard key must be a character or a number.

- Routing method

Calculate a hash value based on the last N digits of any shard key, and then calculate the route by using RANGE_HASH. The number N is the third parameter in the function. For example, during calculation of the RANGE_HASH(COL1, COL2, N) function, COL1 is preferentially selected and then truncated to obtain the last N digits for calculation. If COL1 does not exist, COL2 is selected and truncated for calculation.

- Scenarios

RANGE_HASH is applicable to scenarios where two shard keys are used for sharding but only the value of one shard is used for SQL query. Assume that a PolarDB-X database is partitioned into eight physical databases. Our customer has the following requirements:

- i. The order table of each service needs to be partitioned into database shards by buyer ID and order ID.
- ii. The query is executed based on either the buyer ID or order ID as the condition.

In this case, you can run the following DDL statement to create the order table:

```
create table test_order_tb (
    id int,
    seller_id varchar(30) DEFAULT NULL,
    order_id varchar(30) DEFAULT NULL,
    buyer_id varchar(30) DEFAULT NULL,
    create_time datetime DEFAULT NULL,
    primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by RANGE_HASH(buyer_id, order_id, 10) tpartition by RANGE_HASH(buyer_id, order_id, 10) tpartitions 3;
```

 Note

- Neither of the two shard keys can be modified.
- Data insertion fails if the two shard keys point to different database shards or table shards.

Use date functions for sharding

In addition to using the hash function as the sharding algorithm, you can also use the date functions MM, DD, WEEK, and MMDD as the table sharding algorithms. For more information, see the following examples:

- Create a table that is partitioned into table shards in database shards. The database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates `DAY_OF_WEEK` through the `WEEK(actionDate)` function and then partitions the table into table shards based on the `actionDate` column, with one week counted as seven days.

For example, if the value in the `actionDate` column is 2017-02-27, which is on Monday, the value obtained by calculating the `WEEK(actionDate)` function is 2. In this case, the record is stored in table shard 2, because $2 \% 7 = 2$. This table shard is located in a database shard and is named `user_log_2`. For another example, if the value in the `actionDate` column is 2017-02-26, which is on Sunday, the value obtained by calculating the `WEEK(actionDate)` function is 1. In this case, the record is stored in table shard 1, because $1 \% 7 = 1$. This table shard is located in a database shard and is named `user_log_1`.

```
CREATE TABLE user_log(  
  userId int,  
  name varchar(30),  
  operation varchar(30),  
  actionDate DATE  
) dbpartition by hash(userId) tpartition by WEEK(actionDate) tpartitions 7;
```

According to the node topology of the logical table, you can see that seven table shards are created in each database shard, because one week is counted as seven days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_0 |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_1 |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_2 |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_3 |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_4 |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_5 |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log_6 |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_0 |
| 8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_1 |
| 9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_2 |
| 10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_3 |
| 11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_4 |
| 12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_5 |
| 13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log_6 |
...
| 49 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_0 |
| 50 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_1 |
| 51 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_2 |
| 52 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_3 |
| 53 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_4 |
| 54 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_5 |
| 55 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log_6 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
56 rows in set (0.01 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates the WEEK function by using `actionDate` as the shard key.

```
mysql> show rule from user_log;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | user_log | 0 | userId | hash | 8 | actionDate | week | 7 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

According to the specified database and table shard key parameters, you can see the specific physical table in the specific physical database to which the SQL statement is routed.

View the route of the SQL statement

```
mysql> explain select name from user_log where userId = 1 and actionDate = '2017-02-27'\G
***** 1_row *****
GROUP_NAME: SANGUAN_1490167540907XNDV5SANGUAN_B5QT_0001_RDS
SQL: select `user_log`.`name` from `user_log_2` `user_log` where ((`user_log`.`userId` = 1) AND (`user_log`.`actionDate` = '2017-02-27'))
PARAMS: {}
1 row in set (0.01 sec)
```

- Create a table that is partitioned into table shards in database shards. The database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates `MONTH_OF_YEAR` through the `MM(actionDate)` function and then partitions the table into table shards based on the `actionDate` column, with one year counted as 12 months.

For example, if the value in the `actionDate` column is 2017-02-27, the value obtained by calculating the `MM(actionDate)` function is 02. In this case, the record is stored in table shard 02, because $02 \% 12 = 02$. This table shard is located in a database shard and is named `user_log_02`. For another example, if the value in the `actionDate` column is 2016-12-27, the value obtained by calculating the `MM(actionDate)` function is 12. In this case, the record is stored in table shard 00, because $12 \% 12 = 00$. This table shard is located in a database shard and is named `user_log_00`.

```
CREATE TABLE user_log2(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
) dbpartition by hash(userId) tpartition by MM(actionDate) tpartitions 12;
```

According to the node topology of the logical table, you can see that 12 table shards are created in each database shard, because one year is counted as 12 months in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log2;
+-----+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_00 |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_01 |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_02 |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_03 |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_04 |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_05 |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_06 |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_07 |
| 8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_08 |
| 9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_09 |
| 10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_10 |
| 11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log2_11 |
| 12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_00 |
| 13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_01 |
| 14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_02 |
| 15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_03 |
| 16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_04 |
| 17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_05 |
| 18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_06 |
| 19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_07 |
| 20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_08 |
| 21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_09 |
| 22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_10 |
| 23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | user_log2_11 |
...
| 84 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_00 |
| 85 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_01 |
| 86 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_02 |
| 87 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_03 |
| 88 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_04 |
| 89 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_05 |
| 90 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_06 |
| 91 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_07 |
| 92 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_08 |
| 93 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_09 |
| 94 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_10 |
| 95 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log2_11 |
+-----+-----+-----+-----+
96 rows in set (0.02 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates the `MM` function by using `actionDate` as the shard key.

```
mysql> show rule from user_log2;
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | user_log2 | 0 | userId | hash | 8 | actionDate | mm | 12 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- Create a table that is partitioned into table shards in database shards. The database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates `DAY_OF_MONTH` through the `DD(actionDate)` function and then partitions the table into table shards, with one month counted as 31 days.

For example, if the value in the `actionDate` column is 2017-02-27, the value obtained by calculating the `DD(actionDate)` function is 27. In this case, the record is stored in table shard 27, because $27 \% 31 = 27$. This table shard is located in a database shard and is named `user_log_27`.

```
CREATE TABLE user_log3(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
) dbpartition by hash(userId) tpartition by DD(actionDate) tpartitions 31;
```

According to the node topology of the logical table, you can see that 31 table shards are created in each database shard, because one month is counted as 31 days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log3;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_00 |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_01 |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_02 |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_03 |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_04 |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_05 |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_06 |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_07 |
| 8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_08 |
| 9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_09 |
| 10 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_10 |
| 11 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_11 |
```

```

| 12 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_12 |
| 13 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_13 |
| 14 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_14 |
| 15 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_15 |
| 16 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_16 |
| 17 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_17 |
| 18 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_18 |
| 19 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_19 |
| 20 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_20 |
| 21 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_21 |
| 22 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_22 |
| 23 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_23 |
| 24 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_24 |
| 25 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_25 |
| 26 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_26 |
| 27 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_27 |
| 28 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_28 |
| 29 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_29 |
| 30 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log3_30 |
...
| 237 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_20 |
| 238 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_21 |
| 239 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_22 |
| 240 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_23 |
| 241 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_24 |
| 242 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_25 |
| 243 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_26 |
| 244 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_27 |
| 245 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_28 |
| 246 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_29 |
| 247 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log3_30 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
248 rows in set (0.01 sec)

```

According to the sharding rule of the logical table, you can see that the database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates the DD function by using `actionDate` as the shard key.

```
mysql> show rule from user_log3;
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | user_log3 | 0 | userId | hash | 8 | actionDate | dd | 31 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

- Create a table that is partitioned into table shards in database shards. The database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates `DAY_OF_YEAR % 365` through the `MMDD(actionDate) tpartitions 365` function and then partitions the table into 365 physical tables, with one year counted as 365 days.

For example, if the value in the `actionDate` column is 2017-02-27, the value obtained by calculating the `MMDD(actionDate)` function is 58. In this case, the record is stored in table shard 58. This table shard is located in a database shard and is named `user_log_58`.

```
CREATE TABLE user_log4(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
) dbpartition by hash(userId) tpartition by MMDD(actionDate) tpartitions 365;
```

According to the node topology of the logical table, you can see that 365 table shards are created in each database shard, because one year is counted as 365 days in the function. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log4;
+-----+-----+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+
...
| 2896 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_341 |
| 2897 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_342 |
| 2898 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_343 |
| 2899 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_344 |
| 2900 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_345 |
| 2901 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_346 |
| 2902 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_347 |
| 2903 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_348 |
| 2904 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_349 |
| 2905 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_350 |
| 2906 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_351 |
| 2907 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_352 |
| 2908 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_353 |
| 2909 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_354 |
| 2910 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_355 |
| 2911 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_356 |
| 2912 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_357 |
| 2913 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_358 |
| 2914 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_359 |
| 2915 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_360 |
| 2916 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_361 |
| 2917 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_362 |
| 2918 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_363 |
| 2919 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log4_364 |
+-----+-----+-----+-----+-----+-----+
2920 rows in set (0.07 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates the MMDD function by using `actionDate` as the shard key.

```
mysql> show rule from user_log4;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | user_log4 | 0 | userId | hash | 8 | actionDate | mmdd | 365 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

- Create a table that is partitioned into table shards in database shards. The database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates `DAY_OF_YEAR % 10` through the `MMDD(actionDate)` `tbpartitions 10` function and then partitions the table into 10 physical tables, with one year counted as 365 days.

```
CREATE TABLE user_log5(
  userId int,
  name varchar(30),
  operation varchar(30),
  actionDate DATE
) dbpartition by hash(userId) tbpartition by MMDD(actionDate) tbpartitions 10;
```

According to the node topology of the logical table, you can see that 10 table shards are created in each database shard, because one year is counted as 365 days in the function and the table data is routed to 10 physical tables. The responses are very long, and therefore are omitted by using an ellipsis (...).

```
mysql> show topology from user_log5;
+-----+-----+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+-----+
| 0 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_00 |
| 1 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_01 |
| 2 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_02 |
| 3 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_03 |
| 4 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_04 |
| 5 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_05 |
| 6 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_06 |
| 7 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_07 |
| 8 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_08 |
| 9 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | user_log5_09 |
...
| 70 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_00 |
| 71 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_01 |
| 72 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_02 |
| 73 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_03 |
| 74 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_04 |
| 75 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_05 |
| 76 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_06 |
| 77 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_07 |
| 78 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_08 |
| 79 | SANGUAN_TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | user_log5_09 |
+-----+-----+-----+-----+
80 rows in set (0.02 sec)
```

According to the sharding rule of the logical table, you can see that the database sharding process calculates the hash by using `userId` as the shard key, and the table sharding process calculates the `MMDD` function by using `actionDate` as the shard key, and then routing the table data to 10 physical tables.

```
mysql> show rule from user_log5;
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ID | TABLE_NAME | BROADCAST | DB_PARTITION_KEY | DB_PARTITION_POLICY | DB_PARTITION_COUNT | TB_PARTITION_KEY | TB_PARTITION_POLICY | TB_PARTITION_COUNT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | user_log5 | 0 | userId | hash | 8 | actionDate | mmdd | 10 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

13.11.2.5. Use the primary key as the shard key

When no shard key is specified for the sharding algorithm, the system uses the primary key as the shard key by default. The following illustrates how to use the primary key as the database shard key and the table shard key.

Use the primary key as the database shard key

```
CREATE TABLE prmkey_tbl(
id int,
name varchar(30),
primary key(id)
) dbpartition by hash();
```

Use the primary key as the database shard key and the table shard key

```
CREATE TABLE prmkey_multi_tbl(
id int,
name varchar(30),
primary key(id)
) dbpartition by hash() tpartition by hash() tpartitions 3;
```

13.11.2.6. Create a broadcast table

The BROADCAST clause is used to specify a broadcast table to be created. A broadcast table is replicated to each database shard and data consistency is ensured between the database shards by using a synchronization mechanism with a delay of several seconds. This feature allows you to route a JOIN operation from a Cloud Native Distributed Database PolarDB-X (PolarDB-X) instance to an underlying ApsaraDB RDS for MySQL instance to prevent the JOIN operation from being performed in multiple databases. [Overview](#) describes how to optimize SQL statements by using broadcast tables.

The following is an example statement for creating a broadcast table:

```
CREATE TABLE brd_tbl(
  id int,
  name varchar(30),
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 BROADCAST;
```

13.11.2.7. Other attributes of the MySQL CREATE TABLE statement

When creating table shards in database shards, you can also specify other attributes of the table shards in the MySQL CREATE TABLE statement. For example, you can specify other attributes as follows:

```
CREATE TABLE multi_db_multi_tbl(
  id int,
  name varchar(30),
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(id) tpartition by hash(id) tpartitions 3;
```

13.11.3. ALTER TABLE statement

The syntax of the ALTER TABLE statement used to modify a table is as follows:

```
ALTER [ONLINE|OFFLINE] [IGNORE] TABLE tbl_name
  [alter_specification [, alter_specification] ...]
  [partition_options]
```

In a PolarDB-X instance, you can use this data definition language (DDL) statement to perform routine DDL operations, such as adding a column, adding an index, and modifying a data definition. For more information about the syntax, see [MySQL ALTER TABLE Statement](#).

 **Note** If you need to modify a table shard, you are not allowed to modify the shard key.

- Add a column:

```
ALTER TABLE user_log
  ADD COLUMN idcard varchar(30);
```

- Add an index:

```
ALTER TABLE user_log
  ADD INDEX idcard_idx (idcard);
```

- Delete an index:

```
ALTER TABLE user_log
  DROP INDEX idcard_idx;
```

- Modify a field:

```
ALTER TABLE user_log
  MODIFY COLUMN idcard varchar(40);
```

13.11.4. DROP TABLE statement

The syntax of the DROP TABLE statement used to delete a table is as follows:

```
DROP [TEMPORARY] TABLE [IF EXISTS]
tbl_name [, tbl_name] ...
[RESTRICT | CASCADE]
```

The DROP TABLE statement in a PolarDB-X instance is the same as the DROP TABLE statement in a MySQL database. After the statement is executed, the system automatically deletes the corresponding physical table. For more information about the syntax, see [MySQL DROP TABLE Statement](#).

For example, you can run the following statement to delete the user_log table:

```
DROP TABLE user_log;
```

13.11.5. FAQ about DDL statements

What can I do if an error occurs during table creation?

Data definition language (DDL) statements in a PolarDB-X instance are processed in a distributed manner. If an error occurs, the structures of all table shards are inconsistent from each other. Therefore, you need to perform manual cleanup.

Perform the following steps:

1. Check the basic error descriptions provided by the PolarDB-X instance, such as syntax errors. If the error message is too long, the system will prompt you to call the SHOW WARNINGS command to view the failure cause of each database shard.
2. Run the SHOW TOPOLOGY command to view the topology of physical tables.

```
SHOW TOPOLOGY FROM multi_db_multi_tbl;
+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+
| 0 | corona_qatest_0 | multi_db_multi_tbl_00 |
| 1 | corona_qatest_0 | multi_db_multi_tbl_01 |
| 2 | corona_qatest_0 | multi_db_multi_tbl_02 |
| 3 | corona_qatest_1 | multi_db_multi_tbl_03 |
| 4 | corona_qatest_1 | multi_db_multi_tbl_04 |
| 5 | corona_qatest_1 | multi_db_multi_tbl_05 |
| 6 | corona_qatest_2 | multi_db_multi_tbl_06 |
| 7 | corona_qatest_2 | multi_db_multi_tbl_07 |
| 8 | corona_qatest_2 | multi_db_multi_tbl_08 |
| 9 | corona_qatest_3 | multi_db_multi_tbl_09 |
| 10 | corona_qatest_3 | multi_db_multi_tbl_10 |
| 11 | corona_qatest_3 | multi_db_multi_tbl_11 |
+-----+-----+-----+
12 rows in set (0.21 sec)
```

3. Run the `CHECK TABLE tablename` command to check whether the logical table has been created. For example, the following response indicates that a physical table corresponding to the logical table

multi_db_multi_tbl failed to be created.

```
mysql> check table multi_db_multi_tbl;
+-----+-----+-----+-----+
| TABLE                | OP | MSG_TYPE | MSG_TEXT                |
+-----+-----+-----+-----+
| andor_mysql_qatest. multi_db_multi_tbl | check | Error   | Table 'corona_qatest_0. multi_db_multi_tbl_02' doesn't exist |
+-----+-----+-----+-----+
1 row in set (0.16 sec)
```

- Continue to create or delete the table in idempotent mode to create or delete the remaining physical tables.

```
CREATE TABLE IF NOT EXISTS table1
(id int, name varchar(30), primary key(id))
dbpartition by hash(id);
DROP TABLE IF EXISTS table1;
```

What can I do if I failed to create an index or add a column?

The method for handling the failure in creating an index or adding a column is similar to that for the failure in creating a table. For more information, see [Handle DDL exceptions](#).

13.11.6. DDL functions for sharding

13.11.6.1. Overview

PolarDB-X is a database service that supports both database sharding and table sharding.

Support for PolarDB-X database sharding and table sharding

The following table lists the support for database sharding and table sharding in PolarDB-X data definition language (DDL) sharding functions.

Sharding function	Description	Support for database sharding	Support for table sharding
HASH	Performs a simple modulus operation.	Yes	Yes
UNI_HASH	Performs a simple modulus operation.	Yes	Yes
RIGHT_SHIFT	Shifts the value to the right.	Yes	Yes
RANGE_HASH	Performs double hashing.	Yes	Yes
MM	Performs hashing by month.	No	Yes
DD	Performs hashing by date.	No	Yes
WEEK	Performs hashing by week.	No	Yes

Sharding function	Description	Support for database sharding	Support for table sharding
MMDD	Performs hashing by month and date.	No	Yes
YYYYMM	Performs hashing by year and month.	Yes	Yes
YYYYWEEK	Performs hashing by year and week.	Yes	Yes
YYYYDD	Performs hashing by year and date.	Yes	Yes
YYYYMM_OPT	Performs optimized hashing by year and month.	Yes	Yes
YYYYWEEK_OPT	Performs optimized hashing by year and week.	Yes	Yes
YYYYDD_OPT	Performs optimized hashing by year and date.	Yes	Yes

-  **Note** When using database sharding and table sharding in PolarDB-X, note the following:
- In a PolarDB-X instance, the sharding method of a logical table is defined jointly by a sharding function and a shard key. The sharding function contains the number of shards to be created and the routing algorithm. The shard key also specifies the MySQL data type of the shard key.
 - When the database sharding function is the same as the table sharding function and the database shard key is the same as the table shard key in a PolarDB-X instance, the same sharding method is used for database sharding and table sharding. This allows the PolarDB-X instance to uniquely locate one physical table in a physical database based on the value of the shard key.
 - If the database sharding method and the table sharding method of a logical table are different and an SQL query does not contain both database shard key and table shard key, the PolarDB-X instance scans all database shards or all table shards when processing the SQL query.

Support for data types of PolarDB-X DDL sharding functions

Different PolarDB-X DDL sharding functions support different data types. The following table lists the support for various data types in PolarDB-X sharding functions (✓ indicates supported and × indicates not supported).

Support for data types in PolarDB-X DDL sharding functions

Sharding function	BIGINT	INT	MEDIUMINT	SMALLINT	TINYINT	VARCHAR	CHAR	DATE	DATETIME	TIMESTAMP	Other types
HASH	√	√	√	√	√	√	√	x	x	x	x
UNI_HASH	√	√	√	√	√	√	√	x	x	x	x
RANGE_HASH	√	√	√	√	√	√	√	x	x	x	x
RIGHT_SHIFT	√	√	√	√	√	x	x	x	x	x	x
MM	x	x	x	x	x	x	x	√	√	√	x
DD	x	x	x	x	x	x	x	√	√	√	x
WEEK	x	x	x	x	x	x	x	√	√	√	x
MMDD	x	x	x	x	x	x	x	√	√	√	x
YYYYMM	x	x	x	x	x	x	x	√	√	√	x
YYYYWEEK	x	x	x	x	x	x	x	√	√	√	x
YYYYDD	x	x	x	x	x	x	x	√	√	√	x
YYYYMM_OPT	x	x	x	x	x	x	x	√	√	√	x
YYYYWEEK_OPT	x	x	x	x	x	x	x	√	√	√	x
YYYYDD_OPT	x	x	x	x	x	x	x	√	√	√	x

Syntax description for PolarDB-X DDL sharding functions

PolarDB-X is compatible with the CREATE TABLE statement in MySQL, and additionally provides the `drds_partition_options` keyword to support database sharding and table sharding:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [drds_partition_options]
    [partition_options]
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [drds_partition_options]
    [partition_options]
    select_statement
drds_partition_options:
    DBPARTITION BY
        { {HASH|YYYYMM|YYYYWEEK|YYYYDD|YYYYMM_OPT|YYYYWEEK_OPT|YYYYDD_OPT}{(column)}}
    [TBPARTITION BY
        { {HASH|MM|DD|WEEK|MMDD|YYYYMM|YYYYWEEK|YYYYDD|YYYYMM_OPT|YYYYWEEK_OPT|YYYYDD_OPT}(column)
    }
    [TBPARTITIONS num]
]
```

13.11.6.2. HASH

Requirements

- The shard key must be an integer or a string.
- This sharding function has no requirements on the version of a PolarDB-X instance. It supports all PolarDB-X instances by default.

Routing method

When the HASH function is run by using different shard keys for database sharding and table sharding, perform the remainder operation on the value of the database shard key based on the number of database shards. If the value of the shard key is a string, the string is converted to a hash value before route calculation. For example, `HASH('8')` is equivalent to $8 \% D$, where `D` indicates the number of database shards.

When the HASH function is run by using the same shard key for both database sharding and table sharding, perform the remainder operation on the value of the shard key based on the total number of table shards. For example, assume that two database shards are created, each database shard contains four table shards, table shards 0 to 3 are stored in database shard 0, and table shards 4 to 7 are stored in database shard 1. If a key value is 15, the key value 15 is distributed to table shard 7 in database shard 1, because $15 \% (2 \times 4) = 7$.

Scenarios

- HASH is applicable when database sharding is implemented by user ID or order ID.
- HASH is also applicable when the shard key is a string.

Examples

If you need to create a non-partition table in database shards by using the HASH function based on the ID column, you can use the following CREATE TABLE statement:

```
create table test_hash_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by HASH(ID);
```

Precautions

The HASH is a simple modulus operation. The output distribution of the HASH function can be even only when the values in the partition column are evenly distributed.

13.11.6.3. UNI_HASH

Requirements

- The shard key must be an integer or a string.
- The version of the PolarDB-X instance must be 5.1.28-1508068 or later. For more information about the PolarDB-X release notes, see [View the instance version](#).

Routing method

When the UNI_HASH function is used for database sharding, perform a remainder operation on the value of the database shard key based on the number of database shards. If the value of the shard key is a string, the string is converted to a hash value before route calculation. For example, `HASH('8')` is equivalent to $8 \% D$, where `D` indicates the number of database shards.

When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, perform the remainder operation on the value of the database shard key based on the number of database shards first (this step is different from that in the HASH function). Then, the data is evenly distributed to the table shards in the database shard.

Scenarios

- UNI_HASH is applicable when database sharding is implemented by user ID or order ID.
- UNI_HASH is also applicable when the shard key is an integer or a string.

- UNI_HASH can be used when the following conditions are met: Two logical tables need to be partitioned into different numbers of table shards in database shards based on the same shard key. In addition, the two tables are frequently joined by using a JOIN statement based on the shard key.

Comparison with HASH

When you use the UNI_HASH function to create a non-partition table in database shards, the routing method is the same as that used in the HASH function. Specifically, the route is calculated by performing the remainder operation on the key value of the database shard key based on the number of database shards.

When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, as the number of table shards changes, the database shard route calculated based on the same key value may also change.

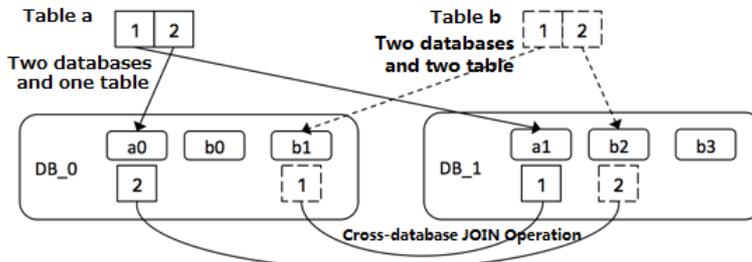
When the UNI_HASH function is run by using the same shard key for both database sharding and table sharding, the database shard route calculated based on the same key value is always the same regardless of the number of table shards.

If two logical tables need to be partitioned into different table shards in database shards based on the same shard key, when the two tables are joined by using the HASH function based on the shard key, multi-database join may occur. However, when the two tables are joined by using the UNI_HASH function based on the shard key, multi-database join does not occur.

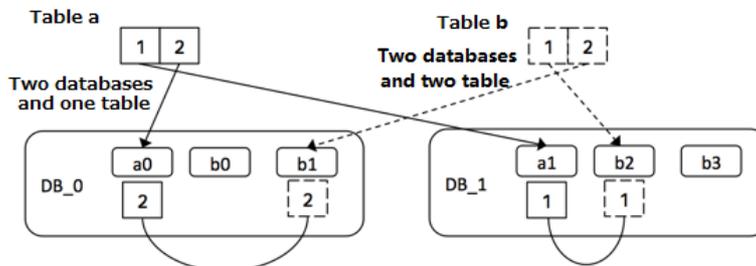
Assume that you have two database shards and two logical tables, and each database shard in logical table a stores one table shard and each database shard in logical table b stores two table shards. The following figures separately show the results of a JOIN query for logical tables a and b after the HASH function is used for sharding and the results of a JOIN query for logical tables a and b after the HASH function is used for sharding.

Comparison between HASH and UNI_HASH

HASH sharding: Two logical tables have different numbers of physical table shards. The same shard key is used in different database shards. Cross-database JOIN queries may be performed.



UNI_HASH sharding: Two logical tables have different numbers of physical table shards. The same shard key is used in the same database shard. No cross-database JOIN queries are performed.



Examples

If you need to create four table shards in each database shard by using the UNI_HASH function based on the ID column, you can run the following CREATE TABLE statement:

```
create table test_hash_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by UNI_HASH(ID)  
tbpartmention by UNI_HASH(ID) tpartitions 4;
```

Precautions

The UNI_HASH is a simple modulus operation. The output distribution of the UNI_HASH function can be even only when the values in the shard column are evenly distributed.

13.11.6.4. RIGHT_SHIFT

Requirements

- The shard key must be an integer.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Shift the value of the database shard key to the right by a specified number of binary digits, and then perform the remainder operation on the obtained integer based on the number of database shards or table shards. In particular, you can specify the number of shifted digits by running a data definition language (DDL) statement.

Scenarios

RIGHT_SHIFT is applicable to improve the evenness of the hash results when the lower-digit parts of most shard key values are very similar to each other but the higher-digit parts vary greatly.

Assume that four shard key values are available: 12340000, 12350000, 12460000, and 12330000. The four lower digits of the four values are all 0000. Directly hashing the values of the shard keys outputs poor results. However, if you run the RIGHT_SHIFT(shardKey, 4) statement to shift the values of the shard keys to the right by four digits, to obtain 1234, 1235, 1246, and 1233, the hashing results are improved.

Examples

If you need to use the ID column as a shard key and shift the values of the ID column to the right by four binary digits to obtain hash values, you can run the following CREATE TABLE statement:

```
create table test_hash_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by RIGHT_SHIFT(id, 4)  
tbpartmention by RIGHT_SHIFT(id, 4) tpartitions 2;
```

Precautions

The number of shifted digits cannot exceed the number of digits occupied by the integer.

13.11.6.5. RANGE_HASH

Requirements

- The shard key must be a character or a number.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Calculate the hash value based on the last N digits of any shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes the route computing. The number N is the third parameter in the function.

For example, during calculation of the RANGE_HASH(COL1, COL2, N) function, COL1 is preferentially selected and then truncated to obtain the last N digits for calculation. If COL1 does not exist, COL2 is selected and truncated for calculation.

Scenarios

RANGE_HASH is applicable to scenarios where a table needs to be partitioned by two shard keys but query is performed only based on the value of one shard key.

Examples

Assume that a PolarDB-X database is partitioned into eight physical databases. Our customer has the following requirements:

The order table of a business needs to be partitioned into database shards by buyer ID and order ID. The query is executed based on either the buyer ID or order ID as the condition.

In this case, you can run the following DDL statement to create the order table:

```
create table test_order_tb (
  id int,
  buyer_id varchar(30) DEFAULT NULL,
  order_id varchar(30) DEFAULT NULL,
  create_time datetime DEFAULT NULL,
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by RANGE_HASH(buyer_id,order_id, 10)
tbpartment by RANGE_HASH (buyer_id,order_id, 10) tbpartitions 3;
```

Precautions

- Neither of the two shard keys can be modified.
- Data insertion fails if the two shard keys point to different database shards or table shards.

13.11.6.6. MM

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- MM is only applicable to table sharding.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Perform the remainder operation based on the month that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

MM can be used to partition tables by month. The table shard name indicates a specific month.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by month, and map every month to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_mm_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by HASH(id)  
tbpartmention by MM(create_time) tpartitions 12;
```

Precautions

When you partition tables with MM, ensure that each database shard has no more than 12 table shards because a year has 12 months.

13.11.6.7. DD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- DD is only applicable to table sharding.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Perform the remainder operation based on the day of the month that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

DD can be used to partition tables based on a specified number of days in a month, that is, a date. The subscript of the table shard name indicates the day in a month. A month has 31 days at most.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by day, and map every day to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_dd_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by HASH(id)  
tbpartmention by DD(create_time) tpartitions 31;
```

Precautions

When you partition tables with DD, ensure that each database shard has no more than 31 table shards because a month has 31 days at most.

13.11.6.8. WEEK

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- WEEK is only applicable to table sharding.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Perform the remainder operation based on the day of a week that corresponds to the time value of the database shard key to obtain the table shard subscript.

Scenarios

WEEK can be used to partition tables based on days in a week. The subscript of the table shard name corresponds to each day of a week, from Monday to Sunday.

Examples

Assume that we need to perform database sharding by ID, perform table sharding for the create_time column by week, and map every day of a week (from Monday to Sunday) to a physical table. The data definition language (DDL) statement is as follows:

```
create table test_week_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by HASH(name)  
tbpartmention by WEEK(create_time) tpartitions 7;
```

Precautions

When you partition tables with WEEK, ensure that each database shard has no more than seven table shards because a week has seven days.

13.11.6.9. MMDD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- MMDD is only applicable to table sharding.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Perform the remainder operation based on the number of days in a year that corresponds to the time value of the database shard key to obtain the table sharding subscript.

Scenarios

MMDD can be used to partition tables based on the number of days in a year that corresponds to a date in that year. The subscript of the table shard name indicates the day in that year, with a maximum of 366 days in a year.

Examples

Assume that we need to perform database sharding by ID, create tables for the create_time column by date (month-day), and map every day of a year to a physical table. The data definition language (DDL) statement is as follows.

```
create table test_mmdd_tb (  
  id int,  
  name varchar(30) DEFAULT NULL,  
  create_time datetime DEFAULT NULL,  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by HASH(name)  
tbpartment by MMDD(create_time) tpartitions 365;
```

Precautions

When you partition tables with MMDD, ensure that each database shard has no more than 366 table shards because a year has 366 days at most.

13.11.6.10. YYYYMM

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Calculate the hash value based on the year and months of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYMM('2012-12-31 12:12:12') is equivalent to $(2012 \times 12 + 12) \% D$, where D indicates the number of database shards.

Scenarios

YYYYMM can be used to partition databases by year and month. We recommend that you use YYYYMM with tbpartition YYYYMM(ShardKey).

Assume that a PolarDB-X database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and month.
- Distribute data from every month within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYMM. For the requirement of distributing data from every month within two years to a table shard (that is, one table shard stores the data of one month), create at least 24 physical table shards because a year has 12 months. Create three physical table shards for each database shard because the PolarDB-X instance contains eight database shards. The data definition language (DDL) statement is as follows.

```
create table test_yyyymm_tb (
  id int,
  name varchar(30) DEFAULT NULL,
  create_time datetime DEFAULT NULL,
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by YYYYMM(create_time)
tbpartmention by YYYYMM(create_time) tbpartitions 3;
```

Precautions

- YYYYMM does not support distributing data from every month in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over months (for example, a cycle exists between 2012-03 and 2013-03), data from the same month may be routed to the same database or table shard, depending on the actual number of table shards.

13.11.6.11. YYYYWEEK

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Calculate the hash value based on the year and weeks of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYWEEK('2012-12-31 12:12:12') is equivalent to $(2013 \times 52 + 1) \% D$, with the date 2012-12-31 falling on the first week of 2013, where D indicates the number of database shards.

Scenarios

YYYYWEEK can be used to partition databases by year and the number of weeks in a year. We recommend that you use YYYYWEEK with tbpartition YYYYWEEK(ShardKey).

Assume that a PolarDB-X database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and by week.
- Distribute data from every week within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYWEEK. For the requirement of distributing data from every week within two years to a table shard (that is, one table shard stores the data of one week), create at least 106 physical table shards because a year has roughly 53 weeks (rounded). Create 14 physical table shards for each database shard because the PolarDB-X instance contains eight database shards ($14 \times 8 = 112 > 106$). We recommend that the number of table shards be an integer multiple of the number of database shards. The data definition language (DDL) statement is as follows:

```
create table test_yyyymm_tb (
  id int,
  name varchar(30) DEFAULT NULL,
  create_time datetime DEFAULT NULL,
  primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
dbpartition by YYYYWEEK(create_time)
tbpartment by YYYYWEEK(create_time) tbpartitions 14;
```

Precautions

- YYYYWEEK does not support distributing data from every week in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over weeks (for example, a cycle exists between the first week of 2012 and the first week of 2013), data from the same week after a cycle may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.11.6.12. YYYYDD

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Routing method

Calculate the hash value based on the year and days of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.

For example, YYYYDD('2012-12-31 12:12:12') is equivalent to $(2012 \times 366 + 365) \% D$, with 2012-12-31 as the 365th day of 2012, where D indicates the number of database shards.

Scenarios

Database sharding is performed by year and the number of days in a year. We recommend that you use YYYYDD with tbpartition YYYYDD(ShardKey).

Assume that a PolarDB-X database is partitioned into eight physical databases. Our customer has the following requirements:

- Perform database sharding for a service by year and day.
- Distribute data from every week within two years to a separate table shard.
- Distribute a query with the database and table shard keys to a physical table shard of a physical database shard.

The preceding requirements can be met by using YYYYDD. For the requirement of distributing data from every day within two years to a table shard (that is, one table shard stores the data of one day), create at least 732 physical table shards because a year has up to 366 days. Create 92 physical table shards for each database shard because the PolarDB-X instance contains eight database shards ($732/8 = 91.5$, rounded to 92). We recommend that the number of table shards be an integer multiple of the number of database shards. The data definition language (DDL) statement is as follows:

```
create table test_yyyydd_tb (  
    id int,  
    name varchar(30) DEFAULT NULL,  
    create_time datetime DEFAULT NULL,  
    primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
dbpartition by YYYYDD(create_time)  
tbpartment by YYYYDD(create_time) tbpartitions 92;
```

Precautions

- YYYYDD does not support distributing data from every day in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle of a specific date (for example, a cycle exists between 2012-03-01 and 2013-03-01), data from the same date may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.11.6.13. YYYYMM_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The year and month of user data increase naturally over time, rather than randomly.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Optimizations

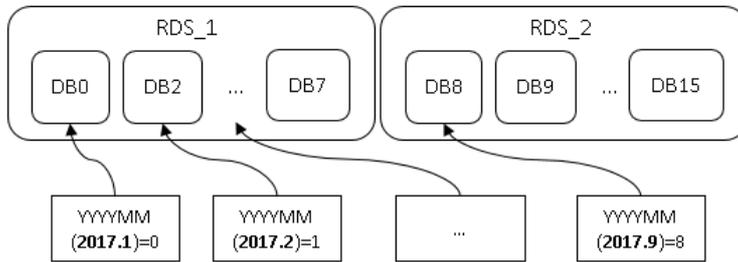
Compared with YYYYMM, YYYYMM_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases.

For example, assume that two ApsaraDB RDS for MySQL instances are attached to a PolarDB-X instance, with 16 database shards. DB0 to DB7 shards are located on one ApsaraDB RDS for MySQL instance, and DB8 to DB15 shards are located on the other ApsaraDB RDS for MySQL instance.

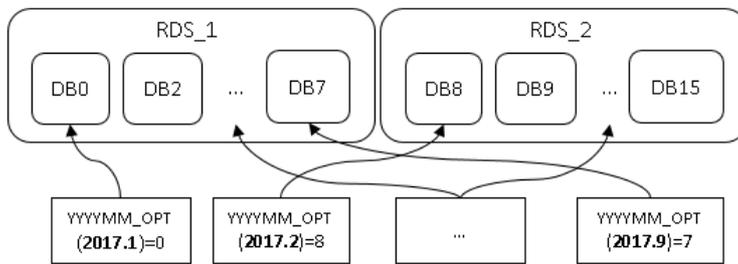
The following figure shows the mappings when YYYYMM and YYYYMM_OPT are used for database sharding, respectively.

Comparison between YYYYMM and YYYYMM_OPT

As the time goes on linearly, YYYYMM fills data in ApsaraDB for RDS instances in sequence (data is first distributed to the database shards of RDS_1, then to the database shards of RDS_2, and then to the database shards of RDS_1 again).



YYYYMM_OPT evenly distributes data between ApsaraDB for RDS instances as the time goes on (data is alternately distributed between RDS_1 and RDS_2, so that the data size of the two RDS instances is balanced).



- YYYYMM_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYMM and YYYYMM_OPT:
 - YYYYMM_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data generation increases sequentially and the data volume does not differ much between the time points.
 - YYYYMM is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and months of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the PolarDB-X instances.

Scenarios

- Databases and tables need to be partitioned by year and month, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the PolarDB-X instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from month to month. For example, the number of monthly journal logs increases every month, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYMM_OPT does not support distributing data from every month in every year to a separate table shard. Instead, the number of table shards must be fixed.

- After a cycle over months (for example, a cycle exists between 2012-03 and 2013-03), data from the same month may be routed to the same database or table shard, depending on the actual number of table shards.

13.11.6.14. YYYYWEEK_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Optimizations

- Compared with YYYYWEEK, YYYYWEEK_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases. The effect is similar to [YYYYMM_OPT](#).
- YYYYWEEK_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYWEEK and YYYYWEEK_OPT:
 - YYYYWEEK_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data increases sequentially and the data volume does not differ much between time points.
 - YYYYWEEK is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and weeks of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the PolarDB-X instances.

Scenarios

- Databases and tables are partitioned by year and week, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the PolarDB-X instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from week to week. For example, the number of weekly journal logs increases every week, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYWEEK_OPT does not support distributing data from every week in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle over weeks (for example, a cycle exists between the first week of 2012 and the first week of 2013), data from the same week after a cycle may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.11.6.15. YYYYDD_OPT

Requirements

- The shard key must be of the DATE, DATETIME, or TIMESTAMP type.
- The version of the PolarDB-X instance must be 5.1.28-1320920 or later. For more information, see [View the instance version](#).

Optimizations

- Compared with YYYYDD, YYYYDD_OPT maintains the even distribution of data among ApsaraDB RDS for MySQL instances as the timeline increases. The effect is similar to [YYYYMM_OPT](#).

- YYYYYD_OPT distributes data evenly to each ApsaraDB RDS for MySQL instance, helping to maximize the performance of each ApsaraDB RDS for MySQL instance.
- How to choose between YYYYYD and YYYYYD_OPT:
 - YYYYYD_OPT can be used to distribute data evenly to each ApsaraDB RDS for MySQL instance if the time of service data generation increases sequentially and the data volume does not differ much between time points.
 - YYYYYD is applicable if the time of data generation increases randomly rather than sequentially.

Routing method

- Calculate the hash value based on the year and days of the year in the time value of the database shard key and then perform the remainder operation on the hash value based on the number of database shards. This completes route computing.
- The hash calculation based on the database and table shard key considers the data distribution among the ApsaraDB RDS for MySQL instances that connect to the PolarDB-X instances.

Scenarios

- Databases and tables need to be partitioned by year and by day, respectively.
- Data must be evenly distributed to each ApsaraDB RDS for MySQL instance that connects to the PolarDB-X instance.
- The time of the shard key increases sequentially rather than randomly, and the data volume is relatively average from day to day. For example, the number of daily journal logs increases every day, and the log data is not concentrated on the same ApsaraDB RDS for MySQL instance.

Precautions

- YYYYYD_OPT does not support distributing data from every day in every year to a separate table shard. Instead, the number of table shards must be fixed.
- After a cycle of a specific date (for example, a cycle exists between 2012-03-01 and 2013-03-01), data from the same date may be routed to the same database shard or table shard, depending on the actual number of table shards.

13.12. Automatic protection of high-risk SQL statements

In PolarDB-X, the data manipulation language (DML) statements are the same as MySQL statements.

We recommend that you include the shard key in the **SELECT** and **UPDATE** statements of PolarDB-X. The **INSERT** statement of PolarDB-X must include the shard key and a non-null key value.

By default, PolarDB-X disables full-table deletion and updating to prevent misoperation.

The following statements are prohibited by default:

- A **DELETE** statement without the **WHERE** or **LIMIT** condition
- An **UPDATE** statement without the **WHERE** or **LIMIT** condition

If you need to perform full-table deletion or update, you can temporarily skip this limit by using the following hint:

```
HINT: /*! TDDL:FORBID_EXECUTE_DML_ALL=false*/
```

Examples

- Full-table deletion is intercepted by default.

```
mysql> delete from tt;
ERR-CODE: [TDDL-4620][ERR_FORBID_EXECUTE_DML_ALL] Forbid execute DELETE ALL or UPDATE ALL sql. More: [http://
/middleware.alibaba-inc.com/faq/faqByFaqCode.html?faqCode=TDDL-4620]
```

The operation is successful if the following HINT is added:

```
mysql> /*! TDDL:FORBID_EXECUTE_DML_ALL=false*/delete from tt;
Query OK, 10 row affected (0.21 sec)
```

- Full-table update is intercepted by default.

```
mysql> update tt set id = 1;
ERR-CODE: [TDDL-4620][ERR_FORBID_EXECUTE_DML_ALL] Forbid execute DELETE ALL or UPDATE ALL sql. More: [http://
/middleware.alibaba-inc.com/faq/faqByFaqCode.html?faqCode=TDDL-4620]
```

The operation is successful if the following HINT is added:

```
mysql> /*! TDDL:FORBID_EXECUTE_DML_ALL=false*/update tt set id = 1;
Query OK, 10 row affected (0.21 sec)
```

- This limit does not apply to DELETE or UPDATE statements that contain the WHERE or LIMIT condition.

```
mysql> delete from tt where id = 1;
Query OK, 1 row affected (0.21 sec)
```

13.13. PolarDB-X sequence

13.13.1. Overview

A PolarDB-X sequence (a 64-digit number of the signed BIGINT type in MySQL) is used to create a globally unique and sequentially incremental numeric sequence, such as values of primary key columns and unique key columns.

PolarDB-X sequences are used in the following two ways:

- Explicit sequences are created and maintained by using sequence-specific data definition language (DDL) syntax and can be used independently. The sequence value can be retrieved by using `select seq.nextval;`, where `seq` indicates the sequence name.
- Implicit sequences are used to automatically fill in primary keys with `AUTO_INCREMENT` defined and are automatically maintained by PolarDB-X.

 **Notice** PolarDB-X creates implicit sequences only after `AUTO_INCREMENT` is defined for partitioned tables and broadcast tables. This is not the case for non-partition tables. The `AUTO_INCREMENT` value of a non-partition table is created by ApsaraDB RDS for MySQL.

Types and features of PolarDB-X sequences

Currently, three types of PolarDB-X sequences are supported.

Type (abbreviation)	Globally unique	Consecutive	Monotonically increasing	Monotonically increasing within the same connection	Non-single point	Date types	Readability
Group sequence (GROUP)	Yes	No	No	Yes	Yes	All integer types	High
Time-based sequence (TIME)	Yes	No	Monotonically increasing at the macro level and non-monotonically increasing at the micro level	Yes	Yes	Only BIGINT	Low
Simple sequence (SIMPLE)	Yes	Yes	Yes	Yes	No	All integer types	High

Concepts:

- **Consecutive:** If the current value is n , the next value must be $n + 1$. If the next value is not $n + 1$, it is nonconsecutive.
- **Monotonically increasing:** If the current value is n , the next value must be a number greater than n .
- **Single point:** The risk of a single point of failure (SPOF) exists.
- **Monotonically increasing at the macro level and non-monotonically increasing at the micro level:** An example of this is 1, 3, 2, 4, 5, 7, 6, 8, ... Such a sequence is monotonically increasing at the macro level and non-monotonically increasing at the micro level.

Group sequence (GROUP, used by default)

Features

A group sequence is a globally unique sequence with natural numeric values, which are not necessarily consecutive or monotonically increasing. If the sequence type is not specified, PolarDB-X uses the group sequence type by default.

- **Advantages:** A group sequence is globally unique and provides excellent performance, preventing single points of failure (SPOFs).
- **Disadvantages:** A group sequence may contain nonconsecutive values, which may not necessarily start from the initial value and do not cycle.

Implementation

The values of a group sequence are created by multiple nodes to ensure high availability. The values in a segment are nonconsecutive if the values are not all used, such as in the case of disconnection.

Time-based sequence (TIME)

Features

A time-based sequence consists of a timestamp, node ID, and serial number. It is globally unique and automatically increments at the macro level. Value updates are database-independent and not persistently stored in databases. Only names and types are stored in databases. This delivers good performance to time-based sequences, which create values like 776668092129345536, 776668098018148352, 776668111578333184, and 776668114812141568.

 **Notice** Sequence values must be of the BIGINT type when used in the auto-increment columns of tables.

- **Advantages:** Time-based sequences are globally unique with good performance.
- **Disadvantages:** The values of a time-based sequence are nonconsecutive. The START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters are invalid for time-based sequences.

Simple sequence (SIMPLE)

Features

Only simple sequences support the START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE parameters.

- **Advantages:** Simple sequences are globally unique and monotonically increasing with consecutive values.
- **Disadvantages:** Simple sequences are prone to SPOFs, poor performance, and bottlenecks. Use them with caution.

Implementation

Each sequence value must be persistently stored.

Scenarios

Group sequences, time-based sequences, and simple sequences are globally unique and can be used in primary key columns and unique index columns.

- We recommend that you use group sequences.
- Use only simple sequences for services that strongly depend on consecutive sequence values. Pay attention to sequence performance.
- We recommend that you use time-based sequences if you have high requirements for sequence performance, the amount of data inserted to tables is small, and large sequence values are acceptable. It is CPU-bound with no requirements on computing lock, database dependence, or persistent storage.

The following example shows how to create a sequence with a start value value of 100000 and a step of 1.

- A simple sequence creates globally unique, consecutive, and monotonically increasing values, such as 100000, 100001, 100002, 100003, 100004, ..., 200000, 200001, 200002, 200003... Simple sequences are persistently stored. Even after services are restarted upon an SPOF, values are still created consecutively from the breakpoint. However, simple sequences have poor performance because each value is persistently stored once it is created.
- A group sequence may create values like 200001, 200002, 200003, 200004, 100001, 100002, 100003...

Notice

- The start value of a group sequence is not necessarily the same as the START WITH value (which is 100000 in this example) but is invariably greater than this value. In this example, the start value is 200001.
- A group sequence is globally unique but may contain nonconsecutive values, for example, when a node is faulty or the connection that only uses partial values is closed. The group sequence in this example contains nonconsecutive values because the values between 200004 and 100001 are missing.

- A time-based sequence may create values like 776668092129345536, 776668098018148352, 776668111578333184, 776668114812141568...

13.13.2. Explicit sequence usage

This topic describes how to use data definition language (DDL) statements to create, modify, delete, and query sequences and how to retrieve the values of explicit sequences.

Create a sequence

Syntax:

```
CREATE [ GROUP | SIMPLE | TIME ] SEQUENCE <name>
[ START WITH <numeric value> ] [ INCREMENT BY <numeric value> ]
[ MAXVALUE <numeric value> ] [ CYCLE | NOCYCLE ]
```

Parameter description:

Parameter	Description	Applicable To
START WITH	The initial sequence value. If it is not set, the default value is 1.	Simple sequence and group sequence
INCREMENT BY	The increment (or interval value or step) of each sequence increase. If it is not set, the default value is 1.	Simple Sequence
MAXVALUE	The maximum sequence value. If it is not specified, the default value is the maximum value of the signed BIGINT type.	Simple Sequence
CYCLE or NOCYCLE	Indicates whether to repeat the sequence value which starts from the value specified by START WITH after the sequence value reaches the maximum value. If it is not specified, the default value is NOCYCLE.	Simple Sequence

Note

- If the sequence type is not specified, the group sequence type is used by default.
- INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE do not take effect for group sequences.
- START WITH, INCREMENT BY, MAXVALUE, and CYCLE or NOCYCLE do not take effect for time-based sequences.
- Group sequences are nonconsecutive. The START WITH parameter only provides reference for group sequences. The start value of a group sequence is not necessarily the same as the START WITH value but is invariably greater than this value.

Example 1: Create a group sequence.

- Method 1:

```
mysql> CREATE SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

- Method 2:

```
mysql> CREATE GROUP SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

Example 2: Create a time-based sequence.

```
mysql> CREATE TIME SEQUENCE seq1;
Query OK, 1 row affected (0.27 sec)
```

Example 3: Create a simple sequence with a start value of 1,000, a step size of 2, and a maximum value of 9999999999, which does not repeat after increasing to the maximum value.

```
mysql> CREATE SIMPLE SEQUENCE seq2 START WITH 1000 INCREMENT BY 2 MAXVALUE 9999999999 NOCYCLE;
Query OK, 1 row affected (0.03 sec)
```

Modify a sequence

PolarDB-X allows you to modify sequences in the following ways:

- For simple sequences, change the values of **START WITH**, **INCREMENT BY**, **MAXVALUE**, and **CYCLE** or **NOCYCLE**.
- For group sequences, change the value of **START WITH**.
- Convert the sequence type to another.

Syntax:

```
ALTER SEQUENCE <name> [ CHANGE TO GROUP | SIMPLE | TIME ]
START WITH <numeric value> [ INCREMENT BY <numeric value> ]
[ MAXVALUE <numeric value> ] [ CYCLE | NOCYCLE ]
```

Parameter description:

Parameter	Description	Applicable To
START WITH	The initial sequence value. If it is not set, the default value is 1.	Simple sequence and group sequence
INCREMENT BY	The increment (or interval value or step) of each sequence increase. If it is not set, the default value is 1.	Simple Sequence
MAXVALUE	The maximum sequence value. If it is not specified, the default value is the maximum value of the signed BIGINT type.	Simple Sequence
CYCLE or NOCYCLE	Indicates whether to repeat the sequence value which starts from the value specified by START WITH after the sequence value reaches the maximum value. If it is not specified, the default value is NOCYCLE .	Simple Sequence

Note

- Group sequences are nonconsecutive. The **START WITH** parameter only provides reference for group sequences. The start value of a group sequence is not necessarily the same as the **START WITH** value but is invariably greater than this value.
- If you set **START WITH** when modifying a simple sequence, the **START WITH** value takes effect immediately. The next automatically generated sequence value starts from the new **START WITH** value. For example, if you change the **START WITH** value to 200 when the sequence value increases to 100, the next automatically generated sequence value starts from 200.
- Before changing the **START WITH** value, you need to analyze the existing sequence values and the speed of creating sequence values to avoid conflicts. Exercise caution when you modify the **START WITH** value.

For example, change the start value of the simple sequence seq2 to 3000, the step size to 5, and the maximum value to 1000000. The sequence value repeats after increasing to the maximum value.

```
mysql> ALTER SEQUENCE seq2 START WITH 3000 INCREMENT BY 5 MAXVALUE 1000000 CYCLE;
Query OK, 1 row affected (0.01 sec)
```

Convert the sequence type to another.

- Use the `CHANGE TO <sequence_type>` clause of `ALTER SEQUENCE`.
- If you specify the `CHANGE TO` clause in `ALTER SEQUENCE`, the `START WITH` parameter must be added to avoid forgetting to specify the start value and get duplicate values. If the `CHANGE TO` clause is not specified, it is not required to add the `START WITH` parameter.

Example: Convert a group sequence to a simple sequence.

```
mysql> ALTER SEQUENCE seq1 CHANGE TO SIMPLE START WITH 1000000;
Query OK, 1 row affected (0.02 sec)
```

Delete a sequence

Syntax:

```
DROP SEQUENCE <name>
```

Example:

```
mysql> DROP SEQUENCE seq3;
Query OK, 1 row affected (0.02 sec)
```

Query sequences

Syntax:

```
SHOW SEQUENCES
```

Example: The TYPE column lists the abbreviations of sequence types.

```
mysql> SHOW SEQUENCES;
+-----+-----+-----+-----+-----+-----+-----+
| NAME   | VALUE      | INCREMENT_BY | START_WITH | MAX_VALUE   | CYCLE | TYPE |
+-----+-----+-----+-----+-----+-----+-----+
| AUTO_SEQ_1 | 91820513   | 1             | 91820200  | 9223372036854775807 | N     | SIMPLE |
| AUTO_SEQ_4 | 91820200   | 2             | 1000      | 9223372036854775807 | Y     | SIMPLE |
| seq_test  | N/A        | N/A           | N/A       | N/A         | N/A   | TIME   |
| AUTO_SEQ_2 | 100000     | N/A           | N/A       | N/A         | N/A   | GROUP  |
| AUTO_SEQ_3 | 200000     | N/A           | N/A       | N/A         | N/A   | GROUP  |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Retrieve a sequence value

Syntax:

```
< sequence name >.NEXTVAL
```

Example:

```
SELECT sample_seq.nextVal FROM dual;
+-----+
| SAMPLE_SEQ.NEXTVAL |
+-----+
|      101001 |
+-----+
1 row in set (0.04 sec)
```

You can also write `SAMPLE_SEQ.nextVal` as a value to the SQL statement:

```
mysql> INSERT INTO some_users (name,address,gmt_create,gmt_modified,intro) VALUES ('sun',SAMPLE_SEQ.nextVal,now(),now(),'aa');
Query OK, 1 row affected (0.01 sec)
```

 **Note** If you set the `AUTO_INCREMENT` parameter when creating a table, you do not need to specify an auto-increment column when running the `INSERT` statement. The auto-increment column is automatically maintained by PolarDB-X.

Retrieve sequence values in batches**Syntax:**

```
SELECT < sequence name >.NEXTVAL FROM DUAL WHERE COUNT = < numeric value >
```

Example:

```
SELECT sample_seq.nextVal FROM dual WHERE count = 10;
+-----+
| SAMPLE_SEQ.NEXTVAL |
+-----+
|      101002 |
|      101003 |
|      101004 |
|      101005 |
|      101006 |
|      101007 |
|      101008 |
|      101009 |
|      101010 |
|      101011 |
+-----+
10 row in set (0.04 sec)
```

13.13.3. Implicit sequence usage

After AUTO_INCREMENT is set for a primary key, the primary key is automatically filled in by using a sequence which is maintained by PolarDB-X.

CREATE TABLE

The standard CREATE TABLE syntax is extended to add the sequence type for auto-increment columns. If the type keyword is not specified, the default type is GROUP. Sequence names automatically created by PolarDB-X and associated with tables are all prefixed with AUTO_SEQ_ and followed by the table names.

```
CREATE TABLE <name> (
  <column> ... AUTO_INCREMENT [ BY GROUP | SIMPLE | TIME ],
  <column definition>,
  ...
) ... AUTO_INCREMENT=<start value>
```

SHOW CREATE TABLE

The sequence type is displayed for the auto-increment column of a table shard or broadcast table.

```
SHOW CREATE TABLE <name>
```

Examples

- If AUTO_INCREMENT is set but the sequence type is not specified when a table is created, a group sequence is used by default.

Example 1

```
mysql> CREATE TABLE `xkv_shard` (
  -> `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT 'id',
  -> `gmt_create` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT 'gmt_create',
  -> `uid` bigint(20) unsigned DEFAULT '10' COMMENT 'uid',
  -> `msg` varchar(40) DEFAULT '127.0.0.1' COMMENT 'desc',
  -> `val` float DEFAULT '0' COMMENT 'val',
  -> `time` time DEFAULT NULL COMMENT 'time',
  -> PRIMARY KEY (`id`),
  -> UNIQUE KEY `msg` (`msg`)
  -> ) ENGINE=InnoDB AUTO_INCREMENT=100009 DEFAULT CHARSET=utf8 dbpartition by hash(`id`);
Query OK, 0 rows affected (1.24 sec)

mysql> show create table xkv_shard;
+-----+-----+
| Table | Create Table |
+-----+-----+
| xkv_shard | CREATE TABLE `xkv_shard` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT BY GROUP COMMENT 'id',
  `gmt_create` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT 'gmt_create',
  `uid` bigint(20) unsigned DEFAULT '10' COMMENT 'uid',
  `msg` varchar(40) DEFAULT '127.0.0.1' COMMENT 'desc',
  `val` float DEFAULT '0' COMMENT 'val',
  `time` time DEFAULT NULL COMMENT 'time',
  PRIMARY KEY (`id`),
  UNIQUE KEY `msg` (`msg`)
) ENGINE=InnoDB AUTO_INCREMENT=100009 DEFAULT CHARSET=utf8 dbpartition by hash(`id`) |
+-----+-----+
1 row in set (0.02 sec)

mysql> drop table xkv_shard;
```

- When creating a table, set AUTO_INCREMENT and specify a time-based sequence as the primary key value.

Example 2

```
mysql> CREATE TABLE `timeseq_test` (
  -> `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT BY TIME COMMENT ' ',
  -> `gmt_create` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT ' ',
  -> `uid` bigint(20) unsigned DEFAULT '10' COMMENT 'uid',
  -> `msg` varchar(40) DEFAULT '127.0.0.1' COMMENT 'desc',
  -> `val` float DEFAULT '0' COMMENT 'val',
  -> `time` time DEFAULT NULL COMMENT 'time',
  -> PRIMARY KEY (`id`),
  -> UNIQUE KEY `msg` (`msg`)
  -> ) ENGINE=InnoDB AUTO_INCREMENT=100009 DEFAULT CHARSET=utf8 dbpartition by hash(`id`);
Query OK, 0 rows affected (1.27 sec)

mysql> show create table timeseq_test;
+-----+-----+
| Table | Create Table |
+-----+-----+
| timeseq_test | CREATE TABLE `timeseq_test` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT BY TIME COMMENT ' ',
  `gmt_create` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMENT ' ',
  `uid` bigint(20) unsigned DEFAULT '10' COMMENT 'uid',
  `msg` varchar(40) DEFAULT '127.0.0.1' COMMENT 'desc',
  `val` float DEFAULT '0' COMMENT 'val',
  `time` time DEFAULT NULL COMMENT 'time',
  PRIMARY KEY (`id`),
  UNIQUE KEY `msg` (`msg`)
) ENGINE=InnoDB AUTO_INCREMENT=100009 DEFAULT CHARSET=utf8 dbpartition by hash(`id`) |
+-----+-----+
1 row in set (0.04 sec)
```

ALTER TABLE

Currently, **ALTER TABLE** cannot be used to change the sequence type but can be used to change the initial value. To modify the implicit sequence type in a table, use the **SHOW SEQUENCES** command to find the sequence names and types, and then use the **ALTER SEQUENCE** command to modify the sequences.

```
ALTER TABLE <name> ... AUTO_INCREMENT=<start value>
```

Notice If a PolarDB-X sequence is used, exercise caution when modifying the start value of **AUTO_INCREMENT**. You need to carefully evaluate the already generated sequence values and the speed of generating new sequence values to prevent conflicts.

13.13.4. Limits and precautions for sequences

This topic describes the limits and precautions for sequences.

Limits and precautions

- When a time-based sequence is used in the auto-increment column of a table, the column must be of the **BIGINT** type.
- **START WITH** must be set when the sequence is changed to another type.
- When the **INSERT** statement is executed on a PolarDB-X database in non-partition mode where only one ApsaraDB RDS for MySQL database is bound or on a database in partition mode that has only one table but no broadcast table, PolarDB-X automatically optimizes and sends the statement, and bypasses the part of the optimizer that allocates the sequence value. In this case, **INSERT INTO ... VALUES (seq.nextval, ...)** is not supported. We recommend that you use the ApsaraDB RDS for MySQL auto-increment column feature instead.
- If the hint for a specific database shard is used by the **INSERT** statement such as **INSERT INTO ... VALUES ...** or **INSERT INTO ... SELECT...** and the target table uses a sequence, PolarDB-X bypasses the optimizer and directly sends the statement so that the sequence does not take effect. The target table creates an ID by using the auto-increment feature of the ApsaraDB RDS for MySQL table.
- The auto-increment ID allocation method for the same table must be kept consistent, which may be based on PolarDB-X sequences or the auto-increment column of the ApsaraDB RDS for MySQL. If both of the two allocation methods are used for the same table, duplicate IDs may be created and making location difficult.

Troubleshoot primary key conflicts

Assume that data is directly written to ApsaraDB RDS for MySQL and that the related primary key value is not the sequence value created by PolarDB-X. If PolarDB-X automatically creates a primary key and writes it to the database, this primary key may conflict with that of the directly written data. This problem can be resolved as follows:

1. View the existing sequences by using the PolarDB-X-specified SQL statement. The sequence prefixed with `AUTO_SEQ_` is an implicit sequence. This sequence is generated when a table is created with the `AUTO_INCREMENT` parameter.

```
mysql> SHOW SEQUENCES;
+-----+-----+-----+-----+-----+-----+-----+
| NAME          | VALUE | INCREMENT_BY | START_WITH | MAX_VALUE | CYCLE | TYPE |
+-----+-----+-----+-----+-----+-----+-----+
| AUTO_SEQ_timeseq_test | N/A | N/A | N/A | N/A | N/A | TIME |
| AUTO_SEQ_xkv_shard_tbl1 | 0 | N/A | N/A | N/A | N/A | GROUP |
| AUTO_SEQ_xkv_shard | 0 | N/A | N/A | N/A | N/A | GROUP |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

2. For example, if the `t_item` table contains conflicts and its primary key is `ID`, then retrieve the maximum primary key value of this table from PolarDB-X:

```
mysql> SELECT MAX(id) FROM t_item;
+-----+
| max(id) |
+-----+
| 8231 |
+-----+
1 row in set (0.01 sec)
```

3. Update the related value in the PolarDB-X sequence table to a value greater than 8231, such as 9000. Then, no error is returned for the auto-increment primary key created by the subsequent `INSERT` statement.

```
mysql> ALTER SEQUENCE AUTO_SEQ_USERS START WITH 9000;
Query OK, 1 row affected (0.01 sec)
```

13.14. Best practices

13.14.1. Select a shard key

A shard key is a field for database sharding and table sharding, which is used to create sharding rules during horizontal partitioning. PolarDB-X partitions a logical table horizontally into the physical database shards on each ApsaraDB RDS for MySQL instance based on the shard key.

The primary principle of sharding is to identify the business logic-specific subject of data in a table as much as possible and confirm that most (or core) database operations are performed based on this subject. Then, use the subject-related field as the shard key to perform database sharding and table sharding.

The business logic-specific subject is related to business scenarios. The following typical scenarios include business logic-specific subjects that can be used as shard keys:

- User-oriented Internet applications are operated to meet user requirements. Users are the business logic-specific subject and the user-related field can be used as the shard key.

- Seller-oriented e-commerce applications are operated to meet seller requirements. Sellers are the business logic-specific subject and the seller-related field can be used as the shard key.
- Game-oriented applications are operated to meet gamer requirements. Gamers are the business logic-specific subject and the gamer-related field can be used as the shard key.
- Internet of Vehicles (IoV) applications are operated based on vehicle information. Vehicles are the business logic-specific subject and the vehicle-related field can be used as the shard key.
- Tax-oriented applications are operated based on taxpayer information to provide frontend services. Taxpayers are the business logic-specific subject and the taxpayer-related field can be used as the shard key.

In other scenarios, you can also use the appropriate subject of business logic as the shard key.

For example, in a seller-oriented e-commerce application, the following single table must be horizontally partitioned:

```
CREATE TABLE sample_order (  
  id INT(11) NOT NULL,  
  sellerId INT(11) NOT NULL,  
  trade_id INT(11) NOT NULL,  
  buyer_id INT(11) NOT NULL,  
  buyer_nick VARCHAR(64) DEFAULT NULL,  
  PRIMARY KEY (id)  
)
```

The sellerId field is used as the shard key because seller is the business logic-specific subject. In the case of database sharding but no table sharding, the distributed data definition language (DDL) statement for table creation is as follows:

```
CREATE TABLE sample_order (  
  id INT(11) NOT NULL,  
  sellerId INT(11) NOT NULL,  
  trade_id INT(11) NOT NULL,  
  buyer_id INT(11) NOT NULL,  
  buyer_nick VARCHAR(64) DEFAULT NULL,  
  PRIMARY KEY (id)  
) DBPARTITION BY HASH(sellerId)
```

If no business logic-specific subject can be used as the shard key, use the following methods to select an appropriate shard key:

- Determine the shard key based on the distribution and access of data. Distribute the data in a table to different physical database shards and table shards as evenly as possible. This method is applicable to scenarios with massive analytical queries, in which query concurrency stays at 1.
- Determine the shard key for database sharding and table sharding by combining fields of the numeric (string) type and time type. This method is applicable to log retrieval.

For example, a log system records all user operations and needs to horizontally partition the following single log table:

```
CREATE TABLE user_log (
  userId INT(11) NOT NULL,
  name VARCHAR(64) NOT NULL,
  operation VARCHAR(128) DEFAULT NULL,
  actionDate DATE DEFAULT NULL
)
```

You can combine the user identifier with the time field to create a shard key for partitioning the table by week. The distributed DDL statement for table creation is as follows:

```
CREATE TABLE user_log (
  userId INT(11) NOT NULL,
  name VARCHAR(64) NOT NULL,
  operation VARCHAR(128) DEFAULT NULL,
  actionDate DATE DEFAULT NULL
) DBPARTITION BY HASH(userId) TBPARTITION BY WEEK(actionDate) TBPARTITIONS 7
```

For more information about shard key selection and table shard forms, see [DDL statements](#).

 Notice Avoid using hotspot data as the shard key if possible.

13.14.2. Select the number of shards

PolarDB-X supports horizontal partitioning of databases and tables. Eight physical database shards are created on each ApsaraDB RDS for MySQL instance by default, and one or more physical table shards can be created on each physical database shard. The number of table shards is also called the number of shards.

Generally, we recommend that each physical table shard contain no more than 5 million rows of data. Generally, you can estimate the data growth in one to two years. Divide the estimated total data size by the total number of physical database shards, and then divide the result by the recommended maximum data size of 5 million, to obtain the number of physical table shards to be created on each physical database shard:

$$\text{Number of physical table shards in each physical database shard} = \text{CEILING}(\text{Estimated total data size} / (\text{Number of ApsaraDB RDS for MySQL instances} \times 8) / 5,000,000)$$

Therefore, when the calculated number of physical table shards is equal to 1, only database sharding needs to be performed, that is, a physical table shard is created in each physical database shard. If the calculation result is greater than 1, we recommend that you perform both database sharding and table sharding, that is, there are multiple physical table shards in each physical database shard.

For example, if a user estimates that the total data size of a table will be about 0.1 billion rows two years later and the user has four ApsaraDB RDS for MySQL instances, then according to the preceding formula:

$$\text{Number of physical table shards in each physical database shard} = \text{CEILING}(100,000,000 / (4 \times 8) / 5,000,000) = \text{CEILING}(0.625) = 1$$

The result is 1, so only database sharding is needed, that is, one physical table shard is created in each physical database shard.

If only one ApsaraDB RDS for MySQL instance is used in the preceding example, the formula is as follows:

$$\text{Number of physical table shards in each physical database shard} = \text{CEILING}(100,000,000 / (1 \times 8) / 5,000,000) = \text{CEILING}(2.5) = 3$$

The result is 3, so we recommend that you create three physical table shards in each physical database shard.

13.14.3. Basic concepts of SQL optimization

PolarDB-X is an efficient and stable distributed relational database service that processes distributed relational computing. PolarDB-X optimizes SQL statements differently from single-instance relational databases such as MySQL. PolarDB-X focuses on the network I/O overheads in a distributed environment and pushes SQL operations down to the underlying database shards (such as ApsaraDB RDS for MySQL) for execution, thereby reducing the network I/O overheads and improving the SQL execution efficiency.

PolarDB-X provides commands for obtaining the SQL execution information to help SQL optimization, for example, EXPLAIN commands for obtaining SQL execution plans and TRACE commands for obtaining SQL execution processes and overheads. This topic describes the basic concepts and common commands related to SQL optimization in PolarDB-X.

Execution plan

An SQL execution plan (or execution plan) is a set of ordered operation steps generated to access data. In PolarDB-X, the execution plan is divided into the execution plan at the PolarDB-X layer and the execution plan at the ApsaraDB RDS for MySQL layer. Execution plan analysis is an effective way to optimize SQL statements. Through execution plan analysis, you can know whether PolarDB-X or ApsaraDB RDS for MySQL has generated optimal execution plans for SQL statements and whether further optimization can be made.

During SQL statement execution, based on the basic information of the SQL statement and related tables, the PolarDB-X optimizer determines on which database shards the SQL statement should be executed, and the specific SQL statement form, execution policy, and data merging and computing policy for each database shard. This process optimizes SQL statement execution and generates execution plans at the PolarDB-X layer. The execution plan at the ApsaraDB RDS for MySQL layer is the native MySQL execution plan.

PolarDB-X provides a set of EXPLAIN commands to display execution plans at different levels or with different levels of detail.

The following table briefly describes the EXPLAIN commands in PolarDB-X.

EXPLAIN command description

Command	Description	Example
EXPLAIN { SQL }	Displays the summary execution plan of SQL statements at the PolarDB-X layer, including the database shards on which the SQL statement is run, physical statements, and general parameters.	EXPLAIN SELECT * FROM test
EXPLAIN DETAIL { SQL }	Displays the detailed execution plans of SQL statements at the PolarDB-X layer, including the statement type, concurrency, returned field information, physical tables, and database groups.	EXPLAIN DETAIL SELECT * FROM test
EXPLAIN EXECUTE { SQL }	Displays the execution plan of the underlying ApsaraDB RDS for MySQL instance, which is equivalent to the EXPLAIN statement of MySQL.	EXPLAIN EXECUTE SELECT * FROM test

Execution plan at the PolarDB-X layer

The following table describes the fields in the results returned for the execution plan at the PolarDB-X layer.

Description of fields in execution plans at the PolarDB-X layer

Field	Description
GROUP_NAME	The name of the PolarDB-X database shard. The suffix identifies the specific database shard. The value is consistent with the result of the SHOW NODE command.
SQL	The SQL statement run on this database shard.
PARAMS	The SQL statement parameters used when PolarDB-X communicates with ApsaraDB RDS for MySQL over the Prepare protocol.

The SQL field can be in two forms:

1. If an SQL statement does not contain the following parts, the execution plan is displayed as an SQL statement:
 - Aggregate function involving multiple database shards.
 - Distributed JOIN queries involving multiple shards.
 - Complex subqueries.

Examples:

```
mysql> EXPLAIN SELECT * FROM test;
+-----+-----+-----+
| GROUP_NAME          | SQL                                | PARAMS |
+-----+-----+-----+
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0001_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0004_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0005_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0006_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
| TESTDB_1478746391548CDTCTESTDB_OXGJ_0007_RDS | select `test`.`c1`,`test`.`c2` from `test` | {} |
+-----+-----+-----+
8 rows in set (0.04 sec)
```

The group names displayed in the GROUP_NAME field can be found in the returned result of SHOW NODE:

```
mysql> SHOW NODE;
+-----+-----+-----+-----+-----+
--+
| ID | NAME                               | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_R
EAD_PERCENT |
+-----+-----+-----+-----+-----+
--+
| 0 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS |      69 |      0 | 100% | 0% |
| 1 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0001_RDS |      45 |      0 | 100% | 0% |
| 2 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0002_RDS |      30 |      0 | 100% | 0% |
| 3 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0003_RDS |      29 |      0 | 100% | 0% |
| 4 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0004_RDS |      11 |      0 | 100% | 0% |
| 5 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0005_RDS |       1 |      0 | 100% | 0% |
| 6 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0006_RDS |       8 |      0 | 100% | 0% |
| 7 | TESTDB_1478746391548CDTCTESTDB_OXGJ_0007_RDS |      50 |      0 | 100% | 0% |
+-----+-----+-----+-----+-----+
--+
8 rows in set (0.10 sec)
```

2. Execution plans that cannot be expressed by SQL statements can be expressed by PolarDB-X in custom format.

Examples:

```
mysql> EXPLAIN DETAIL SELECT COUNT(*) FROM test;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GROUP_NAME          | SQL          | PARAMS |
+-----+-----+-----+-----+-----+-----+-----+
| TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS | Merge as test
  queryConcurrency:GROUP_CONCURRENT
  columns:[count(*)]
  executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS
  Query from test as test
    queryConcurrency:SEQUENTIAL
    columns:[count(*)]
    tableName:test
    executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0000_RDS
  Query from test as test
    queryConcurrency:SEQUENTIAL
    columns:[count(*)]
    tableName:test
    executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0001_RDS
  ... ..
  Query from test as test
    queryConcurrency:SEQUENTIAL
    columns:[count(*)]
    tableName:test
    executeOn: TEST_DB_1478746391548CDTCTEST_DB_OXGJ_0007_RDS
| NULL |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

executeOn in the SQL statement field indicates the database shard on which the SQL statement is run. PolarDB-X finally merges the results returned by the database shards.

Execution plans at the ApsaraDB RDS for MySQL layer

The execution plans at the ApsaraDB RDS for MySQL layer are the same as the native MySQL execution plan. For more information, see [official MySQL documentation](#).

One PolarDB-X logical table may consist of multiple table shards distributed in different database shards. Therefore, you can view the execution plans at the ApsaraDB RDS for MySQL layer in multiple ways.

1. View the execution plan of an ApsaraDB RDS for MySQL database shard.

If the query condition contains a shard key, directly run the EXPLAIN EXECUTE command to display the execution plan on the corresponding database shard. Examples:

```
mysql> EXPLAIN EXECUTE SELECT * FROM test WHERE c1 = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | test | const | PRIMARY | PRIMARY | 4 | const | 1 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

Notice If an SQL statement involves multiple shards (for example, its condition does not contain a shard key), the EXPLAIN EXECUTE command returns an execution plan on a random ApsaraDB RDS for MySQL database shard.

To view the execution plan of an SQL statement on a specified database shard, you can add a hint.
Examples:

```
mysql> /*! TDDL:node='TESTDB_1478746391548CDTCTESTDB_OXGJ_0000_RDS'*/EXPLAIN SELECT * FROM test;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | test | ALL | NULL | NULL | NULL | NULL | 2 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

2. View the execution plans of all ApsaraDB RDS for MySQL database shards.

You can run SCAN HINT to display the execution plans of SQL statements on all database shards:

```
mysql> /*! TDDL:scan='test'*/EXPLAIN SELECT * FROM test;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | test | ALL | NULL | NULL | NULL | NULL | 2 | NULL |
| 1 | SIMPLE | test | ALL | NULL | NULL | NULL | NULL | 3 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.08 sec)
```

Notice

- i. In hint mode, PolarDB-X only replaces the table names in case of database or table sharding, and then directly sends the logical SQL statement to ApsaraDB RDS for MySQL for execution. It will not process the result.
- ii. Execution plans obtained by using an EXPLAIN command are generated by static analysis and are not actually executed in databases.

TRACE command

The TRACE command in PolarDB-X can track the SQL execution process and the overheads in each stage. It can be used together with the execution plan to facilitate SQL statement optimization.

The TRACE command contains two related commands: TRACE and SHOW TRACE, which must be used together.

13.14.4. SQL optimization methods

13.14.4.1. Overview

This topic describes the SQL optimization principles and methods for optimizing different types of SQL statements in PolarDB-X.

Basic principles of SQL optimization

In PolarDB-X, SQL computing that can be performed by ApsaraDB RDS for MySQL instances is called push-down computing. Push-down computing reduces data transmission, decreases overheads at the network layer and PolarDB-X layer, and improves the execution efficiency of SQL statements.

Therefore, the basic principle for SQL statement optimization in PolarDB-X is as follows: Push down as many computations as possible to ApsaraDB RDS for MySQL instances.

Push-down computations include:

- JOIN connections
- Filter conditions, such as `WHERE` or `HAVING` conditions
- Aggregate computing, such as `COUNT` and `GROUP BY`
- Sorting, such as `ORDER BY`
- Deduplication, such as `DISTINCT`
- Function computing, such as the `NOW()` function
- Subqueries

 **Notice** The preceding list only describes possible forms of push-down computations. It does not mean that all clauses or conditions or combinations of clauses or conditions can be pushed down for computing.

SQL statements of different types and containing different conditions can be optimized in different ways. The following uses some examples to describe how to optimize SQL statements:

- Single-table SQL optimization
 - Filter condition optimization
 - Optimization of the number of returned rows for a query
 - Grouping and sorting optimization
- JOIN query optimization
 - Optimization of push-down JOIN queries
 - Optimization of distributed JOIN queries
- Subquery optimization

13.14.4.2. Single-table SQL optimization

Single-table SQL optimization must follow the following principles:

- Make sure that the SQL statements contain the shard key.
- Use an equivalence condition for the shard key whenever possible.
- If the shard key is an IN condition, the number of values after IN should be as small as possible (far fewer than the number of shards, and remain unchanged as the business grows).
- If SQL statements do not contain a shard key, use only one of `DISTINCT`, `GROUP BY`, and `ORDER BY` in the same SQL statement.

Filter condition optimization

PolarDB-X partitions data horizontally by the shard key. Therefore, the filter condition must contain the shard key as much as possible so that PolarDB-X can push queries down to specific database shards based on the shard key values, to avoid scanning all tables in the PolarDB-X instance.

For example, the shard key of the test table is c1. If the filter condition does not contain this shard key, full table scan is performed:

```
mysql> SELECT * FROM test WHERE c2 = 2;
+----+----+
| c1 | c2 |
+----+----+
| 2 | 2 |
+----+----+
1 row in set (0.05 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c2 = 2;
+-----+-----+-----+-----+-----+-----+
| GROUP_NAME          | SQL                                          | PARAMS |
+-----+-----+-----+-----+-----+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0004_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0007_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0005_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0006_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0000_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c2` = 2) | {} |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

The smaller the value range of the filter condition that contains the shard key, the faster the PolarDB-X query speed.

For example, a query on the test table contains a range filter condition with the shard key c1:

```
mysql> SELECT * FROM test WHERE c1 > 1 AND c1 < 4;
+----+----+
| c1 | c2 |
+----+----+
| 2 | 2 |
| 3 | 3 |
+----+----+
2 rows in set (0.04 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c1 > 1 AND c1 < 4;
+-----+-----+
--+-----+
| GROUP_NAME          | SQL                                                                 | PARAMS |
+-----+-----+
--+-----+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where ((`test`.`c1` > 1) AND (`test`.`c1` < 4)) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `test`.`c1`,`test`.`c2` from `test` where ((`test`.`c1` > 1) AND (`test`.`c1` < 4)) | {} |
+-----+-----+
--+-----+
2 rows in set (0.00 sec)
```

The equivalence condition is executed faster than the range condition. Examples:

```
mysql> SELECT * FROM test WHERE c1 = 2;
+----+----+
| c1 | c2 |
+----+----+
| 2 | 2 |
+----+----+
1 row in set (0.03 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM test WHERE c1 = 2;
+-----+-----+-----+
| GROUP_NAME          | SQL                                                                 | PARAMS |
+-----+-----+-----+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `test`.`c1`,`test`.`c2` from `test` where (`test`.`c1` = 2) | {} |
+-----+-----+-----+
1 row in set (0.00 sec)
```

In addition, if you want to insert data into a table shard, the inserted field must contain a shard key.

For example, the data inserted into the test table contains the shard key c1:

```
mysql> INSERT INTO test(c1,c2) VALUES(8,8);
Query OK, 1 row affected (0.07 sec)
```

Optimization of the number of returned rows for a query

When PolarDB-X runs a query that contains `LIMIT [offset,] row_count`, PolarDB-X actually reads records before `offset` in order and directly discards them. In this way, when the value of `offset` is large, the query is slow even if the value of `row_count` is small. For example, run the following SQL statement:

```
SELECT *  
FROM sample_order  
ORDER BY sample_order.id  
LIMIT 10000, 2
```

Although only the 10000th and 10001st records are returned, it takes about 12 seconds to run the SQL statement because PolarDB-X actually reads 10,002 records.

```
mysql> SELECT * FROM sample_order ORDER BY sample_order.id LIMIT 10000,2;  
+-----+-----+-----+-----+-----+  
| id      | sellerId | trade_id | buyer_id | buyer_nick |  
+-----+-----+-----+-----+-----+  
| 242012755468 | 1711939506 | 242012755467 | 244148116334 | zhangsan |  
| 242012759093 | 1711939506 | 242012759092 | 244148138304 | wangwu  |  
+-----+-----+-----+-----+-----+  
2 rows in set (11.93 sec)
```

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT * FROM sample_order ORDER BY sample_order.id LIMIT 10000,2;
+-----+-----+-----+
| GROUP_NAME | SQL | PARAMS |
+-----+-----+-----+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0004_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0007_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0005_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0003_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0006_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0000_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `sample_order`.`id`,`sample_order`.`sellerId`,`sample_order`.`trade_id`,`sample_order`.`buyer_id`,`sample_order`.`buyer_nick` from `sample_order` order by `sample_order`.`id` asc limit 0,10002 | {} |
+-----+-----+-----+
8 rows in set (0.01 sec)
```

To optimize the preceding SQL statement, find the ID set, and use IN to match the actual records. The modified SQL query is as follows:

```
SELECT *
FROM sample_order o
WHERE o.id IN (
    SELECT id
    FROM sample_order
    ORDER BY id
    LIMIT 10000, 2 )
```

The purpose is to cache IDs in the memory first (on the premise that the number of IDs is small). If the shard key of the sample_order table is an ID, PolarDB-X can also push down such an IN query to different database shards through rule-based calculation, avoiding full table scan and unnecessary network I/O operations. Observe the effect of the modified SQL query:

```
mysql> SELECT *
-> FROM sample_order o
-> WHERE o.id IN ( SELECT id FROM sample_order ORDER BY id LIMIT 10000,2);
+-----+-----+-----+-----+-----+
| id      | sellerId | trade_id | buyer_id | buyer_nick |
+-----+-----+-----+-----+-----+
| 242012755468 | 1711939506 | 242012755467 | 244148116334 | zhangsan |
| 242012759093 | 1711939506 | 242012759092 | 244148138304 | wangwu |
+-----+-----+-----+-----+-----+
2 rows in set (1.08 sec)
```

The execution time is significantly reduced from 12 seconds to 1.08 seconds.

The corresponding execution plan is as follows:

```
mysql> EXPLAIN SELECT *
-> FROM sample_order o
-> WHERE o.id IN ( SELECT id FROM sample_order ORDER BY id LIMIT 10000,2);
+-----+-----+-----+-----+-----+
| GROUP_NAME          | SQL                                                                 | PARAMS |
+-----+-----+-----+-----+-----+
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0002_RDS | select `o`.`id`,`o`.`sellerId`,`o`.`trade_id`,`o`.`buyer_id`,`o`.`buyer_nick` from `sample_order` `o` where (`o`.`id` IN (10002)) | {} |
| SEQPERF_1478746391548CDTCSEQPERF_OXGJ_0001_RDS | select `o`.`id`,`o`.`sellerId`,`o`.`trade_id`,`o`.`buyer_id`,`o`.`buyer_nick` from `sample_order` `o` where (`o`.`id` IN (10001)) | {} |
+-----+-----+-----+-----+-----+
2 rows in set (0.03 sec)
```

Grouping and sorting optimization

In PolarDB-X, if an SQL query must use all of DISTINCT, GROUP BY, and ORDER BY, try to ensure that the fields after DISTINCT, GROUP BY, and ORDER BY are the same and the fields are shard keys. In this way, only a small amount of data is returned for the SQL query. This minimizes the network bandwidth consumed by distributed queries and removes the need to retrieve a large amount of data and sort the data in a temporary table, thereby maximizing the system performance.

13.14.4.3. JOIN query optimization

JOIN queries in PolarDB-X are classified into push-down JOIN queries and non-push-down JOIN queries (distributed JOIN queries). The optimization policies for these two types of JOIN queries are different.

Optimize push-down JOIN queries

Push-down JOIN queries are classified into the following types:

- JOIN queries between single tables (non-partition tables).
- The tables involved in the JOIN query contain the shard key in the filter condition and use the same sharding algorithm (that is, the data calculated by the sharding algorithm is distributed to the same shard).

- Tables involved in the JOIN query use the shard key as the JOIN condition and use the same sharding algorithm.
- JOIN query between broadcast tables (or small table broadcast) and table shards.

In PolarDB-X, optimize JOIN queries to push-down JOIN queries that can be executed on database shards.

Take a JOIN query between a broadcast table and table shards as an example. The broadcast table is used as the JOIN driving table (the left table in the JOIN query is called the driving table). The PolarDB-X broadcast table stores the same data in each database shard. When the broadcast table is used as the JOIN driving table, the JOIN query between this broadcast table and table shards can be converted into single-database JOIN queries and combined for computing to improve query performance.

For example, a JOIN query is performed on the following three tables, among which the sample_area table is the broadcast table, and the sample_item and sample_buyer tables are table shards. The query execution time is about 15s:

```
mysql> SELECT sample_area.name
-> FROM sample_item i JOIN sample_buyer b ON i.sellerId = b.sellerId JOIN sample_area a ON b.province = a.id
-> WHERE a.id < 110107
-> LIMIT 0, 10;
+-----+
| name |
+-----+
| BJ |
+-----+
10 rows in set (14.88 sec)
```

If you adjust the JOIN query order and move the broadcast table to the farthest left as the JOIN driving table, the JOIN query is pushed down to a single database shard in the PolarDB-X instance:

```
mysql> SELECT sample_area.name
-> FROM sample_area a JOIN sample_buyer b ON b.province = a.id JOIN sample_item i ON i.sellerId = b.sellerId
-> WHERE a.id < 110107
-> LIMIT 0, 10;
+-----+
| name |
+-----+
| BJ |
+-----+
10 rows in set (0.04 sec)
```

The query execution time decreases from 15 seconds to 0.04 seconds, which is a significant improvement to the query performance.



Notice The broadcast table achieves data consistency through the synchronization mechanism on database shards, with a latency of several seconds.

Optimize distributed JOIN queries

If a JOIN query cannot be pushed down (that is, the JOIN condition and filter condition do not contain the shard key), PolarDB-X must complete part of the computing in the query. Such a query is a distributed JOIN query.

Tables in a distributed JOIN query are classified into two types based on the data size:

- **Small table:** A table that contains a small amount of data (less than 100 data records or less data than other tables) that is involved in JOIN computing after filtering.
- **Large table:** A table that contains a large amount of data (more than 100 data records or more data than other tables) that is involved in JOIN computing after filtering.

In most cases, Nested Loop and its derived algorithms are used in JOIN computing at the PolarDB-X layer. If sorting is required for JOIN queries, the Sort Merge algorithm is used. When the Nested Loop algorithm is used, a smaller data size in the left table in a JOIN query indicates a smaller number of queries performed by PolarDB-X on the right table. If the right table has indexes or contains a small amount of data, the JOIN query is even faster. In PolarDB-X, the left table of a distributed JOIN query is called the driving table. To optimize a distributed JOIN query, use a small table as the driving table and set as many filter conditions as possible for the driving table.

Take the following distributed JOIN query as an example. The query takes about 24 seconds:

```
mysql> SELECT t.title, t.price
-> FROM sample_order o,
-> ( SELECT * FROM sample_item i WHERE i.id = 242002396687 ) t
-> WHERE t.source_id = o.source_item_id AND o.sellerId < 1733635660;
+-----+-----+
| title          | price |
+-----+-----+
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
+-----+-----+
10 rows in set (23.79 sec)
```

The preceding JOIN query is an INNER JOIN query, with the actual size of the intermediate data involved in JOIN computing unknown. In this case, perform COUNT() on the o table and t table respectively to obtain the actual data size.

For the o table, o.sellerId < 1733635660 in the WHERE condition is only related to the o table. Then, extract and add it to the COUNT() condition of the o table. The following query result is returned:

```
mysql> SELECT COUNT(*) FROM sample_order o WHERE o.sellerId < 1733635660;
+-----+
| count(*) |
+-----+
| 504018 |
+-----+
1 row in set (0.10 sec)
```

The intermediate result of the o table contains about 500,000 records. Similarly, the t table is a subquery, which can be extracted directly for the COUNT() query:

```
mysql> SELECT COUNT(*) FROM sample_item i WHERE i.id = 242002396687;
+-----+
| count(*) |
+-----+
| 1 |
+-----+
1 row in set (0.01 sec)
```

The intermediate result of the t table contains only one record. Therefore, the o table is a large table and the t table is a small table. Use the small table as the driving table of the distributed JOIN query. The result of the adjusted JOIN query is as follows:

```
mysql> SELECT t.title, t.price
-> FROM ( SELECT * FROM sample_item i WHERE i.id = 242002396687 ) t,
-> sample_order o
-> WHERE t.source_id = o.source_item_id AND o.sellerId < 1733635660;
+-----+-----+
| title          | price |
+-----+-----+
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
| Sample Item for Distributed JOIN | 239.00 |
+-----+-----+
10 rows in set (0.15 sec)
```

The query execution time decreases from about 24 seconds to 0.15 seconds, with the query performance significantly improved.

13.14.4.4. Subquery optimization

When optimizing an SQL statement that contains subqueries, push the subqueries down to database shards as much as possible to reduce the computing workload at the PolarDB-X layer.

For this purpose, you can try two optimization methods:

- Rewrite subqueries into multi-table JOIN queries, and optimize the JOIN queries.
- Use the shard key in the JOIN condition or filter condition so that PolarDB-X can push the query down to a specific database shard to avoid full table scan.

The following subquery is used as an example:

```
SELECT o.*
FROM sample_order o
WHERE NOT EXISTS
  (SELECT sellerId FROM sample_seller s WHERE o.sellerId = s.id)
```

Rewrite the subquery into a JOIN query:

```
SELECT o.*
FROM sample_order o LEFT JOIN sample_seller s ON o.sellerId = s.id
WHERE s.id IS NULL
```

13.14.5. Select connection pools for an application

A database connection pool is used to manage database connections in a centralized manner, so as to improve application performance and reduce database loads.

- **Reuse resources:** Connections can be reused to avoid high performance overheads caused by frequent connection creation and release. Resource reuse can also improve the system stability.
- **Improve the system response efficiency:** After the connection initialization is complete, all requests can directly use the existing connections, which avoids the overheads of connection initialization and release and improves the system response efficiency.
- **Avoid connection leakage:** The connection pool forcibly revokes connections based on the preset de-allocation policy to avoid connection resource leakage.

We recommend that you use a connection pool to connect applications and databases for service operations. For Java programs, we recommend that you use the [Druid connection pool](#).

The following is an example of standard Druid Spring configuration:

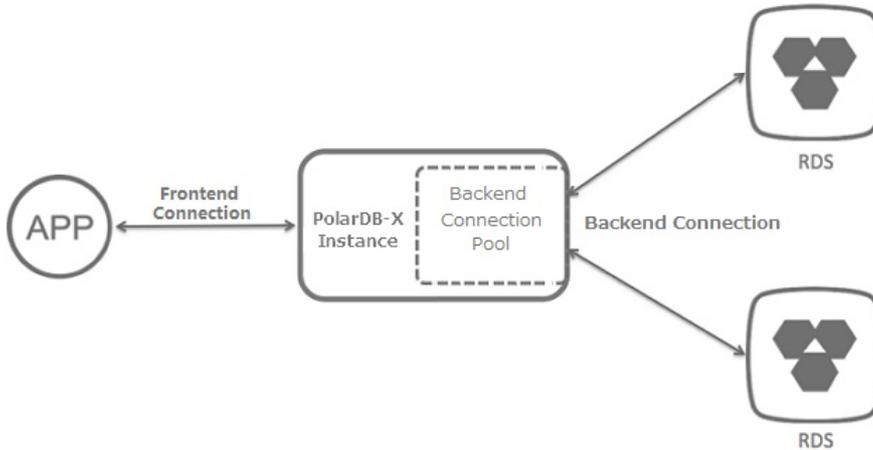
```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
  <property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <!-- Basic attributes URL, user, and password -->
  <property name="url" value="jdbc:mysql://ip:port/db? autoReconnect=true&rewriteBatchedStatements=true&socketTimeout=30000&connectTimeout=3000" />
  <property name="username" value="root" />
  <property name="password" value="123456" />
  <!-- Configure the initial size, minimum value, and maximum value -->
  <property name="maxActive" value="20" />
  <property name="initialSize" value="3" />
  <property name="minIdle" value="3" />
  <!-- maxWait indicates the time-out period for obtaining the connection -->
  <property name="maxWait" value="60000" />
  <!-- timeBetweenEvictionRunsMillis indicates the interval for detecting idle connections to be closed, in milliseconds -->
  <property name="timeBetweenEvictionRunsMillis" value="60000" />
  <!-- minEvictableIdleTimeMillis indicates the minimum idle time of a connection in the connection pool, in milliseconds -->
  <property name="minEvictableIdleTimeMillis" value="300000" />
  <!-- SQL statement used to check whether connections are available -->
  <property name="validationQuery" value="SELECT 'z'" />
  <!-- Whether to enable idle connection check -->
  <property name="testWhileIdle" value="true" />
  <!-- Whether to check the connection status before obtaining a connection -->
  <property name="testOnBorrow" value="false" />
  <!-- Whether to check the connection status before releasing a connection -->
  <property name="testOnReturn" value="false" />
</bean>
```

13.14.6. Connections to PolarDB-X instances

When an application connects to a PolarDB-X instance for operation, there are two types of connections from the perspective of the PolarDB-X instance:

- **Frontend connection:** a connection established by an application to the logical database in the PolarDB-X instance.
- **Backend connection:** a connection established by a node in a PolarDB-X instance to a physical database in a backend ApsaraDB RDS for MySQL instance.

PolarDB-X instance connection diagram



Frontend connection

Theoretically, the number of frontend connections is limited by the available memory size and the number of network connections to the nodes of the PolarDB-X instance. However, in actual application scenarios, when an application connects to a PolarDB-X instance, the nodes of the PolarDB-X instance usually manage a limited number of connections to perform requested operations, and do not maintain a large number of concurrent persistent connections (for example, tens of thousands of concurrent persistent connections). Therefore, the number of frontend connections that a PolarDB-X instance can accept can be considered to be unlimited.

Considering that the number of frontend connections is unlimited and a large number of idle connections are allowed, this method applies to scenarios where a large number of servers are deployed and need to maintain their connections to the PolarDB-X instance.

Note Although the number of frontend connections is considered as unlimited, operation requests obtained from frontend connections are actually executed by internal threads of the PolarDB-X instance through backend connections. Due to the limited number of internal threads and backend connections, the total number of concurrent requests that can be processed by the PolarDB-X instance is limited.

Backend connection

Each node of a PolarDB-X instance creates a backend connection pool to automatically manage and maintain the backend connections to the physical databases in the ApsaraDB RDS for MySQL instance. Therefore, the maximum number of connections in the backend connection pool of a PolarDB-X instance is directly related to the maximum number of connections supported by the ApsaraDB RDS for MySQL instance. You can use the following formula to calculate the maximum number of connections in the backend connection pool of a PolarDB-X instance:

$$\text{Maximum number of connections in a backend connection pool of a PolarDB-X instance} = \text{FLOOR}(\text{Maximum number of connections in an ApsaraDB RDS for MySQL instance} / \text{Number of physical database shards in the ApsaraDB RDS for MySQL instance} / \text{Number of nodes on the PolarDB-X instance})$$

For example, a user has purchased an ApsaraDB RDS for MySQL instance and a PolarDB-X instance of the following types:

- The ApsaraDB RDS for MySQL instance has eight physical database shards, four cores, and 16 GB memory, supporting a maximum number of 4,000 connections.
- The PolarDB-X dedicated instance has 32 cores and 32 GB memory, with each PolarDB-X node having two cores and 2 GB memory (that is, the instance has 16 PolarDB-X nodes).

You can use the following formula to calculate the maximum number of connections in the backend connection pool of the PolarDB-X instance:

```
Maximum number of connections in the backend connection pool of the PolarDB-X instance = FLOOR (4000/8/16) = FLOOR (31.25) = 31
```

Note

- The calculation result of the preceding formula is the maximum number of connections in the backend connection pool of the PolarDB-X instance. In actual use, to reduce the connection pressure on the ApsaraDB RDS for MySQL instance, the PolarDB-X instance adjusts the maximum number of connections in the backend connection pool to make it smaller than the upper limit.
- We recommend that you create databases in a PolarDB-X instance on a dedicated ApsaraDB RDS for MySQL instance. Do not create databases for other applications or PolarDB-X instances on the dedicated ApsaraDB RDS for MySQL instance.

Relationship between frontend and backend connections

After an application establishes frontend connections to a PolarDB-X instance and sends SQL statement execution requests, the PolarDB-X nodes process the requests asynchronously and obtain backend connections through the internal backend connection pool, and then run optimized SQL statements on one or more physical databases.

PolarDB-X nodes process requests asynchronously and frontend connections are not bound to backend connections. Therefore, a small number of backend connections can process a large number of requests for short transactions and simple queries from many concurrent frontend connections. This is why you need to focus on the queries per second (QPS) in PolarDB-X, rather than the number of concurrent connections.

Although the number of frontend connections is considered to be unlimited, the maximum number of connections maintained in the backend connection pool of a PolarDB-X instance is limited. For more information, see "Backend connections." Therefore, note the following points in actual application scenarios:

- Avoid long or large transactions in applications. These transactions occupy many or even all backend connections when they are not committed or rolled back for a long time, which reduces the overall concurrent processing capability and increases the response time (RT).
- Monitor and optimize or remove slow SQL queries run in the PolarDB-X instance, to prevent them from occupying too many backend connections. Otherwise, the PolarDB-X instance or the ApsaraDB RDS for MySQL instance is under greater processing pressure, which may lead to reduced concurrent processing capability, increased RT, or higher SQL execution failure rate due to execution timeout. For troubleshooting and optimization of slow SQL queries, see [Troubleshoot slow SQL statements in PolarDB-X](#) and [Overview](#).
- Under normal use of connections and execution of queries, if the maximum number of connections in the backend connection pool of the PolarDB-X instance is reached, contact Customer Services for assistance.

13.14.7. Perform instance upgrade

Database performance can be measured by the response time (RT) and queries per second (QPS). RT reflects the performance of a single SQL statement. This type of performance problem can be solved through SQL optimization. PolarDB-X upgrade expands the capacity to improve performance, and is suitable for database access services with low latency and high QPS.

The performance of a PolarDB-X instance depends on the performance of PolarDB-X and ApsaraDB RDS for MySQL. Insufficient performance of any PolarDB-X or ApsaraDB RDS for MySQL node can create a bottleneck in the overall performance. This topic describes how to observe the performance metrics of a PolarDB-X instance and upgrade the PolarDB-X instance to solve the performance bottleneck. For more information about how to determine the performance of an ApsaraDB RDS for MySQL instance and upgrade the ApsaraDB RDS for MySQL instance, see the ApsaraDB RDS for MySQL documentation.

Determine the performance bottleneck of a PolarDB-X instance

The QPS and CPU performance of a PolarDB-X instance are in positive correlation. When a PolarDB-X instance encounters a performance bottleneck, the CPU utilization of the PolarDB-X instance remains high.

Observe the CPU utilization

1. On the **Basic Information** page of the PolarDB-X instance, choose **Monitoring and Alerts > Instance Monitoring** from the left-side navigation pane.
2. On the **Instance Monitoring** page, select a monitoring dimension and the corresponding metrics to view details.

If the CPU utilization exceeds 90% or remains above 80%, the PolarDB-X instance faces a performance bottleneck. If there is no bottleneck for the ApsaraDB RDS for MySQL instance, the current PolarDB-X instance specifications cannot meet the QPS performance requirements of the business. In this case, the PolarDB-X instance needs to be upgraded.

For more performance-related service monitoring scenarios and methods for configuring the PolarDB-X CPU utilization alert, see [View monitoring information](#).

Upgrade PolarDB-X

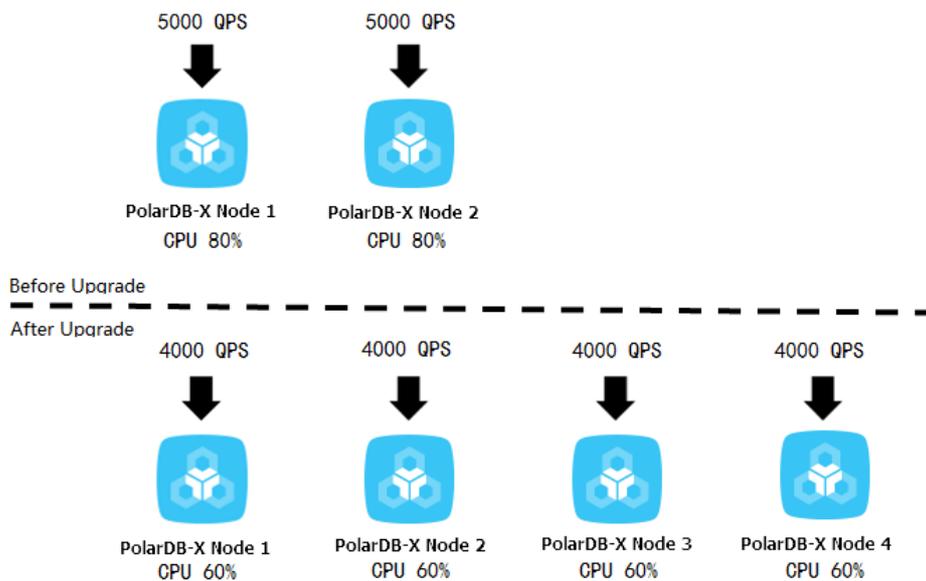
QPS is an important metric for determining whether the PolarDB-X instance specifications can meet the business requirements. Each type of instance specifications corresponds to a reference QPS value.

Some special SQL statements require more computing (such as temporary table sorting and aggregate computing) in PolarDB-X. In this case, the QPS supported by each PolarDB-X instance is lower than the standard value in its type.

PolarDB-X upgrade improves the processing performance of a PolarDB-X instance by adding nodes to share the QPS. As PolarDB-X nodes are stateless, this upgrade method linearly improves the performance of PolarDB-X instances.

For example, service A requires QPS of about 15 thousand. The current PolarDB-X instance has a 4-core virtual CPU (vCPU), 4 GB memory, and two nodes, supporting QPS of only 10 thousand. After finding that the CPU utilization of the PolarDB-X instance remains high, we upgraded the instance to 8-core vCPU and 8 GB memory, with each node handling about 4,000 QPS. Then, the performance meets service requirements, and the CPU utilization also drops to a reasonable level, as shown in the following figure.

PolarDB-X upgrade

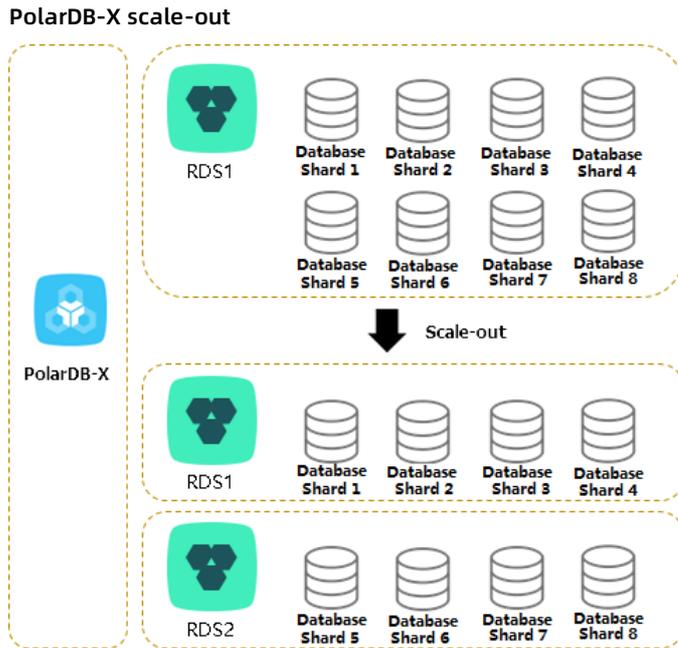


For more information about how to upgrade a PolarDB-X instance, see [Change specifications](#).

13.14.8. Perform scale-out

In PolarDB-X, smooth scale-out improves the overall performance by increasing the number of ApsaraDB RDS for MySQL instances. You can increase the number of ApsaraDB RDS for MySQL instances through PolarDB-X smooth scale-out to increase the PolarDB-X database capacity when the following conditions are met: 1. The input/output operations per second (IOPS), CPU utilization, disk space, and other metrics of the ApsaraDB RDS for MySQL instance reach their bottlenecks. 2. The bottlenecks cannot be removed through SQL optimization or ApsaraDB RDS for MySQL upgrade (for example, the disk has been upgraded to the top configuration).

PolarDB-X smooth scale-out reduces the pressure on the original ApsaraDB RDS for MySQL instance by migrating database shards to the new ApsaraDB RDS for MySQL instance. For example, before scale-out, all the eight databases are deployed in one ApsaraDB RDS for MySQL instance. After scale-out, the eight databases are deployed in two ApsaraDB RDS for MySQL instances, and the pressure on a single ApsaraDB RDS for MySQL instance is significantly reduced, as shown in the following figure.



After multiple scale-out operations, if the number of ApsaraDB RDS for MySQL instances is equal to the number of database shards, you need to create another PolarDB-X instance and ApsaraDB RDS for MySQL databases with the expected capacity, and then migrate data to further increase the data capacity. This process is complex. We recommend that you consider the data growth expected in the next two to three years and plan the number of ApsaraDB RDS for MySQL instances properly when creating a PolarDB-X database.

Determine whether scale-out is required

You can determine whether PolarDB-X smooth scale-out is required based on three ApsaraDB RDS for MySQL metrics: IOPS, CPU utilization, and disk space. You can view these metrics in the ApsaraDB RDS for MySQL console. For more information, see the ApsaraDB RDS for MySQL documentation.

IOPS and CPU utilization

If you find that the IOPS or CPU utilization remains above 80% for a long time or you frequently receive alerts, follow these steps:

1. Optimize SQL statements. Generally, you can solve the high CPU utilization problem by this method.
2. If the problem persists, upgrade the ApsaraDB RDS for MySQL instance. For more information, see the ApsaraDB RDS for MySQL documentation.
3. When the CPU utilization or IOPS exceeds the threshold, you can set read-only databases to share the load on the primary database. However, read/write splitting affects read consistency. For more information, see the [Read/write splitting](#) documentation.
4. If the problem persists, scale out the PolarDB-X instance.

Disk space

ApsaraDB RDS for MySQL has the following types of disk space:

1. **Data space:** the space occupied by data. The space usage continues increasing as more data is inserted. We recommend that you keep the remaining disk space above 30%.
2. **System file space:** the space occupied by shared tables and error log files.
3. **Binary log file space:** the space occupied by binary logs generated during database operation. The more update transactions there are, the larger the occupied space is.

Whether scale-out is required depends on the data space. When the data space is about to or expected to exceed the disk capacity, you can distribute the data to the databases on multiple ApsaraDB RDS for MySQL instances through scale-out.

Scale-out risks and precautions

PolarDB-X scale-out consists of four steps: **configuration > migration > switchover > cleanup**. For more information, see the [Perform smooth scale-out](#) documentation.

Note the following points before scale-out:

- To reduce the pressure of read operations on the source ApsaraDB RDS for MySQL instance, perform scale-out when the load on the source ApsaraDB RDS for MySQL instance is low.
- During scale-out, do not submit data definition language (DDL) tasks in the console or connect to the PolarDB-X instance to directly run DDL SQL statements. Otherwise, the scale-out task may fail.
- Scale-out requires that the source database table have a primary key. If the source database does not have a primary key, add one first.
- During scale-out, the read and write traffic is switched to the new ApsaraDB RDS for MySQL instance. The switchover process takes three to five minutes. We recommend that you perform a switchover during off-peak hours.
- Scale-out does not affect the PolarDB-X instance before the switchover. Therefore, you can cancel the scale-out through rollback before the switchover.
- Scale-out creates pressure on databases. We recommend that you perform this operation during off-peak hours.

13.14.9. Troubleshoot slow SQL statements in PolarDB-X

13.14.9.1. Details about a slow SQL statement

PolarDB-X defines an SQL statement that takes more than 1 second to run as a slow SQL statement. Slow SQL statements in PolarDB-X are classified into slow logical SQL statements and slow physical SQL statements. In PolarDB-X, an SQL statement is run step by step on PolarDB-X and ApsaraDB RDS for MySQL nodes. Large execution loss on any node will result in slow SQL statements.

- Slow logical SQL statements are slow SQL statements sent by an application to PolarDB-X.
- Slow physical SQL statements are slow SQL statements sent by PolarDB-X to ApsaraDB RDS for MySQL.

Syntax

```
SHOW FULL {SLOW | PHYSICAL_SLOW} [WHERE where_condition]
      [ORDER BY col_name [ASC | DESC], ...]
      [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

Description

The `SHOW FULL SLOW` command shows slow logical SQL statements, that is, SQL statements sent by an application to PolarDB-X.

The result set of the `SHOW FULL SLOW` command contains the following columns:

- **TRACE_ID:** the unique identifier of the SQL statement. A logical SQL statement and the physical SQL

statements generated by the logical SQL statement have the same TRACE_ID. The TRACE_ID is also sent as a comment to ApsaraDB RDS for MySQL.

- **HOST**: the IP address of the client that sends the SQL statement.

 **Notice** The client IP address may not be obtained when the network type is Virtual Private Cloud (VPC).

- **START_TIME**: the time when PolarDB-X starts running the SQL statement.
- **EXECUTE_TIME**: the time consumed by PolarDB-X to run the SQL statement.
- **AFFECT_ROW**: the number of records returned or the number of rows affected by the SQL statement.
- **SQL**: the statement that is run.

The `SHOW FULL PHYSICAL_SLOW` command shows the slow physical SQL statements, that is, SQL statements sent by PolarDB-X to ApsaraDB RDS for MySQL.

The result set of `SHOW FULL PHYSICAL_SLOW` contains the following columns:

- **TRACE_ID**: the unique identifier of the SQL statement. A logical SQL statement and the physical SQL statements generated by the logical SQL statement have the same TRACE_ID. The TRACE_ID is also sent as a comment to ApsaraDB RDS for MySQL.
- **GROUP_NAME**: the name of a database group. Grouping aims to manage multiple groups of databases with identical data, such as the primary and secondary databases after data replication through ApsaraDB RDS for MySQL, which are mainly used for read/write splitting and primary/secondary switchover.
- **DBKEY_NAME**: the name of the database shard on which the SQL statement is run.
- **START_TIME**: the time when PolarDB-X starts running the SQL statement.
- **EXECUTE_TIME**: the time consumed by PolarDB-X to run the SQL statement.
- **SQL_EXECUTE_TIME**: the time consumed by PolarDB-X to call ApsaraDB RDS for MySQL to run this SQL statement.
- **GETLOCK_CONNECTION_TIME**: the time that PolarDB-X takes to get connections from the connection pool. If the value is large, the ApsaraDB RDS for MySQL connections have been exhausted. This is typically due to a large number of slow SQL statements. You can log on to the corresponding ApsaraDB RDS for MySQL instance and run `SHOW PROCESSLIST` for troubleshooting.
- **CREATE_CONNECTION_TIME**: the time consumed by PolarDB-X to establish a connection to ApsaraDB RDS for MySQL. If the value is large, it is largely because the ApsaraDB RDS for MySQL instance is overloaded or faulty.
- **AFFECT_ROW**: the number of records returned or the number of rows affected by the SQL statement.
- **SQL**: the statement that is run.

Example 1

The following example describes how to locate the execution of a slow SQL statement on PolarDB-X and between PolarDB-X and ApsaraDB RDS for MySQL.

1. You can use certain conditions, such as the execution time and SQL string match, to obtain the specified slow SQL statement:

```
mysql> show full slow where `SQL` like '%select sleep(50)%';
+-----+-----+-----+-----+-----+-----+
| TRACE_ID | HOST | START_TIME | EXECUTE_TIME | AFFECT_ROW | SQL |
+-----+-----+-----+-----+-----+-----+
| ae0e565b8c00000 | 127.0.0.1 | 2017-03-29 19:28:43.028 | 50009 | 1 | select sleep(50) |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

2. Based on the TRACE_ID of the slow logical SQL statement, run `SHOW FULL PHYSICAL_SLOW` to obtain the physical execution information of this SQL statement.

```
mysql> show full physical_slow where trace_id = 'ae0e565b8c00000';
+-----+-----+-----+-----+-----+
| TRACE_ID | GROUP_NAME | DBKEY_NAME | START_TIME | EXEC
UTE_TIME | SQL_EXECUTE_TIME | GETLOCK_CONNECTION_TIME | CREATE_CONNECTION_TIME | AFFECT_ROW | SQL
|
+-----+-----+-----+-----+-----+
| ae0e565b8c00000 | PRIV_TEST_1489167306631PJAFPRIV_TEST_APKK_0000_RDS | rdso6g5b6206sdq832ow_priv_test_a
pkk_0000_nfup | 2017-03-29 19:27:53.02 | 50001 | 50001 | 0 | 0 | 1 | select slee
p(50) |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

3. In the SQL statement details and slow SQL statement records of the ApsaraDB RDS for MySQL instance, you can query the execution information of this SQL statement on the ApsaraDB RDS for MySQL instance based on TRACE_ID.

Slow query logs

Example 2

This example describes how to locate the original SQL statement in PolarDB-X based on the slow SQL statement located in ApsaraDB RDS for MySQL.

1. Based on the slow SQL query log in ApsaraDB RDS for MySQL, TRACE_ID of the slow SQL statement is ae0e55660c00000.
2. Based on the TRACE_ID obtained in Step 1, run `SHOW FULL PHYSICAL_SLOW` to obtain the physical execution information of this SQL statement.

```
mysql> show full physical_slow where trace_id = 'ae0e55660c00000';
+-----+-----+-----+-----+-----+
| TRACE_ID | GROUP_NAME | DBKEY_NAME | START_TIME | EXEC
UTE_TIME | SQL_EXECUTE_TIME | GETLOCK_CONNECTION_TIME | CREATE_CONNECTION_TIME | AFFECT_ROW | SQL
|
+-----+-----+-----+-----+-----+
| ae0e55660c00000 | PRIV_TEST_1489167306631PJAFPRIV_TEST_APKK_0000_RDS | rdso6g5b6206sdq832ow_priv_test_a
pkk_0000_nfup | 2017-03-29 19:27:37.308 | 10003 | 10001 | 0 | 0 | 1 | select sle
ep(10) |
+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

13.14.9.2. Locate slow SQL statements

Generally, you can locate a slow SQL statement in two ways: Obtain historical information about slow SQL statements from slow SQL statement records, or run `SHOW PROCESSLIST` to display the real-time execution information about slow SQL statements.

You can troubleshoot slow SQL statements as follows:

1. Locate slow SQL statements.
2. Locate nodes with performance loss.
3. Troubleshoot the performance loss.

 **Note** During troubleshooting, we recommend that you use the MySQL command line `mysql -hIP -PPORT -uUSER -pPASSWORD -c` to create the connection. Be sure to add `-c` to prevent the MySQL client from filtering out the comments (default operation) and therefore affecting the execution of HINT.

- View slow SQL statement records

Run the following command to query top 10 slow SQL statements. This command can query logical SQL statements in PolarDB-X. One logical SQL statement corresponds to SQL statements of one or more databases or tables of the ApsaraDB RDS for MySQL instance. For more information, see [Details about a low SQL statement](#).

```
mysql> SHOW SLOW limit 10;
+-----+-----+-----+-----+-----+-----+
| TRACE_ID | HOST | START_TIME | EXECUTE_TIME | AFFECT_ROW | SQL |
|-----+-----+-----+-----+-----+-----+
| ac3133132801001 | xx.xxx.xx.97 | 2017-03-06 15:48:32.330 | 900392 | -1 | select detail_url, sum(price) from t_item group by detail_url; |
.....
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

- View real-time SQL execution information

If the execution of an SQL statement is slow in the current server, run `SHOW PROCESSLIST` to view the real-time SQL execution information in the current PolarDB-X database. The value in the TIME column indicates how long the current SQL statement has been run.

```
mysql> SHOW PROCESSLIST WHERE COMMAND != 'Sleep';
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
----+-----+
| ID      | USER      | DB          | COMMAND | TIME | STATE | DB          | INFO |
| ROWS_SENT | ROWS_EXAMINED | ROWS_READ |         |      |       |            |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
| 0-0-352724126 | ifisibhk0 | test_123_wvvp_0000 | Query | 13 | Sending data | NULL | /*
DRDS /42.120.74.88/ac47e5a72801000/ */select `t_item`.`detail_url`,SUM(`t_item`.`price`) from `t_i | NULL | NU
LL | NULL |
| 0-0-352864311 | cowxhthg0 | NULL | Binlog Dump | 17 | Master has sent all binlog to slave; waiting for binl
og to be updated | NULL | NULL | NULL | NULL |
| 0-0-402714795 | ifisibhk0 | test_123_wvvp_0005 | Alter | 114 | Sending data | NULL | /*D
RDS /42.120.74.88/ac47e5a72801000/ */ALTER TABLE `Persons` ADD `Birthday` date | NULL | NULL |
NULL |
.....
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
----+-----+
12 rows in set (0.03 sec)
```

The following describes each column:

- **ID:** the ID of the connection.
- **USER:** the user name of the database shard in which this SQL statement is run.
- **DB:** the specified database. If no database is specified, the value is NULL.
- **COMMAND:** the type of the command being executed. SLEEP indicates an idle connection. For more information about other commands, see [MySQL thread information documentation](#).
- **TIME:** the elapsed execution time of the SQL statement, in seconds.
- **STATE:** the current execution status. For more information, see [MySQL thread status documentation](#).
- **INFO:** the SQL statement being executed. The SQL statement may be too long to be displayed completely. You can derive the complete SQL statement based on information such as service parameters.

In the current example, the following slow SQL statement is identified:

```
ALTER TABLE `Persons` ADD `Birthday` date
```

13.14.9.3. Locate nodes with performance loss

When you locate a slow SQL statement in slow SQL statement records or real-time SQL execution information, you can run the TRACE command to trace the running time of the SQL statement in PolarDB-X and ApsaraDB RDS for MySQL to locate the bottleneck.

The TRACE command actually runs the SQL statement, records the time consumed on all nodes, and returns the execution result. For more information about TRACE and other control commands, see [Help statements](#).

 **Note** The PolarDB-X TRACE command needs to maintain the context information of the connection. Some GUI clients may use connection pools, which results in command exceptions. Therefore, we recommend that you use the MySQL command line to run the TRACE command.

Run the following command for the identified slow SQL statement:

```
mysql> trace select detail_url, sum(distinct price) from t_item group by detail_url;
+-----+-----+
| detail_url | sum(price) |
+-----+-----+
| www.xxx.com | 1084326800.00 |
| www.xx1.com | 1084326800.00 |
| www.xx2.com | 1084326800.00 |
| www.xx3.com | 1084326800.00 |
| www.xx4.com | 1084326800.00 |
| www.xx5.com | 1084326800.00 |
.....
+-----+-----+
1 row in set (7 min 2.72 sec)
```

After the TRACE command is run, run SHOW TRACE to view the result. You can identify the bottleneck of the slow SQL statement based on the time consumption of each component.

```
mysql> SHOW TRACE;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | TIMESTAMP | TYPE      | GROUP_NAME                | DBKEY_NAME                | TIME_COST
(MS) | CONNECTION_TIME_COST(MS) | ROWS | STATEMENT
| PARAMS |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | 0.000 | Optimize  | DRDS                      | DRDS                      | 2      | 0.00
| 0 | select detail_url, sum(price) from t_item group by detail_url | NULL |
| 1 | 423507.342 | Merge Sorted | DRDS                      | DRDS                      | 411307 | 0.00
| 8 | Using Merge Sorted, Order By (`t_item`.`detail_url` asc) | NULL |
| 2 | 2.378 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0003_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0003_hbpz | 15     | 1.59      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 3 | 2.731 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0000_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0000_hbpz | 11     | 1.78      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 4 | 2.933 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0004_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0004_hbpz | 15     | 1.48      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 5 | 3.111 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0001_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0001_hbpz | 15     | 1.56      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 6 | 3.323 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0007_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0007_hbpz | 15     | 1.54      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 7 | 3.496 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0006_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0006_hbpz | 18     | 1.30      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 8 | 3.505 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0005_hbpz | 423507 | 1.97      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`
) from `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 9 | 3.686 | Query     | TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0002_RDS | rds06g5b6206sdq832o
w_test_123_wvvp_0002_hbpz | 14     | 1.47      | 1 | select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) f
rom `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc | NULL |
| 10 | 423807.906 | Aggregate  | DRDS                      | DRDS                      | 1413   | 0.00
| 1 | Aggregate Function (SUM(`t_item`.`price`)), Group By (`t_item`.`detail_url` asc)
| NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.01 sec)
```

In the returned results of SHOW TRACE, you can determine which node has a long execution time based on the values (in milliseconds) in the TIME_COST column. You can also see the corresponding GROUP_NAME (that is, the PolarDB-X or ApsaraDB RDS for MySQL node) and the STATEMENT column information (that is, the SQL statement being executed). By checking whether the value of GROUP_NAME is PolarDB-X, you can determine whether the slow node exists in PolarDB-X or ApsaraDB RDS for MySQL.

According to the preceding results, the Merge Sorted action on the PolarDB-X node and the TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS node of ApsaraDB RDS for MySQL take a lot of time.

13.14.9.4. Troubleshoot the performance loss

Slow nodes may exist on the PolarDB-X or ApsaraDB RDS for MySQL instance. Troubleshoot the fault accordingly after the cause is determined.

Solution for slow PolarDB-X nodes

When the GROUP_NAME of a slow node is in the PolarDB-X instance, check whether time-consuming computing operations such as Merge Sorted, Temp Table Merge, and Aggregate exist during SQL statement execution. If so, rectify it. For more information, see [Overview](#).

Solution for slow ApsaraDB RDS for MySQL nodes

When the slow node is on the ApsaraDB RDS for MySQL instance, check the execution plan of this SQL statement on the ApsaraDB RDS for MySQL instance.

In PolarDB-X, you can run `/*! TDDL:node={GROUP_NAME}*/ EXPLAIN` to check the SQL execution plan of an ApsaraDB RDS for MySQL instance. The execution plan displays the SQL execution process information, including inter-table association and index information.

The detailed process is as follows:

1. Based on GROUP_NAME, assemble the HINT: `/*! TDDL:node='TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS'*/`.
2. Combine the assembled HINT and the statement prefixed by EXPLAIN to form a new SQL statement and run it. The EXPLAIN command does not actually run. It only displays the execution plan of the SQL statement.

The following example describes how to query the execution plan of the identified slow node.

```
mysql> /*! TDDL:node='TEST_123_1488766060743ACTJSANGUAN_TEST_123_WVVP_0005_RDS'*/ EXPLAIN select `t_item`.`detail_url`,SUM(distinct `t_item`.`price`) from `t_item` group by `t_item`.`detail_url` order by `t_item`.`detail_url` asc;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | t_item | ALL | NULL | NULL | NULL | NULL | 1322263 | Using temporary; Using filesort |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

When the preceding SQL statement is run in ApsaraDB RDS for MySQL, the message `Using temporary; Using filesort` is returned. It indicates that low SQL statement execution is caused by improper use of the index. In this case, you can correct the index and run the SQL statement again.

13.14.10. Handle DDL exceptions

When you run any data definition language (DDL) commands of PolarDB-X, PolarDB-X performs the corresponding DDL operation on all table shards.

Failures can be divided into two types:

1. A DDL statement fails to be executed in a database shard. DDL execution failure in any database shard may result in inconsistent table shard structures.
2. The system does not respond for a long time after a DDL statement is executed. When you perform a DDL statement on a large table, the system may make no response for a long time due to the long execution time of the DDL statement in a database shard.

Execution failures in database shards may occur for various reasons. For example, the table you want to create already exists, the column you want to add already exists, or the disk space is insufficient.

No response for a long time is generally caused by the long execution time of a DDL statement in a database shard. Taking ApsaraDB RDS for MySQL as an example, the DDL execution time depends mostly on whether the operation is an in-place (directly modifying the source table) or copy (copying data in the table) operation. An in-place operation only requires modification of metadata, while a copy operation reconstructs the whole table and also involves log and buffer operations.

To determine whether a DDL operation is an in-place or copy operation, you can view the returned value of "rows affected" after the operation is completed.

Example:

- Change the default value of a column (this operation is very fast and does not affect the table data at all):

```
Query OK, 0 rows affected (0.07 sec)
```

- Add an index (this operation takes some time, but "0 rows affected" indicates that the table data is not replicated):

```
Query OK, 0 rows affected (21.42 sec)
```

- Change the data type of column (this operation takes a long time and reconstructs all data rows in the table):

```
Query OK, 1671168 rows affected (1 min 35.54 sec)
```

Therefore, before executing a DDL operation on a large table, perform the following steps to determine whether the operation is a fast or slow operation:

1. Copy the table structure to generate a cloned table.
2. Insert some data.
3. Perform the DDL operation on the cloned table.
4. Check whether the value of "rows affected" is 0 after the operation is completed. A non-zero value means that this operation reconstructs the entire table. In this case, you need to perform this operation in off-peak hours.

Solution for failures

PolarDB-X DDL operations distribute all SQL statements to all database shards for parallel execution. Execution failure on any database shard does not affect the execution on other database shards. In addition, PolarDB-X provides the CHECK TABLE command to check the structure consistency of the table shards. Therefore, failed DDL operations can be performed again, and errors reported on database shards on which the operations have been executed do not affect the execution on other database shards. Make sure that all table shards ultimately have the same structure.

Procedure for handling DDL operation failures

1. Run the CHECK TABLE command to check the table structure. If the returned result contains only one row and the status is normal, the table statuses are consistent. In this case, go to Step 2. Otherwise, go to Step 3.
2. Run the SHOW CREATE TABLE command to check the table structure. If the displayed table structure is the same as the expected structure after the DDL statement is run, the DDL statement is run. Otherwise, go to Step 3.
3. Run the SHOW PROCESSLIST command to check the statuses of all SQL statements being executed. If any

ongoing DDL operations are detected, wait until these operations are completed, and then perform Steps 1 and 2 to check the table structure. Otherwise, go to Step 4.

4. Perform the DDL operation again on PolarDB-X. If the Lock conflict error is reported, go to Step 5. Otherwise, go to Step 3.
5. Run the **RELEASE DBLOCK** command to release the DDL operation lock, and then go to Step 4.

The procedure is as follows:

1. Check the table structure consistency

Run the **CHECK TABLE** command to check the table structure. When the returned result contains only one row and the displayed status is OK, the table structures are consistent.

 **Notice** If no result is returned after you run **CHECK TABLE**, retry by using the CLI.

```
mysql> check table `xxxx`;
+-----+-----+-----+-----+
| TABLE      | OP | MSG_TYPE | MSG_TEXT |
+-----+-----+-----+-----+
| TDDL5_APP.xxxx | check | status | OK |
+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

2. Check the table structure

Run the **SHOW CREATE TABLE** command to check the table structure. If table structures are consistent and correct, the DDL statement has been run.

```
mysql> show create table `xxxx`;
+-----+-----+-----+-----+
| Table | Create Table |
+-----+-----+-----+-----+
| xxxx | CREATE TABLE `xxxx` (
`id` int(11) NOT NULL DEFAULT '0',
`NAME` varchar(1024) NOT NULL DEFAULT "",
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(`id`) tpartition by hash(`id`) tpartitions 3
|
+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

3. Check the SQL statements being executed.

If some DDL statement executions are slow and no response is received for a long time, you can run the **SHOW PROCESSLIST** command to check the status of all SQL statements being executed.

```
mysql> SHOW PROCESSLIST WHERE COMMAND != 'Sleep';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
| ID      | USER  | DB      | COMMAND | TIME | STATE | INFO
| ROWS_SENT | ROWS_EXAMINED | ROWS_READ |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
| 0-0-352724126 | ifisibhk0 | test_123_wvvp_0000 | Query | 15 | Sending data | /
*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */select `t_item`.`detail_url`,SUM(`t_item`.`price`) from `t_i | NULL |
NULL | NULL |
| 0-0-352864311 | cowxhthg0 | NULL | Binlog Dump | 13 | Master has sent all binlog to slave; waiting for bi
nlog to be updated | NULL | NULL | NULL | NULL |
| 0-0-402714566 | ifisibhk0 | test_123_wvvp_0005 | Query | 14 | Sending data | /
*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */select `t_item`.`detail_url`,`t_item`.`price` from `t_i | NULL | NU
LL | NULL |
| 0-0-402714795 | ifisibhk0 | test_123_wvvp_0005 | Alter | 114 | Sending data | /
*DRDS /xx.xxx.xx.88/ac47e5a72801000/ */ALTER TABLE `Persons` ADD `Birthday` date | NULL | N
ULL | NULL |
.....
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
+-----+-----+
12 rows in set (0.03 sec)
```

The value in the TIME column indicates the number of seconds that the command has been executed. If a command execution is too slow, as shown in the figure, you can run the KILL '0-0-402714795' command to cancel the slow command.

 **Notice** In PolarDB-X, one logical SQL statement corresponds to multiple statements on database shards. Therefore, you may need to kill multiple commands to stop a logical DDL statement. You can determine the logical SQL statement to which a command belongs based on the INFO column in the SHOW PROCESSLIST result set.

4. Handle the lock conflict error

PolarDB-X adds a database lock before performing a DDL operation and releases the lock after the operation. The KILL DDL operation may not release the lock. If you perform the DDL operation again, the following error message will be returned:

```
Lock conflict , maybe last DDL is still running
```

In this case, run **RELEASE DBLOCK** to release the lock. After the command is canceled and the lock is released, run the DDL statement again during off-peak hours or when the service is stopped.

Other problems

Clients cannot display the modified table structures.

To enable some clients to obtain table structures from system tables (such as COLUMNS or TABLES), PolarDB-X creates a shadow database in database shard 0 on your ApsaraDB RDS for MySQL instance. The shadow database name must be the same as the name of your PolarDB-X logical database. It stores all table structures and other information in the user database.

The client obtains the PolarDB-X table structure from the system table of the shadow database. During the processing of DDL exceptions, the table structure may be modified normally in the user database but not in the shadow database due to some reasons. In this case, you need to connect to the shadow database and perform the DDL operation on the table again in the database.

 **Notice** The CHECK TABLE command does not check whether the table structure in the shadow database is consistent with that in the user database.

13.14.11. Efficiently scan PolarDB-X data

PolarDB-X supports efficient data scanning and uses aggregate functions for statistical summary during full table scan.

The following describes common scanning scenarios:

- **Scan of tables without database or table shards:** PolarDB-X transmits the original SQL statement to the backend ApsaraDB RDS for MySQL database for execution. In this case, PolarDB-X supports any aggregate functions.
- **Non-full table scan:** PolarDB-X transmits the original SQL statement to each single ApsaraDB RDS for MySQL database for execution. For example, when the shard key in the WHERE clause is Equal, non-full table scan is performed. In this case, PolarDB-X also supports any aggregate functions.
- **Full table scan:** Currently, the supported aggregate functions are COUNT, MAX, MIN, and SUM. In addition, LIKE, ORDER BY, LIMIT, and GROUP BY are also supported during full table scan.
- **Parallel scan of all table shards:** If you need to export data from all databases, you can run the SHOW command to view the table topology and scan all table shards in parallel. For more information, see the following section.

Traverse tables by using a hint

1. Run the SHOW TOPOLOGY FROM TABLE_NAME command to obtain the table topology.

```
mysql:> SHOW TOPOLOGY FROM DRDS_USERS;
+----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+----+-----+-----+
| 0 | DRDS_00_RDS | drds_users |
| 1 | DRDS_01_RDS | drds_users |
+----+-----+-----+
2 rows in set (0.06 sec)
```

By default, the non-partition table is stored in database shard 0.

2. Traverse each table for TOPOLOGY.
 - i. Run the current SQL statement in database shard 0.

```
/*! TDDL:node='DRDS_00_RDS'*/ SELECT * FROM DRDS_USERS;
```

- ii. Run the current SQL statement in database shard 1.

```
/*! TDDL:node='DRDS_01_RDS'*/ SELECT * FROM DRDS_USERS;
```

 **Notice** We recommend that you run `SHOW TOPOLOGY FROM TABLE_NAME` to obtain the latest table topology before each scan.

Scan data in parallel

PolarDB-X allows you to run `mysqldump` to export data. However, if you want to scan data faster, you can establish multiple sessions for each table shard to scan tables in parallel.

```
mysql> SHOW TOPOLOGY FROM LJLTEST;
+-----+-----+-----+
| ID | GROUP_NAME | TABLE_NAME |
+-----+-----+-----+
| 0 | TDDL5_00_GROUP | ljlttest_00 |
| 1 | TDDL5_00_GROUP | ljlttest_01 |
| 2 | TDDL5_00_GROUP | ljlttest_02 |
| 3 | TDDL5_01_GROUP | ljlttest_03 |
| 4 | TDDL5_01_GROUP | ljlttest_04 |
| 5 | TDDL5_01_GROUP | ljlttest_05 |
| 6 | TDDL5_02_GROUP | ljlttest_06 |
| 7 | TDDL5_02_GROUP | ljlttest_07 |
| 8 | TDDL5_02_GROUP | ljlttest_08 |
| 9 | TDDL5_03_GROUP | ljlttest_09 |
| 10 | TDDL5_03_GROUP | ljlttest_10 |
| 11 | TDDL5_03_GROUP | ljlttest_11 |
+-----+-----+-----+
12 rows in set (0.06 sec)
```

As shown above, the table has four database shards, and each database shard has three table shards. Run the following SQL statement to operate on the table shards of the `TDDL5_00_GROUP` database:

```
#!/ TDDL:node='TDDL5_00_GROUP'*/ select * from ljlttest_00;
```

 **Note** `TDDL5_00_GROUP` in HINT corresponds to the `GROUP_NAME` column in the execution results of the `SHOW TOPOLOGY` command. In addition, the table name in the SQL statement is the table shard name.

At this time, you can establish up to 12 sessions (corresponding to 12 table shards respectively) to process data in parallel.

13.15. Appendix: PolarDB-X terms

This topic lists common terms of PolarDB-X for your reference.

Term	Description	Remarks
Cloud Native Distributed Database PolarDB-X	PolarDB-X is a distributed database service that was independently developed by Alibaba to solve the bottlenecks of single-instance database services. PolarDB-X is compatible with MySQL protocols and syntax. It supports automatic sharding, smooth scale-out, auto scaling, and transparent read/write splitting, and provides O&M capabilities for distributed databases throughout their entire lifecycle.	-

Term	Description	Remarks
TDDL	Taobao Distributed Data Layer (TDDL) was developed by Alibaba and has become a preferred component for nearly 1,000 core applications of Alibaba.	-
PolarDB-X Console	PolarDB-X Console is designed for database administrators (DBAs) to isolate resources as required and perform operations, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring data display, and IP address whitelist.	-
PolarDB-X Manager	PolarDB-X Manager is designed for global O&M personnel and DBAs to manage all PolarDB-X resources and monitor the system.	-
PolarDB-X Server	PolarDB-X Server is the service layer of PolarDB-X. Multiple server nodes make up a server cluster to provide distributed database services, including the read/write splitting, routed SQL execution, result merging, dynamic database configuration, and globally unique ID (GUID).	-
Load balancer	PolarDB-X server nodes are stateless, and therefore requests can be randomly routed to any PolarDB-X server node. The load balancer is used to complete this task. Server Load Balancer (SLB) is used for overall output by Apsara Stack. VIPServer is typically used for Alibaba middleware output.	-
Diamond	Diamond manages the configuration and storage of PolarDB-X. It provides the configuration functions for storage, query, and notification. In PolarDB-X, Diamond stores the source data of databases, and configuration data including the sharding rules, and switches.	-
Data Replication System	Data Replication System migrates and synchronizes data for PolarDB-X. Its core capabilities include full data migration and incremental data synchronization. Its derived features include smooth data import, smooth scale-out, and global secondary index. Data Replication System requires the support of ZooKeeper and PolarDB-X Rtools.	-
PolarDB-X instance (PolarDB-X instance)	A PolarDB-X instance consists of multiple PolarDB-X server nodes. A PolarDB-X instance can contain multiple PolarDB-X databases.	-
PolarDB-X instance ID (PolarDB-X instance ID)	An instance ID uniquely identifies an PolarDB-X instance.	-
Number of nodes on a PolarDB-X instance	The number of PolarDB-X server nodes in a PolarDB-X instance.	-
VIP	The virtual IP addresses (VIPs) of the load balancer can be classified as: <ul style="list-style-type: none"> 1. Public VIP, which is accessible from the Internet. It is used for testing. 2. Private VIP, which is accessible only from the Alibaba Cloud internal network. 	-
VPC	Virtual Private Cloud (VPC) is generally used on Alibaba Cloud.	-
Region	A region is a geographical location, such as East China. This concept is generally used for Alibaba Cloud.	-

Term	Description	Remarks
Azone	A physical area with independent power grids and networks within one region, such as Hangzhou Zone A. This concept is generally used for Alibaba Cloud.	-
Logical SQL statement	A logical SQL statement is an SQL statement sent from an application to PolarDB-X.	-
Physical SQL statement	A physical SQL statement is an SQL statement obtained after PolarDB-X parses a logical SQL statement and sends it to ApsaraDB RDS for MySQL for execution.	Logical SQL statements and physical SQL statements may be the same or different. Logical and physical SQL statements may be in a one-to-one or one-to-many mapping.
QPS	The queries per second (QPS) is the average number of logical SQL statements executed by PolarDB-X per second in a statistical period,	instead of the number of transactions. Most control statements, such as COMMIT and SET, are not counted in QPS.
RT	The response time (RT) is the average response time (in milliseconds) of logical SQL statements executed by PolarDB-X in a statistical period. The RT of an SQL statement is calculated as follows: (Time when PolarDB-X writes the last packet of the result set) - (Time when PolarDB-X receives the SQL statement)	-
Physical QPS	The physical QPS is the average number of physical SQL statements that PolarDB-X executes on ApsaraDB RDS for MySQL per second in a statistical period.	-
Physical RT	The physical RT is the average response time (in milliseconds) of physical SQL statements executed by PolarDB-X on ApsaraDB RDS for MySQL in a statistical period. The RT of a physical SQL statement is calculated as follows: (Time when PolarDB-X receives the result set returned by ApsaraDB RDS for MySQL) - (Time when PolarDB-X starts to obtain the connection to ApsaraDB RDS for MySQL)	This includes the time of establishing a connection to ApsaraDB RDS for MySQL or obtaining a connection from the connection pool, the network transmission time, and the time of executing the SQL statement by ApsaraDB RDS for MySQL.

Term	Description	Remarks
Connections	The number of connections established between the application and PolarDB-X,	instead of the number of connections established between PolarDB-X and ApsaraDB RDS for MySQL.
Inbound traffic	The network traffic generated when the application sends SQL statements to PolarDB-X.	This traffic is irrelevant to the traffic used for interaction between PolarDB-X and ApsaraDB RDS for MySQL.
Outbound traffic	The network traffic generated when PolarDB-X sends the result set to the application.	This traffic is irrelevant to the traffic used for interaction between PolarDB-X and ApsaraDB RDS for MySQL.
Number of active threads (ThreadRunning)	The number of threads running on a PolarDB-X instance. This parameter can be used to indicate the load of the PolarDB-X instance.	-
Global	The total monitoring data of all databases on a PolarDB-X instance.	-
Memory usage	The Java Virtual Machine (JVM) memory usage of a PolarDB-X server process.	-
Total memory usage	The memory usage of the machine where the PolarDB-X server node is located.	This metric is available only when PolarDB-X servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.
CPU utilization	The CPU utilization of the machine where a PolarDB-X server node is located.	This metric is available only when PolarDB-X servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.
System load	The load of the machine where a PolarDB-X server node is located.	This metric is available only when PolarDB-X servers are deployed on ECS instances. Generally, this metric is used for Alibaba Cloud.

Term	Description	Remarks
Service port	The port used by PolarDB-X servers to provide MySQL-based services to external applications.	Generally, the port number is 3306. However, when multiple PolarDB-X nodes (mostly physical machines) are deployed on one machine, the port number will change accordingly.
Management port	The port used by PolarDB-X servers to provide management application program interfaces (APIs).	Generally, the port number is the service port number plus 100.
Start time	The time when PolarDB-X servers start.	-
Running time	The continuous running time of the PolarDB-X servers since the last startup time.	-
Total memory size	The maximum JVM memory size of a PolarDB-X server node.	-
Memory usage	The JVM memory that is already used by the PolarDB-X server nodes.	-
Number of nodes	Required. The number of machines. A PolarDB-X instance is essentially a PolarDB-X cluster, and the number of nodes refers to the number of machines in the cluster.	-
Instance type	Required. The type of the instance, including dedicated and shared instances. A dedicated instance works in the exclusive mode. A shared instance works in the multi-tenant mode, which is generally used in Alibaba Cloud.	-
Machine type	Required. The type of the machine where a PolarDB-X server node is deployed. Valid values are Auto-selected, PHY, and ECS. The PolarDB-X inventory is divided into physical machine inventory and virtual machine inventory according to the type of machines where the PolarDB-X servers are deployed. The two types cannot be mixed because their deployment and O&M methods are different.	-
AliUid	Required. The UID of the instance. In Apsara Stack, this ID is provided by the account system in the deployment environment.	-
Backend port	The backend port of the VIP. For a PolarDB-X server node, this port is the service port of machine where the PolarDB-X server node is deployed.	-
Frontend port	The frontend port of the VIP for user access. Each VIP has a set of frontend ports and backend ports. The VIP forwards data from frontend ports to backend ports.	-
Private network/Internet	The network type of the VIP. Valid values: <ul style="list-style-type: none"> Internet: the public VIP, which is accessible from the Internet. Private network: the private VIP (including VPC VIP), which is accessible from private networks. 	-
lbld	The ID of an SLB instance, which is the unique ID of VIP. A VIP is managed based on this ID.	-

Term	Description	Remarks
Forwarding mode	<p>The port forwarding mode of the VIP. The following modes are supported:</p> <ul style="list-style-type: none"> • FNAT: This mode is recommended when the backend machine is a virtual machine or VPC needs to be supported. • NAT: This mode can be selected when the backend machine is a physical machine. Currently, this mode is only used on Alibaba Cloud. • Open FNAT: This mode is applicable only to Alibaba Cloud. 	-
VPC ID	The ID of the destination VPC, that is, the VPC to be accessed.	-
VSwitch ID	The ID of the destination VSwitch, which determines the CIDR block where the VPC VIP of the instance is in.	-
APPName	The app name of the destination PolarDB-X database. Each PolarDB-X database has a corresponding app name for loading configurations.	-
UserName	The user name used to log on to the destination PolarDB-X database.	-
DBName	The name of the destination PolarDB-X database you want to log on to.	-
IP address whitelist	Only the IP addresses specified in the IP address whitelist can access the PolarDB-X instance.	-
Read-only instance	<p>ApsaraDB RDS for MySQL instances where physical databases reside are divided into the following two types based on whether data can be written into the instances:</p> <ul style="list-style-type: none"> • Primary instance: Both read and write requests are allowed on such an instance. In Apsara Stack, ApsaraDB RDS for MySQL is supported. In Alibaba Cloud, ApsaraDB for RDS is supported. • Read-only instance: Only read requests are allowed on such an instance. In Apsara Stack, ApsaraDB RDS for MySQL is supported. In Alibaba Cloud, ApsaraDB for RDS is supported. 	-
Read SQL statement	A type of SQL statements used to read data, such as the SELECT statement. PolarDB-X determines whether an SQL statement is a read-only SQL statement when it is not in a transaction. If the SQL statement is in a transaction, PolarDB-X treats it as a write SQL statement during read/write splitting.	-
Read/write splitting	If read-only ApsaraDB RDS for MySQL instances exist, you can configure in the PolarDB-X console to allocate read SQL statements to the primary and read-only instances proportionally. PolarDB-X automatically identifies the type of SQL statements and allocates them proportionally.	-
Smooth scale-out	On the basis of horizontal partitioning, the data distribution on ApsaraDB RDS for MySQL instances is dynamically adjusted for scale-out. Generally, scale-out is completed asynchronously without any modification to the business code.	-
Broadcast of small tables	You can synchronize the data in a single table in a database to all database shards in advance, to convert the cross-database JOIN query into a JOIN query that can be completed on physical databases.	-

Term	Description	Remarks
Horizontal partitioning	Horizontal partitioning distributes the data rows originally stored in one table to multiple tables based on specified rules to achieve horizontal linear scaling.	-
Partition mode	This mode allows you to create multiple database shards on an ApsaraDB RDS for MySQL instance. These database shards make up a PolarDB-X database. In this mode, all PolarDB-X functions can be used.	-
Non-partition mode	In this mode, a database that has been created on an ApsaraDB RDS for MySQL instance is used as a PolarDB-X database. In this mode, only PolarDB-X read/write splitting is allowed, while other PolarDB-X features such as database sharding and table sharding are not allowed.	-
Imported database	An existing database on the ApsaraDB RDS for MySQL instance selected for creating a PolarDB-X database. This is a unique concept for the creation of a PolarDB-X database.	-
Read policy	The ratio of read SQL statements assigned by PolarDB-X to the primary and read-only ApsaraDB RDS for MySQL instances.	-
Full table scan	If no shard field is specified in a SQL statement, PolarDB-X runs the SQL statement on all table shards and summarizes the results. You can disable this function because of its high overheads.	-
Shard key	A column in a logical table. PolarDB-X routes data and SQL statements to a physical table based on this column.	-
Data import	The operation of importing data from an existing ApsaraDB RDS for MySQL instance to a PolarDB-X database.	-
Full data migration	The operation of migrating all existing records from a database to PolarDB-X. An offset is recorded before full migration starts.	-
Offset	In a MySQL binary log file, each row represents a data change operation. The position of a line in the binary log file is called an offset.	-
Incremental data migration	The operation of reading all MySQL binary log records from the recorded offset, converting them into SQL statements, and then running them in PolarDB-X. Incremental migration continues before the switchover.	-
Switchover	A step of data import and smooth scale-out, which writes all the remaining incremental records from MySQL binary logs to PolarDB-X.	-
Cleanup	The last step of smooth scale-out, which cleans redundant data and configurations generated during smooth scale-out.	-
Heterogeneous indexing	For table shards of a PolarDB-X database, the WHERE condition of a SQL statement for query must contain the shard key whenever possible. In this way, PolarDB-X routes the query request to a specific database shard, improving the query efficiency. If the WHERE condition of the SQL statement does not contain the shard key, PolarDB-X performs a full table scan. PolarDB-X provides heterogeneous indexing to solve this problem. The data in a database shard or table shard of a PolarDB-X instance is fully or partially synchronized to another table based on different shard keys. The destination table to which the data is synchronized is called a heterogeneous index table.	-

Term	Description	Remarks
PolarDB-X sequence	A PolarDB-X sequence (a 64-digit number of the BIGINT data type in MySQL) aims to ensure that the data (for example, PRIMARY KEY and UNIQUE KEY) in the defined unique field is globally unique and in ordered increments.	-
PolarDB-X hint	To facilitate PolarDB-X usage, PolarDB-X defines some hints to specify special actions.	-

14. AnalyticDB for MySQL

14.1. What is AnalyticDB for MySQL?

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service that is developed by Alibaba Cloud to analyze large amounts of data at high concurrency. AnalyticDB for MySQL can analyze hundreds of billions of data records across multiple dimensions within milliseconds and provide you with data-driven insights into your business.

Note OLAP systems are often compared with online transaction processing (OLTP) systems. OLAP systems are ideal for systems that require complex multidimensional queries and analytics on large amounts of data. The OLAP model is commonly adopted in analytical databases. OLTP systems are suitable for transactional processing, and ensures strong atomicity and consistency in data manipulation. The OLTP model supports frequent INSERT and UPDATE operations and is often used for relational database management systems, such as MySQL and Microsoft SQL Server.

AnalyticDB for MySQL is an RT-OLAP system that offers the following benefits:

- Compatible with MySQL, business intelligence (BI) tools, and extract, transform, and load (ETL) tools for easy, cost-effective, and efficient analysis and integration of data.
- Uses relational models to store data and provides SQL statements to flexibly compute and analyze data. You do not need to create a data model in advance.
- Uses distributed computing technologies to provide excellent real-time computing capabilities.

When AnalyticDB for MySQL processes tens of billions of data records or more, its performance can match or even surpass that of multidimensional online analytical processing (MOLAP) systems. AnalyticDB for MySQL can compute tens of billions of data records within several hundred milliseconds. You can then explore large amounts of data without constraints, instead of viewing data reports based on a predefined logic.

- Computes hundreds of billions of data records in real time.

AnalyticDB for MySQL uses all data generated in your business system for data analysis, rather than sampling a portion of the data. This maximizes the effectiveness of analysis results.

- Supports a large number of concurrent queries and ensures high system availability through dynamic multi-copy storage and computing technology. Therefore, AnalyticDB for MySQL can serve as a backend system for various products, including user-facing and enterprise-facing products.

AnalyticDB for MySQL is used in Internet business systems that have hundreds of thousands to tens of millions of users, such as Data Cube, Taobao Index, Kuaidi Dache, Alimama DMP, and Taobao Groceries.

AnalyticDB for MySQL is a real-time computing system that provides rapid and flexible online data analysis and computation.

14.2. Limits

Take note of the following limits before you use AnalyticDB for MySQL.

Object	Naming convention	Limit
Database name	A database name can be up to 64 characters in length, and can contain letters, digits, and underscores (_). It must start with a lowercase letter and cannot contain consecutive underscores (_).	Do not use analyticdb as the database name. The name analyticdb is reserved for a built-in database.

Object	Naming convention	Limit
Table name	A table name must be 1 to 127 characters in length, and can contain letters, digits, and underscores (_). It must start with a letter or underscore (_).	<ul style="list-style-type: none"> A table name cannot contain single quotation marks ('), double quotation marks ("), exclamation points (!), or spaces. A table name cannot be SQL reserved keywords.
Column name	A column name must be 1 to 127 characters in length, and can contain letters, digits, and underscores (_). It must start with a letter or underscore (_).	<ul style="list-style-type: none"> A column name cannot contain single quotation marks ('), double quotation marks ("), exclamation points (!), or spaces. A column name cannot be SQL reserved keywords.
Account name	An account name must be 2 to 16 characters in length, and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.	None
Password	A password must be 8 to 32 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @ # \$ % ^ & * () _ + - =	None
Comment in a table	None	A table comment can be up to 1,024 characters in length.
Comment in a column	None	A column comment can be up to 1,024 characters in length.
Index name	None	An index name can be up to 64 characters in length.
Default value of a column	None	The default value of a column can be up to 127 characters in length.

14.3. Quick start

14.3.1. Log on to the AnalyticDB for MySQL console

This topic describes how to log on to the AnalyticDB for MySQL console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

- In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.

2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click Login to go to the ASCM console homepage.

4. In the top navigation bar, choose Products > Database Services > AnalyticDB for MySQL.

14.3.2. Create a database cluster

This topic describes how to create an AnalyticDB for MySQL cluster.

Procedure

1. [Log on to the AnalyticDB for MySQL console.](#)
2. Click **Create Cluster** in the upper-right corner of the Clusters page and configure the parameters.

Parameter	Description
Region	The region where the cluster resides. You cannot change the region after the cluster is created. We recommend that you select a region that is closest to the geographic area of your business to improve access speed and stability.
Zone	The zone where the new cluster resides. A zone is an independent physical location within a region. There are no substantial differences between different zones within the same region.
Organization	The organization to which the cluster belongs.
Resource Set	The resource set of the cluster.
Version	Only version 3.0 is supported.
Edition	Only Basic is supported.
Network Type	AnalyticDB for MySQL supports two types of networks. <ul style="list-style-type: none"> ○ <i>VPC</i>: A VPC helps you build an isolated network environment in Apsara Stack. You can customize the route table, CIDR blocks, and gateway of a VPC. We recommend that you select VPC for higher security. ○ <i>Classic Network</i>: Cloud services in the classic network are not isolated. Access control to cloud services in a classic network is implemented by the security groups or whitelist policies of the services.
Specifications	The ECU specifications.

Parameter	Description
Node Groups	The number of node groups. By default, each node group consists of three replicas.
Storage	The storage space of a node group.

3. After you configure the preceding parameters, click **Submit**.

14.3.3. Configure a whitelist

This topic describes how to configure a whitelist for a cluster in the AnalyticDB for MySQL console.

Context

After you create an AnalyticDB for MySQL cluster, you must configure a whitelist for the cluster to allow external devices to access the cluster. The default whitelist contains only the default IP address 127.0.0.1, which indicates that no devices are allowed to access the cluster. The whitelist can enhance access security of AnalyticDB for MySQL clusters. We recommend that you maintain the whitelist on a regular basis. The whitelist does not affect the normal operation of AnalyticDB for MySQL clusters.

Procedure

1. [Log on to the AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to configure a whitelist.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Edit** to the right of the default whitelist.

 **Note** You can also click **Create Whitelist** to create a whitelist.

6. In the **Edit Whitelist** dialog box, delete the default IP address 127.0.0.1, enter the IP addresses or CIDR blocks that you want to allow, and then click **OK**.
 - o The CIDR block 0.0.0.0/0 indicates that all IP addresses are allowed to access the cluster. Exercise caution when you add this CIDR block.
 - o If you enter a CIDR block such as 10.10.10.0/24, any IP addresses in the 10.10.10.X format can access the cluster.
 - o If you want to add multiple IP addresses or CIDR blocks, separate multiple entries with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
 - o The whitelist modification takes effect within one minute.

14.3.4. Create a database account

This topic describes the database account types in AnalyticDB for MySQL and how to create database accounts.

Account types

AnalyticDB for MySQL supports two types of database accounts: privileged accounts and standard accounts.

Account types

Account type	Description
--------------	-------------

Account type	Description
Privileged account	<ul style="list-style-type: none"> You can create and manage privileged accounts only in the console. You can create only one privileged account for a cluster. You can use the privileged account to manage all standard accounts and databases of the cluster. You can use the privileged account of a cluster to disconnect any standard account of the cluster from AnalyticDB for MySQL. A privileged account has more permissions, which allows you to perform fine-grained management operations. For example, you can grant query permissions on different tables to different users. The privileged account in AnalyticDB for MySQL is equivalent to the root account in MySQL.
Standard account	<ul style="list-style-type: none"> You can use only SQL statements to create and manage standard accounts. You can create up to 256 standard accounts for a cluster. You must manually grant a standard account the permissions to access a specific database. You cannot use a standard account to disconnect other accounts from AnalyticDB for MySQL.

Create a privileged account

1. [Log on to the AnalyticDB for MySQL console](#)
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to create a privileged account.
4. In the left-side navigation pane, click **Accounts**. On the **Accounts** page, click **Create Account**.
5. In the **Create Account** pane, configure the following parameters.

Parameter	Description
Account	<p>The name of the privileged account.</p> <p>The name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.</p>
Account Type	<p>The type of the account. The value is Privileged Account. This value cannot be changed.</p>
Password	<p>The password of the privileged account.</p> <p>The password must be 8 to 32 characters in length and contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @ # \$ % ^ & * () _ + - =</p>

Parameter	Description
Confirm Password	Enter the password of the privileged account again.
Description	Optional. The description of the privileged account.

6. Click **OK** to create the privileged account.

Create a standard account

For more information about how to create a standard account and grant permissions to the standard account, see [CREATE USER](#).

What to do next

[Create a database](#)

14.3.5. Create a database

You can use a MySQL client (such as Navicat for MySQL, DBeaver, DbVisualizer, or SQL Workbench/J), application code, or the MySQL command-line tool to connect to an AnalyticDB for MySQL cluster, and execute the CREATE DATABASE statement to create a database.

Prerequisites

You have connected to a database cluster. For more information, see [Connect to a database cluster](#).

Syntax

```
CREATE DATABASE [IF NOT EXISTS] db_name;
```

Example

You can execute the following statement to create a database named adb_demo:

```
create database adb_demo;
```

14.3.6. Connect to a database cluster

This topic describes how to connect to an AnalyticDB for MySQL cluster.

Prerequisites

After you configure a whitelist and create a database account, you can use a MySQL client (such as Navicat for MySQL, DBeaver, DbVisualizer, or SQL Workbench/J), or the MySQL command-line tool to connect to the AnalyticDB for MySQL cluster. For more information, see [Configure a whitelist](#) and [Create a database account](#). You can also configure the connection information such as the endpoint, port, and account of an AnalyticDB for MySQL cluster in an application to connect to the cluster.

Use the code in an application to connect to AnalyticDB for MySQL

- [C#](#)
- [PHP](#)
- [Python](#)
- [Configure the Druid connection pool](#)
- [Java](#)

Use the MySQL command-line tool to connect to AnalyticDB for MySQL

Use the [MySQL command-line tool](#) to connect to AnalyticDB for MySQL

Use a client to connect to AnalyticDB for MySQL

- [SQL Workbench/J](#)
- [DbVisualizer](#)
- [DBeaver](#)
- [Navicat](#)

14.3.7. Apply for a public endpoint

This topic describes how to use the AnalyticDB for MySQL console to apply for a public endpoint for a cluster.

Context

- You must manually apply for a public endpoint. You can release the public endpoint if you do not need it.
- The public endpoint applies to the following scenarios:
 - You want to access an AnalyticDB for MySQL cluster on devices that are not deployed in the Alibaba Cloud environment.
 - You want to access an AnalyticDB for MySQL cluster from a region or of a network type that is different from the cluster.

Apply for a public endpoint

1. [Log on to the AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to apply for a public endpoint.
4. In the **Network Information** section of the **Cluster Information** page, click **Apply for Public Endpoint**.
5. In the **Apply for Public Endpoint** message, click **OK** to obtain a public endpoint.

 **Note** Before you use the public endpoint to access the AnalyticDB for MySQL cluster, you must add the IP address of the device that you use to access the cluster to the whitelist. For more information, see [Configure a whitelist](#).

Release a public endpoint

1. [Log on to the AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to release a public endpoint.
4. On the **Cluster Information** page, click **Release Public Endpoint**.
5. In the **Release Public Endpoint** message, click **OK** to release the public endpoint.

14.3.8. Synchronize data

To synchronize data in different scenarios, you can use the following data loading solutions provided by AnalyticDB for MySQL:

- Use Kettle to synchronize data from relational databases, NoSQL databases such as HBase, or Microsoft Office Excel or Access to AnalyticDB for MySQL.
- Use external tables to import data from OSS to AnalyticDB for MySQL or export data from AnalyticDB for MySQL to OSS.
- Use the Load Data statement to write local data to AnalyticDB for MySQL.
- If the data already exists in AnalyticDB for MySQL tables, use the `INSERT INTO...SELECT FROM` statement to

synchronize data.

14.4. Connect to a database cluster

14.4.1. Use the MySQL command-line tool to connect to AnalyticDB for MySQL

This topic describes how to use the MySQL command-line tool to connect to an AnalyticDB for MySQL cluster.

Syntax

```
mysql -hadb_url -P3306 -uadb_user -padb_password
```

Parameters

- `adb_url` : the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can obtain the endpoint of the cluster in the **Network Information** section of the **Cluster Information** page in the **AnalyticDB for MySQL console**.
- `3306` : the port number of the AnalyticDB for MySQL cluster endpoint. The default port number is `3306` .
- `adb_user` : the account used to connect to the AnalyticDB for MySQL cluster. The account can be a privileged account or a standard account that has the required permissions.
- `adb_password` : the password of the account used to connect to the AnalyticDB for MySQL cluster.

Example

```
mysql -ham-bp****.ads.aliyuncs.com -P3306 -utest -pTest123
```

14.4.2. Use the code in a business system to connect to AnalyticDB for MySQL

14.4.2.1. C#

This topic describes how to use the MySQL Connector/NET connector to connect to an AnalyticDB for MySQL cluster.

```

using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
namespace adbdemo
{
    public class Tutorial2
    {
        public static void Main()
        {
            // server: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can obtain the endpoint on the Cluster Information page of the AnalyticDB for MySQL console.
            // UID: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the following types of accounts: privileged accounts and standard accounts.
            // database: the name of the database in the AnalyticDB for MySQL cluster.
            // port: the port number of the AnalyticDB for MySQL cluster endpoint.
            // password: the password of the account used to connect to the AnalyticDB for MySQL cluster.
            string connStr = "server=...;UID=...;database=...;port=...;password=...;SslMode=none;";
            MySqlConnection conn = new MySqlConnection(connStr);
            try
            {
                Console.WriteLine("Connecting to MySQL...");
                conn.Open();
                string sql = "select c_custkey, c_name from customer limit 1";
                MySqlCommand cmd = new MySqlCommand(sql, conn);
                MySqlDataReader rdr = cmd.ExecuteReader();
                while (rdr.Read())
                {
                    Console.WriteLine(rdr[0] + " --- " + rdr[1]);
                }
                rdr.Close();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
            }
            conn.Close();
            Console.WriteLine("Done.");
        }
    }
}

```

14.4.2.2. PHP

This topic describes how to connect to an AnalyticDB for MySQL cluster in PHP.

Precautions

- If your operating system is Linux, you must install the php-mysql 5.1.x module.
- If your operating system is Windows, you must install the php_MySQL.dll library.
- For more information about how to enable PreparedStatement if you use PDO to connect to AnalyticDB for MySQL, see Enable PreparedStatement for a client in different programming languages.

Use MySQLi to connect to AnalyticDB for MySQL

```
// am-bp***.ads.aliyuncs.com: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can
obtain the endpoint on the Cluster Information page of the AnalyticDB for MySQL console.
$ads_server_name="am-bp***.ads.aliyuncs.com";
// account_name: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the fo
llowing types of accounts: privileged accounts and standard accounts.
$ads_username="account_name";
// account_password: the password of the account used to connect to the AnalyticDB for MySQL cluster.
$ads_password="account_password";
// db_name: the name of the database in the AnalyticDB for MySQL cluster.
$ads_database="db_name";
// 3306: the port number of the AnalyticDB for MySQL cluster endpoint.
$ads_port=3306;
// Connect to the AnalyticDB for MySQL cluster.
$ads_conn=mysqli_connect($ads_server_name,$ads_username,$ads_password,$ads_database, $ads_port);
```

```
$strsql="SELECT user_id FROM my_ads_db.my_first_table limit 20;";
$result=mysqli_query($ads_conn, $strsql);
while($row = mysqli_fetch_array($result)) {
// user_id: the name of the column to be queried.
echo $row["user_id"] ;
}
```

Use PDO to connect to AnalyticDB for MySQL

```
// am-bp***.ads.aliyuncs.com: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can
obtain the endpoint on the Cluster Information page of the AnalyticDB for MySQL console.
$ads_server_name = "am-bp***.ads.aliyuncs.com";
// account_name: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the fo
llowing types of accounts: privileged accounts and standard accounts.
$ads_username = "account_name";
// account_password: the password of the account used to connect to the AnalyticDB for MySQL cluster.
$ads_password = "account_password";
// db_name: the name of the database in the AnalyticDB for MySQL cluster.
$ads_database = 'db_name';
// 3306: the port number of the AnalyticDB for MySQL cluster endpoint.
$ads_port = 3306;
$dsn = "mysql:host={$ads_server_name};dbname={$ads_database};port={$ads_port}";
try {
    $dbh = new PDO($dsn, $ads_username, $ads_password);
    echo 'PDO Success !';
} catch (PDOException $e) {
    echo 'PDO Connection failed: '. $e->getCode() ."\n" . $e->getMessage() ."\n". $e->getTraceAsString();
}
}
```

14.4.2.3. Python

This topic describes how to use Python MySQLdb to connect to an AnalyticDB for MySQL cluster.

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import MySQLdb
# Create a database connection.
# host: the endpoint or IP address of the AnalyticDB for MySQL cluster to which you want to connect.
# port: the port number of the AnalyticDB for MySQL cluster endpoint.
# user: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the following ty
pes of accounts: privileged accounts and standard accounts.
# passwd: the password of the account used to connect to the AnalyticDB for MySQL cluster.
# db: the name of the database in the AnalyticDB for MySQL cluster.
db = MySQLdb.connect(host='am-bp***.ads.aliyuncs.com', port=3306, user='account_name', passwd='account_password'
, db='db_name')
# Use the cursor() method to obtain an operation cursor.
cursor = db.cursor()
# Use the execute() method to execute SQL statements.
cursor.execute("SELECT VERSION()")
# Use the fetchone() method to obtain a data entry.
data = cursor.fetchone()
print "Database version : %s " % data
# Close the database connection.
db.close()
```

14.4.2.4. Druid connection pool

This topic describes how to use the Java Database Connectivity (JDBC) Druid connection pool to connect to an AnalyticDB for MySQL cluster.

Precautions

- When you use the Druid connection pool to connect to an AnalyticDB for MySQL cluster, we recommend that you set `keepAlive` to true. In this way, you can reuse connections and avoid short-lived connections.
- Use Druid 1.1.12 or later.

Configure the Druid connection pool

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
  <!-- jdbc_url: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can obtain the endpoint on the Cluster Information page of the AnalyticDB for MySQL console.-->
  <property name="url" value="{jdbc_url}" />
  <!-- jdbc_user: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the following types of accounts: privileged accounts and standard accounts.-->
  <property name="username" value="{jdbc_user}" />
  <!-- jdbc_password: the password of the account used to connect to the AnalyticDB for MySQL cluster.-->
  <property name="password" value="{jdbc_password}" />
  <!-- Set the initial size of the connection pool, and the minimum and maximum numbers of connections.-->
  <property name="initialSize" value="5" />
  <property name="minIdle" value="10" />
  <property name="maxActive" value="20" />
  <!-- Set the timeout period for obtaining a connection from the connection pool.-->
  <property name="maxWait" value="60000" />
  <!-- Set the interval for detecting idle connections to be closed. Unit: milliseconds.-->
  <property name="timeBetweenEvictionRunsMillis" value="2000" />
  <!-- Set the minimum validity period of a connection in the connection pool. Unit: milliseconds.-->
  <property name="minEvictableIdleTimeMillis" value="600000" />
  <property name="maxEvictableIdleTimeMillis" value="900000" />
  <property name="validationQuery" value="select 1" />
  <property name="testWhileIdle" value="true" />
  <!-- Specify whether to check the validity of the connection each time you obtain a connection from the connection pool. A value of true indicates that the validity of the connection is checked. A value of false indicates that the validity of the connection is not checked.-->
  <property name="testOnBorrow" value="false" />
  <!-- Specify whether to check the validity of the connection each time you return a connection to the connection pool. A value of true indicates that the validity of the connection is checked. A value of false indicates that the validity of the connection is not checked.-->
  <property name="testOnReturn" value="false" />
  <property name="keepAlive" value="true" />
  <property name="phyMaxUseCount" value="100000" />
  <!-- Set the filters to be used for monitoring statistics.-->
  <property name="filters" value="stat" />
</bean>
```

14.4.2.5. Java

This topic describes how to use MySQL Connector/J to connect to an AnalyticDB for MySQL cluster.

Supported MySQL Connector/J versions

AnalyticDB for MySQL supports the following MySQL Connector/J versions:

- 5.0 series: 5.0.2, 5.0.3, 5.0.4, 5.0.5, 5.0.7, and 5.0.8.
- 5.1 series: 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.7, 5.1.8, 5.1.11, 5.1.12, 5.1.13, 5.1.14, 5.1.15, 5.1.16, 5.1.17, 5.1.18, 5.1.19, 5.1.20, 5.1.21, 5.1.22, 5.1.23, 5.1.24, 5.1.25, 5.1.26, 5.1.27, 5.1.28, 5.1.29, 5.1.31, 5.1.32, 5.1.33, and 5.1.34.

Precautions

To create MySQL Connector/J connections in Java, you must add the MySQL Connector/J package to your project. You must add the path of the `mysql-connector-java-x.x.x.jar` file to the value of the `CLASSPATH` variable in your project. Otherwise, you cannot create a MySQL Connector/J connection.

Sample code for creating a MySQL Connector/J connection without retries

To connect to AnalyticDB for MySQL databases through MySQL Connector/J, you can add the following Java code to your business system:

```

Connection connection = null;
Statement statement = null;
ResultSet rs = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    // adb_url: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can obtain the endpoint on the Cluster Information page of the AnalyticDB for MySQL console. The default port number is 3306.
    // db_name: the name of the database in the AnalyticDB for MySQL cluster.
    String url = "jdbc:mysql://adb_url:3306/db_name? useUnicode=true&characterEncoding=UTF-8";
    Properties connectionProps = new Properties();
    // account_name: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the following types of accounts: privileged accounts and standard accounts.
    connectionProps.put("user", "account_name");
    // account_password: the password of the account used to connect to the AnalyticDB for MySQL cluster.
    connectionProps.put("password", "account_password");
    connection = DriverManager.getConnection(url, connectionProps);
    statement = connection.createStatement();
    String query = "select count(*) from information_schema.tables";
    rs = statement.executeQuery(query);
    while (rs.next()) {
        System.out.println(rs.getObject(1));
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (rs != null) {

```

```

if (rs != null) {
    try {
        rs.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
if (statement != null) {
    try {
        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
if (connection != null) {
    try {
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

Sample code for creating a MySQL Connector/J connection with retries

When you create a MySQL Connector/J connection, you can configure the following parameters to implement a retry mechanism:

```

public static final int MAX_QUERY_RETRY_TIMES = 3;
public static Connection conn = null;
public static Statement statement = null;
public static ResultSet rs = null;
public static void main(String[] args) throws ClassNotFoundException {
    // db_name: the name of the database in the AnalyticDB for MySQL cluster.
    String yourDB = "db_name";
    // account_name: the account used to connect to the AnalyticDB for MySQL cluster. AnalyticDB for MySQL offers the
    following types of accounts: privileged accounts and standard accounts.
    String username = "account_name";
    // account_password: the password of the account used to connect to the AnalyticDB for MySQL cluster.
    String password = "account_password";
    Class.forName("com.mysql.jdbc.Driver");
    // adb_url: the endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can obtain the endp
    oint on the Cluster Information page of the AnalyticDB for MySQL console. The default port number is 3306.
    String url = "jdbc:mysql://adb_url:3306/" + yourDB + "? useUnicode=true&characterEncoding=UTF-8";
    Properties connectionProps = new Properties();
    connectionProps.put("user", username);
    connectionProps.put("password", password);
    String query = "select id from test4dmp.test limit 10";

```

```

int retryTimes = 0;
// Run automatic retries by using loops.
while (retryTimes < MAX_QUERY_RETRY_TIMES) {
    try {
        getConn(url, connectionProps);
        execQuery(query);// Run a query.
        break; // If the query is run, exit the loop.
    } catch (SQLException e) {
        System.out.println("Met SQL exception: " + e.getMessage() + ", then go to retry task ...");
        try {
            if (conn == null || conn.isClosed()) {
                retryTimes++;
            }
        } catch (SQLException e1) {
            if (conn != null) {
                try {
                    conn.close();
                } catch (SQLException e2) {
                    e.printStackTrace();
                }
            }
        }
    }
}
// Clear connection resource.
closeResource();
}
/**
 * Get connection.
 *
 * @param url
 * @param connectionProps
 * @throws SQLException
 */
public static void getConn(String url, Properties connectionProps) throws SQLException {
    conn = DriverManager.getConnection(url, connectionProps);
}
/**
 * Query task execution logic.
 *
 * @param sql
 * @throws SQLException
 */
public static void execQuery(String sql) throws SQLException {
    Statement statement = null;
    ResultSet rs = null;
    statement = conn.createStatement();
}

```

```
for (int i = 0; i < 10; i++) {
    long startTs = System.currentTimeMillis();
    rs = statement.executeQuery(sql);
    int cnt = 0;
    while (rs.next()) {
        cnt++;
        System.out.println(rs.getObject(1) + " ");
    }
    long endTs = System.currentTimeMillis();
    System.out.println("Elapse Time: " + (endTs - startTs));
    System.out.println("Row count: " + cnt);
    try {
        Thread.sleep(160000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
/**
 * Close connection resource.
 */
public static void closeResource() {
    if (rs != null) {
        try {
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (statement != null) {
        try {
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

14.4.3. Enable PreparedStatement for a client in different programming languages

Background information

In most databases, SQL statements can be preprocessed on the server to improve database performance. AnalyticDB for MySQL databases deliver high performance without preprocessing SQL statements on the server side because these databases have the strong query capability, computing power, and plan cache feature.

AnalyticDB for MySQL databases do not support the protocol for preprocessing SQL statements on the server. In most programming languages, you can enable PreparedStatement for a client to prepare statements or interpolate parameters on the client.

This topic describes how to enable PreparedStatement for a client in different programming languages.

MySQL Connector/J (JDBC) driver

You can enable PreparedStatement for the MySQL Connector/J (JDBC) driver when the following condition is met:

`useServerPrepStmts=false` . For more information, visit [Configuration Properties for Connector/J](#).

 **Note** If the `useCursorFetch=true` condition is met, the `useServerPrepStmts=false` condition is overwritten and PreparedStatement cannot be enabled.

MariaDB Connector/J

You can enable PreparedStatement for MariaDB Connector/J if the following condition is met:

`useServerPrepStmts=false` . For more information, visit [About MariaDB Connector/J](#).

Go MySQL driver

You can enable PreparedStatement for the Go MySQL driver if the following condition is met:

`interpolateParams=true` . For more information, visit [go-sql-driver](#).

PDO

You can enable PreparedStatement for PDO if the following condition is met:

`PDO::ATTR_EMULATE_PREPARES=TRUE` . For more information, visit [setAttribute](#).

14.4.4. Use a client to connect to AnalyticDB for MySQL

14.4.4.1. SQL Workbench/J

SQL Workbench/J is a DBMS-independent, cross-platform SQL query tool. This topic describes how to use SQL Workbench/J to connect to an AnalyticDB for MySQL cluster.

Preparations

Before you use SQL Workbench/J, complete the following steps:

- Install the MySQL JDBC driver.
- Install SQL Workbench/J.
- Add the IP address of the device on which SQL Workbench/J is installed to the whitelist of the AnalyticDB for MySQL cluster. For more information, see [Configure a whitelist](#).

Precautions

For more information about how to enable PreparedStatement in SQL Workbench/J, see [Enable PreparedStatement for a client in different programming languages](#).

Procedure

1. Start SQL Workbench/J and choose **File > Manage Drivers....**

 **Note** If you use SQL Workbench/J for the first time, you must add the JDBC driver and its JAR file. When you use SQL Workbench/J later, you can skip step 2.

2. In the **Manage drivers** dialog box that appears, select **MySQL** as the driver, add the JAR file of the driver, and then click **OK**.
3. Choose **File > Connect window**. In the **Select Connection Profile** dialog box that appears, configure the connection parameters as required.

Parameter	Description
New profile	The name of the connection, which facilitates subsequent management.
Driver	The type of the driver. Select MySQL from the drop-down list.
URL	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect, including the endpoint, port number, and database name. The format is <code>jdbc:mysql://hostname:port/name_of_database</code> , where: <ul style="list-style-type: none"> ◦ <code>hostname</code>: the public endpoint or VPC endpoint of the cluster. ◦ <code>port</code>: the port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306. ◦ <code>name_of_database</code>: the name of the database in the cluster. This parameter is optional.
Username	The account used to connect to the AnalyticDB for MySQL cluster. You can use either of the following accounts: <ul style="list-style-type: none"> ◦ Privileged accounts. ◦ Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.

4. After configuring the preceding parameters, click **Test** to test connectivity. After the connection passes the test, click **OK** to connect to the AnalyticDB for MySQL database.

14.4.4.2. DbVisualizer

This topic describes how to use DbVisualizer to connect to an AnalyticDB for MySQL cluster.

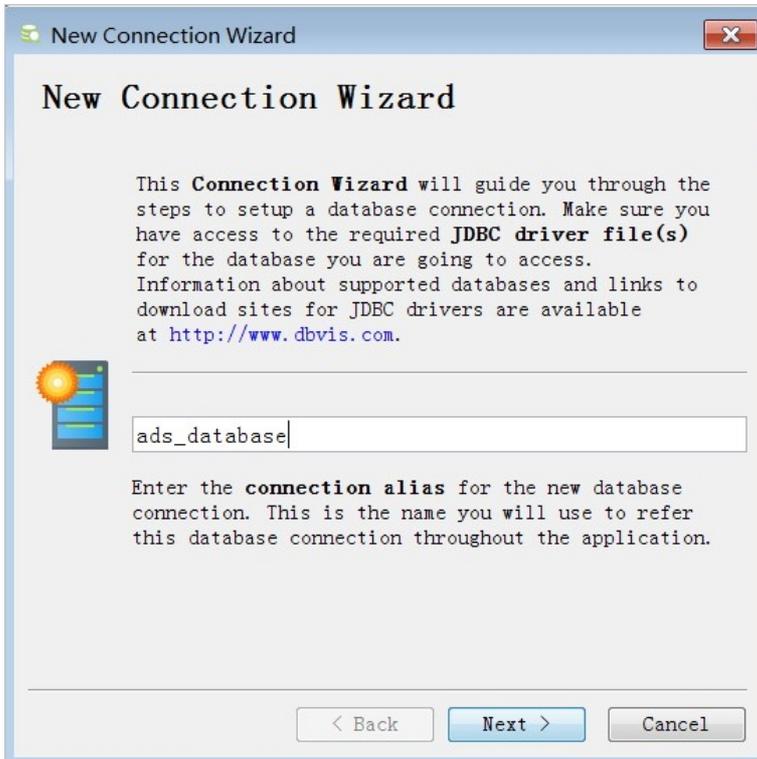
Preparations

Before you use DbVisualizer, complete the following steps:

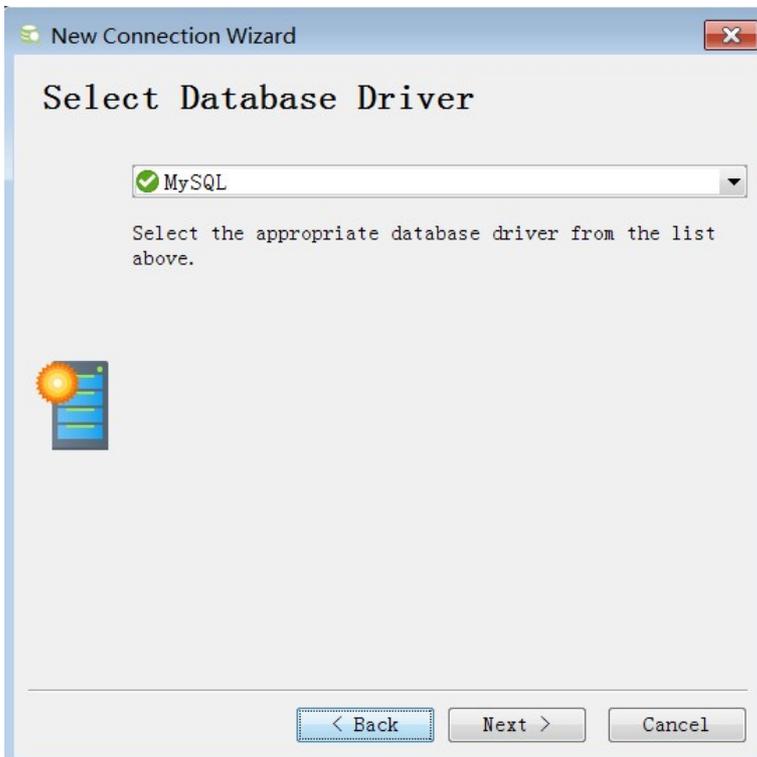
- Install MySQL Connector/J.
- Install DbVisualizer.
- Add the IP address of the device on which DbVisualizer is installed to the whitelist of the AnalyticDB for MySQL cluster. For more information, see [Configure a whitelist](#).

Procedure

1. Start DbVisualizer and choose **Tools > Connection Wizard**. In the **New Connection Wizard** dialog box, enter a name for the connection to facilitate subsequent management.

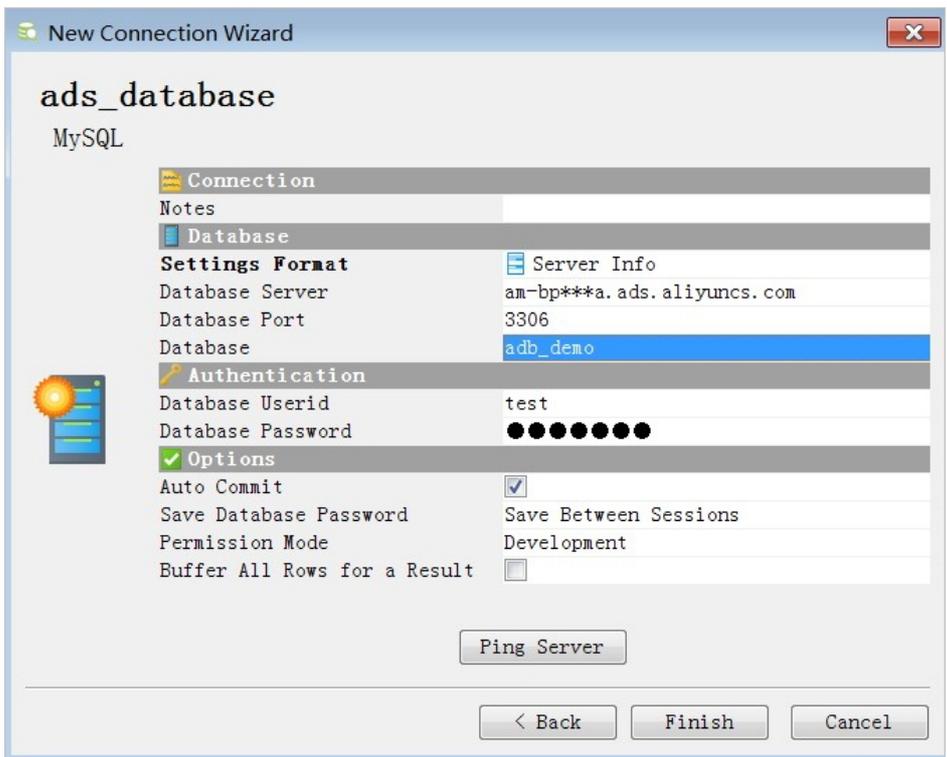


2. Click **Next** and select **MySQL** as the database driver from the drop-down list.



3. Click **Next** and configure the connection parameters.

Parameter	Description
Notes	The description of the connection.
Database Server	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can view the connection information about the cluster on the Cluster Information page of the AnalyticDB for MySQL console.
Database Port	The port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306.
Database	The name of the database in the AnalyticDB for MySQL cluster.
Database Userid	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> ○ Privileged accounts. ○ Standard accounts that have the permissions to connect to the cluster.
Database Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.



4. After you configure the preceding parameters, click **Ping Server** to test connectivity. After the connection passes the test, click **Finish**.

After the AnalyticDB for MySQL cluster is connected, you can use DbVisualizer to manage data.

14.4.4.3. DBeaver

This topic describes how to use DBeaver to connect to an AnalyticDB for MySQL cluster.

Context

DBeaver is a free and open source database management tool distributed under General Public License (GPL). It is designed for developers and database administrators. DBeaver supports databases that are compatible with Java Database Connectivity (JDBC), such as MySQL, PostgreSQL, Oracle, DB2, SQL Server, and Sybase. DBeaver provides a graphical user interface (GUI), on which you can view database schemas, execute SQL statements and scripts, view and export data, and process binary large object (BLOB) or character large object (CLOB) data.

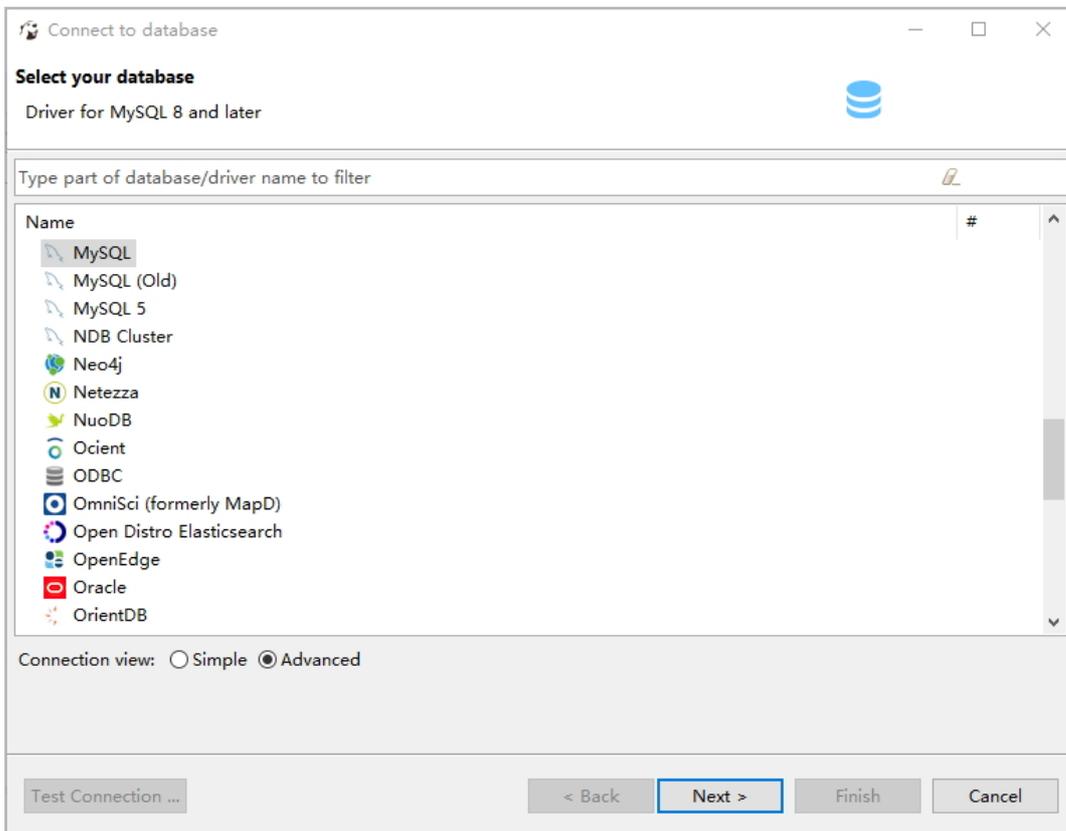
Preparations

Before you use DBeaver, complete the following steps:

- Click [here](#) to download and install DBeaver.
- Install MySQL Connector/J.
- Add the IP address of the device on which DBeaver is installed to the whitelist of the AnalyticDB for MySQL cluster. For more information, see [Configure a whitelist](#).

Procedure

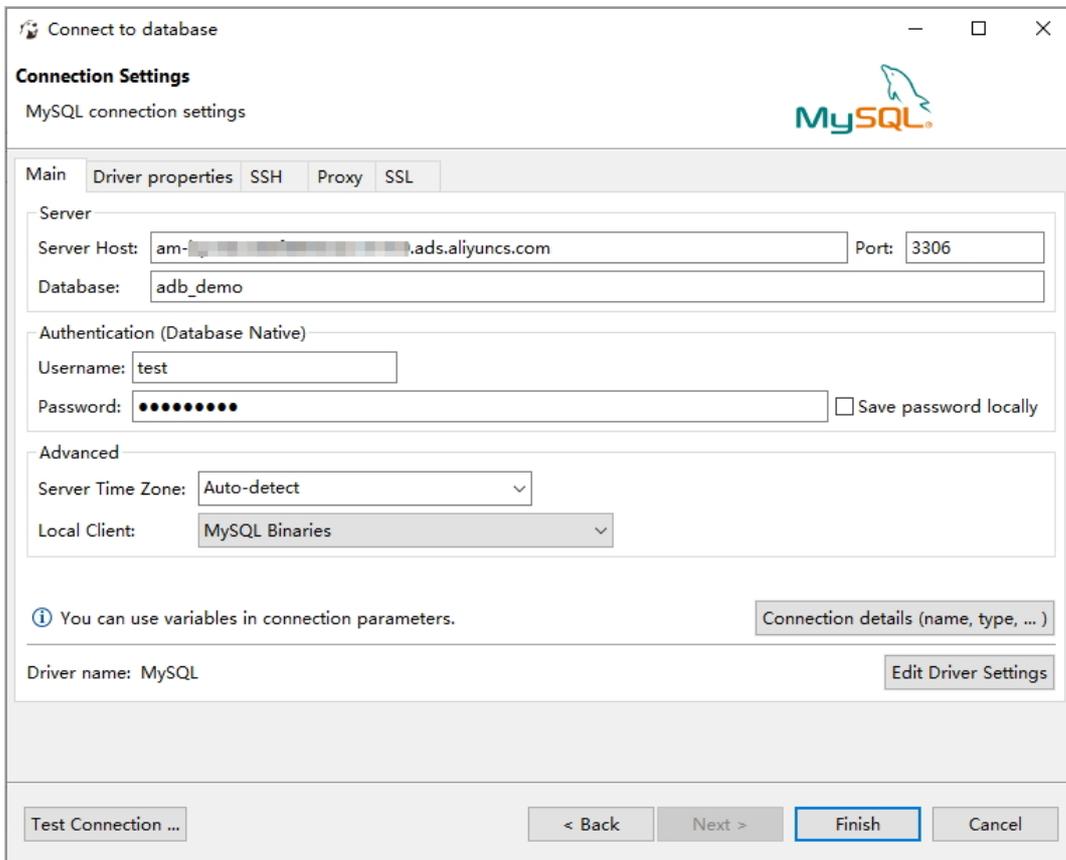
1. Start DBeaver and choose **Database > New Connection**.
2. In the **Create new connection** dialog box, select **MySQL** as the connection type and click **Next**.



3. On the **General** tab of the **Create new connection** dialog box, configure the connection parameters.

Parameter	Description
Server Host	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can view the connection information about the cluster on the Cluster Information page of the AnalyticDB for MySQL console.

Parameter	Description
Port	The port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306.
Database	The name of the database in the AnalyticDB for MySQL cluster.
User Name	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> Privileged accounts. Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.



- After you configure the preceding parameters, click **Test Connection...**. After the connection passes the test, click **Finish** to connect to the cluster.

14.4.4.4. Navicat

This topic describes how to use Navicat to connect to an AnalyticDB for MySQL cluster.

Context

Navicat is a fast, reliable, and cost-effective database management tool designed to simplify database management and reduce costs. Navicat provides a graphical user interface (GUI) for you to create remote connections from the local computer to AnalyticDB for MySQL clusters and manage data.

Preparations

Before you use Navicat for MySQL, complete the following steps:

- Click [here](#) to download and install Navicat for MySQL.
- Add the IP address of the device on which Navicat for MySQL is installed to the whitelist of the AnalyticDB for MySQL cluster to which you want to connect. For more information, see [Configure a whitelist](#).

Procedure

1. Start Navicat for MySQL and choose **File > New Connection > MySQL**. In the **New Connection** dialog box, configure the connection parameters.

Parameter	Description
Connection Name	The name of the connection. We recommend that you choose an identifiable name to facilitate subsequent management.
Host	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can view the connection information about the cluster on the Cluster Information page of the AnalyticDB for MySQL console.
Port	The port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306.
User Name	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> ◦ Privileged accounts. ◦ Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.

If your operating system is macOS, add the database name after you configure the connection information.

2. Click **Test Connection**. After the connection passes the test, click **OK**.

The connection to the AnalyticDB for MySQL cluster is created but is not enabled. You must manually enable the connection.

3. Right-click a connection name and select **Open Connection**. Right-click a database name to enable the database connection. Then, you can use Navicat for MySQL to manage data.

14.5. Manage database clusters

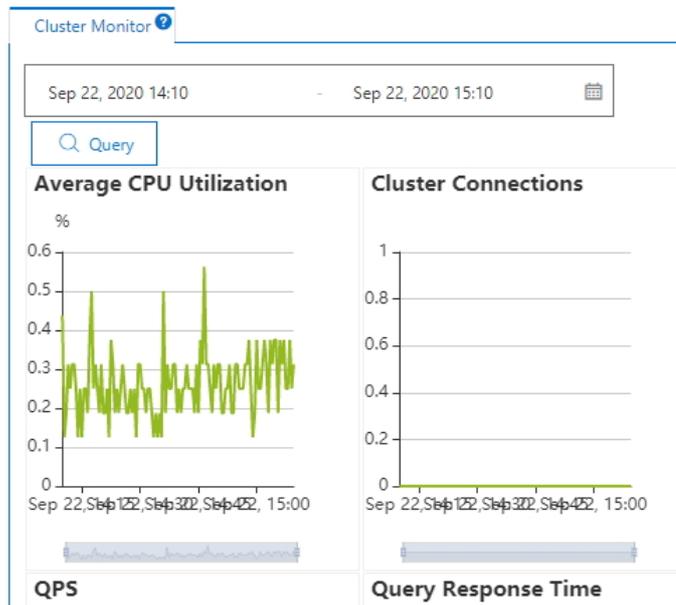
14.5.1. View monitoring information

You can view cluster monitoring information in real time in the AnalyticDB for MySQL console.

Procedure

1. [Log on to the AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to view monitoring information.
4. In the left-side navigation pane, click **Monitoring Information** to view monitoring information of the cluster.

Monitoring Informat...



The monitoring information includes CPU utilization, cluster connections, QPS, and query response time.

14.5.2. Change specifications

This topic describes how to change the specifications of AnalyticDB for MySQL clusters.

Context

Note You can modify only the number of node groups.

Procedure

1. Log on to the [AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. Click **Change Specifications** in the Actions column corresponding to the cluster.
4. On the **Change Specifications** page, set **Node Groups** and click **Submit**.

14.5.3. Delete a cluster

You can delete AnalyticDB for MySQL clusters.

Procedure

1. Log on to the [AnalyticDB for MySQL console](#).
2. In the upper-left corner of the page, select the region where the cluster resides.
3. Choose **More > Delete** in the Actions column corresponding to the cluster that you want to delete.
4. In the **Delete Cluster** message, click **OK**.

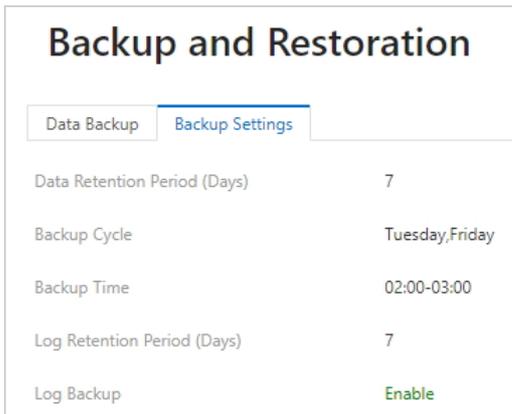
14.6. Backup and restoration

14.6.1. Back up data

AnalyticDB for MySQL uses physical backups (snapshots) to automatically back up data every Tuesday and Friday from 02:00 to 03:00. The backup files are retained for seven days.

Procedure

1. [Log on to the AnalyticDB for MySQL console.](#)
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to back up data.
4. In the left-side navigation pane, click **Backup and Restoration**.
5. Click the **Backup Settings** tab to view the automatic settings.



14.6.2. Restore data

In AnalyticDB for MySQL, the cluster administrator can restore data from backup sets.

14.7. Diagnostics and optimization

14.7.1. Use functions related to slow SQL queries

AnalyticDB for MySQL provides the slow SQL analysis feature. You can view the trends and statistics of slow query logs and obtain diagnostic analysis and suggestions.

Procedure

1. [Log on to the AnalyticDB for MySQL console.](#)
2. In the upper-left corner of the page, select the region where the cluster resides.
3. On the Clusters page, click **Cluster ID** corresponding to the cluster for which you want to view slow query logs.
4. In the left-side navigation pane, click **Query**.

You can use one of the following methods to view slow query logs:

- Slow SQL Trend: displays the trend line chart of the slow query logs.
- Slow SQL Details: displays the detailed data of the slow query logs.

14.8. Account and permission management

14.8.1. Permission model

Permission levels

An AnalyticDB for MySQL cluster supports the following four levels of permission control:

- **GLOBAL** : cluster-level permissions
- **DB** : database-level permissions
- **TABLE** : table-level permissions
- **COLUMN** : column-level (field) permissions

If you want a user to query the data of one specific column in a table, you can grant the **SELECT** permission on the column to the user. Example: `GRANT select (customer_id) ON customer TO 'test321'` .

Operations and corresponding permissions

Operation	Required permission	Supported permission level
SELECT	SELECT	<ul style="list-style-type: none"> • DB • TABLE • COLUMN
INSERT	INSERT	<ul style="list-style-type: none"> • DB • TABLE • COLUMN
INSERT...SELECT...FROM...	<ul style="list-style-type: none"> • INSERT • SELECT 	<ul style="list-style-type: none"> • DB • TABLE • COLUMN
UPDATE	UPDATE	<ul style="list-style-type: none"> • DB • TABLE • COLUMN
DELETE	DELETE	<ul style="list-style-type: none"> • DB • TABLE
TRUNCATE TABLE	DROP	<ul style="list-style-type: none"> • DB • TABLE
ALTER TABLE	<ul style="list-style-type: none"> • ALTER • INSERT • CREATE 	<ul style="list-style-type: none"> • DB • TABLE
CREATE DATABASE	CREATE	N/A
CREATE TABLE	CREATE	<ul style="list-style-type: none"> • DB • TABLE
SHOW CREATE TABLE	SELECT	<ul style="list-style-type: none"> • DB • TABLE
DROP DATABASE	DROP	DB

Operation	Required permission	Supported permission level
DROP TABLE	DROP	<ul style="list-style-type: none"> DB TABLE
CREATE VIEW	<ul style="list-style-type: none"> CREATE_VIEW SELECT 	<ul style="list-style-type: none"> DB TABLE <p>To execute the <code>CREATE VIEW REPLACE</code> statement, the DROP permission is also required in addition to the preceding permissions.</p>
DROP VIEW	DROP	<ul style="list-style-type: none"> DB TABLE
SHOW CREATE VIEW	<ul style="list-style-type: none"> SHOW_VIEW SELECT 	<ul style="list-style-type: none"> DB TABLE
CREATE_PROCEDURE	CREATE_ROUTINE	N/A
DROP_PROCEDURE	ALTER_ROUTINE	N/A
CREATE_EVENT	EVENT	N/A
DROP_EVENT	EVENT	N/A
CREATE USER/DROP USER/RENAME USER	CREATE_USER	N/A
SET PASSWORD	SUPER	N/A
GRANT/REVOKE	GRANT	N/A

14.8.2. Manage database accounts and permissions

- For more information about how to create a RAM user, see `CREATE USER`.
- For more information about how to grant permissions to a RAM user, see `GRANT`.
- For more information about how to revoke the permissions of a RAM user, see `REVOKE`.
- For more information about how to modify the name of a database account, see `RENAME USER`.
- For more information about how to delete a database account, see `DROP USER`.

14.9. Data visualization

14.9.1. Tableau

This topic describes how to use Tableau to connect to an AnalyticDB for MySQL cluster and perform data analytics and visualization.

Preparations

Before you use Tableau, complete the following steps:

- Install the MySQL Open Database Connectivity (ODBC) driver. We recommend that you use MySQL Connector/ODBC 3.5.1 or 5.3.
- Install Tableau 9.0 or later.

Procedure

1. Start Tableau and click **MySQL** in the left-side navigation pane to create a MySQL connection. In the MySQL dialog box, configure the connection parameters.

Parameter	Description
Server	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect.
Port	The port number of the AnalyticDB for MySQL cluster endpoint.
User Name	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> ◦ Privileged accounts. ◦ Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.

2. After you configure the preceding parameters, click **Log On** to connect to the AnalyticDB for MySQL cluster.

Use Tableau

In Tableau, you can view the databases on which you have permissions. After you select a database, you can read tables, preview data, and generate visual reports. For more information about how to use Tableau, visit [Tableau](#).

14.9.2. QlikView

This topic describes how to use QlikView to connect to an AnalyticDB for MySQL cluster and build a business intelligence (BI) system.

Preparations

Before you use QlikView, complete the following steps:

- Install the MySQL Open Database Connectivity (ODBC) driver. We recommend that you use MySQL Connector/ODBC 3.5.1 or 5.3.
- Install QlikView 11.20.x.

Procedure

1. On the host where QlikView is installed, click **Control Panel**, and choose **System and Security > Administrative Tools > ODBC Data Sources**. This path may vary with different operating systems. In the ODBC Data Source Administrator dialog box, add a system data source name (DSN) and select **MySQL ODBC 5.xx Driver** as the data source.

Parameter	Description
Data Source Name	The name of the database in the AnalyticDB for MySQL cluster.

Parameter	Description
TCP/IP Server	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect.
Port	The port number of the AnalyticDB for MySQL cluster endpoint.
User	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> Privileged accounts. Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.

- After you configure the preceding parameters, click **Test** to test connectivity. After the connection passes the test, click **OK** to create the ODBC connection.
- Start QlikView and choose **File > Edit Script**. Select the database name specified in Step 1 to test connectivity.
- After the database passes the connectivity test, you can use the following **SELECT** statement to obtain the data in the AnalyticDB for MySQL database:

```
SELECT * FROM DATABASE_NAME.TABLE_NAME;
```

For example, use the following statement to obtain data from the user_info table:

```
SELECT * FROM adb_database.user_info;
```

Use QlikView

After you obtain data from AnalyticDB for MySQL databases, you can use QlikView to perform more operations. For more information about how to use QlikView, visit [QlikView](#).

14.9.3. FineReport

This topic describes how to use FineReport to connect to an AnalyticDB for MySQL cluster and manage reports.

Preparations

Before you use FineReport, complete the following steps:

- Install MySQL Connector/J.
- Install FineReport.

Procedure

- Start FineReport and choose **Server > Define Data Connection**.
- In the **Define Data Connection** dialog box, configure the parameters.

Parameter	Description
Database	The type of the database. Select MySQL from the drop-down list.
Driver	The type of the driver. Select MySQL JDBC driver from the drop-down list.

Parameter	Description
URL	<p>The connection string of the AnalyticDB for MySQL cluster to which you want to connect. The format is <code>jdbc:mysql://hostname:port</code>.</p> <ul style="list-style-type: none"> <code>hostname</code>: the public or VPC endpoint of the cluster. <code>port</code>: the port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306.
User Name	<p>The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types:</p> <ul style="list-style-type: none"> Privileged accounts. Standard accounts that have the permissions to connect to the cluster.
Password	<p>The password of the account used to connect to the AnalyticDB for MySQL cluster.</p>

- After you configure the preceding parameters, click **Test Connection**. After the connection passes the test, click **Confirm** to the AnalyticDB for MySQL cluster.

Use FineReport

After FineReport is connected to the AnalyticDB for MySQL cluster, you can obtain the data in AnalyticDB for MySQL databases and use FineReport to generate reports. For more information about how to use FineReport, visit [FineReport](#).

14.10. Data migration and synchronization

14.10.1. Use Kettle to synchronize local data to AnalyticDB for MySQL

This topic describes how to use Kettle to synchronize local data to AnalyticDB for MySQL. Excel data is used in the example.

Background information

Kettle is a popular open source extract-transform-load (ETL) tool used for data collection, conversion, and migration. Kettle supports not only various relational databases and NoSQL databases such as HBase and MongoDB, but also niche data sources such as Microsoft Office Excel and Access. Kettle can support more data sources by using extensions and plug-ins.

For more information, visit [Kettle](#).

Preparations

Before you use Kettle, complete the following steps:

- Click [here](#) to download and install Kettle.
- Create a database and a table in an AnalyticDB for MySQL cluster.

For more information about how to create databases and tables, see [Create a database](#) and [CREATE TABLE](#).

- Add the IP address of the device on which Kettle is installed to the whitelist of the AnalyticDB for MySQL cluster. For more information, see [Configure a whitelist](#).

Procedure

1. Start Kettle, and choose **File > New > Conversion** to create a conversion task.
2. Choose **File > New > Database Connection** to create a database connection for the conversion task.

Parameter	Description
Connection Name	The name of the connection. We recommend that you choose an identifiable name, which facilitates subsequent management.
Connection Type	The type of the connection. Select MySQL from the drop-down list.
Access	The access mode of the connection. Select Native (JDBC) .
Host Name	The endpoint of the AnalyticDB for MySQL cluster to which you want to connect. You can view the connection information about the cluster on the Cluster Information page of the AnalyticDB for MySQL console.
Database Name	The name of the database in the AnalyticDB for MySQL cluster.
Port Number	The port number of the AnalyticDB for MySQL cluster endpoint. The default port number is 3306.
User Name	The account used to connect to the AnalyticDB for MySQL cluster. You can use one of the following account types: <ul style="list-style-type: none"> ◦ Privileged accounts. ◦ Standard accounts that have the permissions to connect to the cluster.
Password	The password of the account used to connect to the AnalyticDB for MySQL cluster.

 **Note** Do not select **Use Result Streaming Cursor** when you configure the parameters.

3. After you configure the preceding parameters, click **Test**. In the **Database Connection Test** dialog box, follow the prompts to verify if the connection to the database is successful. After the connection passes the test, click **OK**.
4. In the left-side navigation pane of Kettle, click the **Core objects** tab, and choose **Input > Excel Input**. Drag and drop **Excel Input** to the workspace.
5. Double-click **Excel Input** in the workspace. In the **Excel Input** dialog box, click **Browse** and **Add** to add an Excel file to **Selected Files**.
Configure parameters on tabs such as **Worksheet**, **Content**, and **Field**, and click **Preview** to check whether the specified values meet your requirements.
6. In the left-side navigation pane of Kettle, click the **Core objects** tab, and choose **Output > Table Output**. Drag and drop **Table Output** to the workspace.
7. Add a connection line from **Excel Input** to **Table Output**.
8. Double-click **Table Output** in the workspace. In the **Table Output** dialog box, configure the parameters.
 - **Target Schema**: Enter the name of the AnalyticDB for MySQL database.
 - **Target Table**: Enter the name of the table in the AnalyticDB for MySQL database.

- Select **Specify database fields**.
- Select **Use batch update for inserts**.

On the Database fields tab of the **Table Output** dialog box, click **Get fields** and **Enter field mapping** to map columns in the Excel file to those in the AnalyticDB for MySQL table.

9. Click the white arrow to perform the conversion. During this period, you can check the operation logs and operating statuses.

After the data in the Excel file is synchronized to the AnalyticDB for MySQL database, you can use AnalyticDB for MySQL to analyze the data.

14.11. SQL manual

14.11.1. Data types

Data types supported by AnalyticDB for MySQL

Keyword	Data type	Valid value
BOOLEAN	The Boolean type	Valid values: <code>0</code> and <code>1</code> . A value of <code>0</code> indicates false. A value of <code>1</code> indicates true. A BOOLEAN value is 1 bit in size.
TINYINT	The tiny integer type	Valid values: <code>-128 to 127</code> . A TINYINT value is 1 byte in size.
SMALLINT	The small integer type	Valid values: <code>-32768 to 32767</code> . A SMALLINT value is 2 bytes in size.
INT or INTEGER	The integer type	Valid values: <code>-2147483648 to 2147483647</code> . An INT value is 4 bytes in size.
BIGINT	The big integer type	Valid values: <code>-9223372036854775808 to 9223372036854775807</code> . A BIGINT value is 8 bytes in size.
FLOAT	The single-precision floating-point type	Valid values: <code>-3.402823466E+38 to -1.175494351E-38</code> , <code>0</code> , and <code>1.175494351E-38 to 3.402823466E+38</code> . The FLOAT type follows the IEEE standard. A FLOAT value is 4 bytes in size.
DOUBLE	The double-precision floating-point type	Valid values: <code>-1.7976931348623157E+308 to -2.2250738585072014E-308</code> , <code>0</code> , and <code>2.2250738585072014E-308 to 1.7976931348623157E+308</code> . The DOUBLE type follows the IEEE standard. A DOUBLE value is 8 bytes in size.

Keyword	Data type	Valid value
DECIMAL(M,D)	The decimal type	M is the maximum precision, and its value range is 1 to 1000. D is the decimal scale. The value of D must be less than or equal to that of M.
VARCHAR	The variable-length string type	A VARCHAR value can be up to 16 MB in size. You do not need to specify the size when you use VARCHAR.
DATE	The date type	Valid values: '0001-01-01' to '9999-12-31'. A DATE value is in the 'YYYY-MM-DD' format and is 4 bytes in size.
TIME	The time type	Valid values: '00:00:00' to '23:59:59'. A TIME value is in the 'HH:MM:SS' format and is 8 bytes in size.
DATETIME	The date and time type	Valid values: '0001-01-01 00:00:00.000' to '9999-12-31 23:59:59.999'. A DATETIME value is in the 'YYYY-MM-DD HH:MM:SS' format. It is 8 bytes in size and in the UTC format.  Note DATETIME uses UTC time. You cannot change the time zone for DATETIME values.
TIMESTAMP	The timestamp type	Valid values: '0001-01-01 00:00:00.000' to '9999-12-31 23:59:59.999'. A TIMESTAMP value is in the 'YYYY-MM-DD HH:MM:SS' format. It is 4 bytes in size and in the UTC format.  Note TIMESTAMP uses the time zone of the database system by default. You can specify the time zone for each session.

Comparison with MySQL data types

AnalyticDB for MySQL	MySQL	Difference
BOOLEAN	BOOL and BOOLEAN	No difference.

AnalyticDB for MySQL	MySQL	Difference
TINYINT	TINYINT	No difference.
SMALLINT	SMALLINT	No difference.
INT and INTEGER	INT and INTEGER	No difference.
BIGINT	BIGINT	No difference.
FLOAT	FLOAT[(M,D)]	No difference.
DOUBLE	DOUBLE[(M,D)]	No difference.
DECIMAL	DECIMAL	AnalyticDB for MySQL supports a maximum precision of 1,000 digits, while MySQL supports a maximum precision of only 65 digits.
VARCHAR	VARCHAR	The VARCHAR type in AnalyticDB for MySQL corresponds to the CHAR, VARCHAR, TEXT, MEDIUMTEXT, and LONGTEXT types in MySQL.
DATE	DATE	MySQL supports the value 0000-00-00, whereas AnalyticDB for MySQL automatically converts the value 0000-00-00 to NULL.
TIME	TIME	AnalyticDB for MySQL is precise to the millisecond. MySQL supports custom precision levels.
DATETIME	DATETIME	MySQL supports the value 0000-00-00 00:00:00, whereas AnalyticDB for MySQL automatically converts the value 0000-00-00 00:00:00 to NULL. AnalyticDB for MySQL is precise to the millisecond. MySQL supports custom precision levels.
TIMESTAMP	TIMESTAMP	AnalyticDB for MySQL is precise to the millisecond. MySQL supports custom precision levels.

14.11.2. Data definition statements

14.11.2.1. CREATE DATABASE

Create a database

 **Note** You can create up to 256 databases in each AnalyticDB for MySQL cluster.

Syntax

```
CREATE DATABASE [IF NOT EXISTS] db_name
```

Parameters

`db_name` : the name of the database. The database name can be up to 64 characters in length, and can contain letters, digits, and underscores (_). It must start with a lowercase letter and cannot contain consecutive underscores (_).

 **Note** Do not use `analyticdb` as the database name. The name `analyticdb` is reserved for a built-in database.

Example

```
CREATE DATABASE adb_demo;
```

Use a database

You can execute the `USE db_name` statement to use a database after it is created.

Syntax

```
USE db_name
```

Example

```
use adb_demo;
show tables;
+-----+
|Tables_in_adb_demo      |
+-----+
|customer                |
|test_table               |
```

14.11.2.2. CREATE TABLE

You can execute the `CREATE TABLE` statement to create a table in AnalyticDB for MySQL.

Syntax

```
CREATE TABLE [IF NOT EXISTS] table_name
({column_name column_type [column_attributes] [ column_constraints ] [COMMENT 'string']
| table_constraints}
[, ... ] )
table_attribute
[partition_options]
[AS] query_expression
COMMENT 'string'
```

column_attributes:
 [DEFAULT default_expr]
 [AUTO_INCREMENT]

column_constraints:
 [{(NOT NULL|NULL)}]
 [PRIMARY KEY]

table_constraints:
 [{(INDEX|KEY) [index_name] (column_name,...)}]
 [PRIMARY KEY [index_name] (column_name,...)]
 [CLUSTERED KEY [index_name] (column_name,...)]

table_attribute:
 DISTRIBUTED BY HASH(column_name,...) | DISTRIBUTED BY BROADCAST

partition_options:
 PARTITION BY
 {VALUE(column_name) | VALUE(date_format(column_name, ?))}
 [LIFECYCLE N]

Parameters

Parameter	Description
table_name	<p>The name of the table.</p> <p>The table name must be 1 to 127 characters in length, and can contain letters, digits, and underscores (_). The table name must start with a letter or underscore (_).</p> <p>Specify the table name in the db_name.table_name format to distinguish tables that have the same name across different databases.</p>
column_name	<p>The name of the column.</p> <p>The column name must be 1 to 127 characters in length and can contain letters, digits, and underscores (_). The table name must start with a letter or underscore (_).</p>

Parameter	Description
<code>column_type</code>	<p>The data type of the column to be added.</p> <p>For more information about the data types supported by AnalyticDB for MySQL, see Data types.</p>
<code>column_attributes</code>	<ul style="list-style-type: none"> <code>DEFAULT default_expr</code> : the default value of the column. Enter an expression without variables, such as <code>current_timestamp</code> . <p>If this parameter is not specified, the default value is NULL.</p> <ul style="list-style-type: none"> <code>AUTO_INCREMENT</code> : optional. Specifies whether the column is an auto-increment column. The data type of an auto-increment column must be BIGINT. This is because AnalyticDB for MySQL provides unique values for an auto-increment column, but these values are not incremented in sequence.
<code>column_constraints</code>	<ul style="list-style-type: none"> <code>NOT NULL NULL</code> : specifies whether the column accepts the NULL value. A value of <code>NOT NULL</code> indicates that the column does not accept the <code>NULL</code> value. A value of <code>NULL</code> indicates that the column accepts the <code>NULL</code> value. Default value: NULL. <code>PRIMARY KEY</code> : the primary key of the column. You can define multiple primary keys in the <code>PRIMARY KEY(column_name [, ...])</code> format.
<code>table_constraints</code>	<p><code>INDEX KEY</code> : the inverted index.</p> <p>AnalyticDB for MySQL automatically creates indexes for whole tables. You do not need to manually create an index.</p>
<code>PRIMARY KEY</code>	<p>The index for the primary keys.</p> <ul style="list-style-type: none"> Only tables with primary keys support the DELETE and UPDATE operations. The primary keys must include the partition key. We recommend that you put the partition key before the primary key combination.
<code>CLUSTERED KEY</code>	<p>The clustered index. It defines the columns used for sorting data in the table. The logical order of the key values in the clustered index determines the physical order of the corresponding rows in the table. You can add only one clustered index for each table.</p> <p>For example, <code>clustered key col5_col6_cls_index(col5,col6)</code> specifies the <code>col5 col6</code> clustered index. <code>col5 col6</code> and <code>col6 col5</code> are different clustered indexes.</p>

Parameter	Description
<code>DISTRIBUTED BY HASH(column_name,...)</code>	<p>The distribution key of the fact table. The data of the table is distributed based on the hash value of the columns specified by <code>column_name</code> .</p> <p>AnalyticDB for MySQL allows you to select multiple fields as the partition key.</p>
<code>DISTRIBUTED BY BROADCAST</code>	<p>The dimension table. The dimension table is stored on each node of a cluster. For performance reasons, we recommend that you do not store large amounts of data in the dimension table.</p>
<code>partition_options</code>	<p>The options for fact table partitions.</p> <p>AnalyticDB for MySQL manages the lifecycle of tables based on the <code>LIFECYCLE N</code> parameter. That is, AnalyticDB for MySQL only stores N number of partitions. All other partitions are deleted.</p> <p>For example, <code>PARTITION BY VALUE(column_name)</code> specifies that the table is partitioned based on the column specified by <code>column_name</code> . <code>PARTITION BY VALUE(DATE_FORMAT(column_name, '%Y%m%d'))</code> specifies that the table is partitioned based on the column specified by <code>column_name</code> after the column is formatted to a date format such as <code>20190101</code> .</p> <p><code>LIFECYCLE 365</code> specifies that a maximum of 365 partitions can be retained on each node. That is, only data of the last 365 days is stored. When you write data on the 366th day, the data from the first day is deleted.</p>

Precautions

- AnalyticDB for MySQL clusters use the UTF-8 encoding format during table creation. This encoding format is equivalent to the utf8mb4 format in MySQL. AnalyticDB for MySQL does not support other encoding formats.
- You can create up to `256 × Number of node groups` tables in an AnalyticDB for MySQL cluster.

Examples

- Create a test table.

```
create table test (
  id bigint auto_increment,
  name varchar,
  value int,
  ts timestamp
)
DISTRIBUTED BY HASH(id)
```

The test table is a fact table. The `id` column is an auto-increment column. The distribution key is `id`. The data of the table is distributed based on the hash value of the `id` column.

- Create a customer table.

```
CREATE TABLE customer (
  customer_id bigint NOT NULL COMMENT 'Customer ID',
  customer_name varchar NOT NULL COMMENT 'Customer name',
  phone_num bigint NOT NULL COMMENT 'Phone number',
  city_name varchar NOT NULL COMMENT 'City',
  sex int NOT NULL COMMENT 'Gender',
  id_number varchar NOT NULL COMMENT 'ID card number',
  home_address varchar NOT NULL COMMENT 'Home address',
  office_address varchar NOT NULL COMMENT 'Office address',
  age int NOT NULL COMMENT 'Age',
  login_time timestamp NOT NULL COMMENT 'Logon time',
  PRIMARY KEY (login_time,customer_id,phone_num)
) DISTRIBUTED BY HASH(customer_id)
PARTITION BY VALUE(DATE_FORMAT(login_time, '%Y%m%d')) LIFECYCLE 30
COMMENT 'Customer information table';
```

The customer table is a fact table. In the table, the distribution key is `customer_id`. The partition key is `login_time`. `login_time`, `customer_id`, and `phone_num` form the primary key combination.

14.11.2.3. ALTER TABLE

You can execute the ALTER TABLE statement to modify a table.

 **Note** You cannot modify the data type of a column in AnalyticDB for MySQL.

Syntax

```
ALTER TABLE table_name
  ADD COLUMN (column_name column_definition,...)
| ADD {INDEX|KEY} [index_name] (column_name,...)
| ADD CLUSTERED [INDEX|KEY] [index_name] (column_name,...)
| DROP COLUMN column_name
| DROP {INDEX|KEY} index_name
| DROP CLUSTERED [INDEX|KEY] index_name
| MODIFY COLUMN column_name column_definition
| RENAME new_table_name
| TRUNCATE PARTITION {partition_names | ALL}
```

Add a column

Syntax

```
ALTER TABLE db_name.table_name ADD column_name data_type;
```

Example

Add the `province` column of the `VARCHAR` type to the customer table.

```
ALTER TABLE adb_demo.customer ADD COLUMN province varchar comment 'Province';
```

Delete a column

Syntax

```
ALTER TABLE db_name.table_name DROP column_name data_type;
```

Example

Delete the `province` column of the `VARCHAR` type from the customer table.

Modify the comments of a column

Syntax

```
ALTER TABLE db_name.table_name MODIFY COLUMN column_name data_type comment 'new_comment';
```

Example

Modify the `COMMENT` of the `province` column to the province where the customer locates in the customer table.

```
ALTER TABLE adb_demo.customer MODIFY COLUMN province varchar comment 'Province where the customer locates';
```

Change the value constraint for a column from NOT NULL to NULL

 **Note** You can change the value constraint only from NOT NULL to NULL, but not from NULL to NOT NULL.

Syntax

```
ALTER TABLE db_name.table_name MODIFY COLUMN column_name data_type {NULL}
```

Example

Change the value of the `province` column to `NULL` in the customer table.

```
ALTER TABLE adb_demo.customer MODIFY COLUMN province varchar NULL;
```

Change the default value of a column**Syntax**

```
ALTER TABLE db_name.table_name MODIFY COLUMN column_name data_type DEFAULT 'default'
```

Example

Change the default value of the `sex` column to `0` (indicating male) in the customer table.

```
ALTER TABLE adb_demo.customer MODIFY COLUMN sex int(11) NOT NULL DEFAULT 0;
```

Add a clustered index

 **Note** You can add only one clustered index for each table.

Syntax

```
ALTER TABLE db_name.table_name ADD CLUSTERED KEY index_name(column_name1,column_name2);
```

```
ALTER TABLE db_name.table_name ADD CLUSTERED KEY index_name(column_name1);
```

Example

Create a clustered index for the `customer_id` and `id_number` columns in the customer table.

```
ALTER TABLE adb_demo.customer ADD CLUSTERED KEY c_k(customer_id, id_number);
```

Delete a clustered index**Syntax**

```
ALTER TABLE db_name.table_name DROP CLUSTERED KEY index_name;
```

Example

Delete the clustered index of the customer table.

```
ALTER TABLE adb_demo.customer DROP CLUSTERED KEY c_k;
```

14.11.2.4. CREATE VIEW

You can execute the `CREATE VIEW` statement to create a view.

Syntax

```
CREATE VIEW view_name AS select_stmt
```

Parameters

- `view_name` : the name of the view to create. You can prefix the view name with the database name to distinguish views with the same name in different databases.
- `select_stmt` : the data source of the view.

Example

Create a view named `v` and set its data source to the `customer` table.

```
CREATE VIEW adb_demo.v AS SELECT * FROM customer;
```

14.11.2.5. DROP DATABASE

You can execute the `DROP DATABASE` statement to delete a database.

Syntax

```
DROP DATABASE db_name;
```

 **Note** Before you delete a database, you must first delete the tables in the database.

Example

```
use adb_demo2;
+-----+
show tables;
+-----+
|Tables_in_adb_demo2  |
+-----+
| test2                |
+-----+
drop table test2;
drop database adb_demo2;
```

14.11.2.6. DROP TABLE

You can execute the `DROP TABLE` statement to delete a table.

Syntax

```
DROP TABLE db_name.table_name;
```

 **Note** When you execute this statement, both the table data and schema are deleted.

Example

```
DROP TABLE adb_demo.customer;
```

14.11.2.7. DROP VIEW

You can execute the DROP VIEW statement to delete a view.

Syntax

```
DROP VIEW [IF EXISTS] view_name, [, view_name] ...
```

Parameters

`view_name` : the name of the view to delete. You can prefix the view name with the database name to distinguish views with the same name in different databases.

Example

Delete the view named v.

```
DROP VIEW v;
```

14.11.3. Data manipulation statements

14.11.3.1. INSERT INTO

You can execute the `INSERT INTO` statement to insert data to a table. If a primary key is the same as an existing one in the records, the new record is not inserted. This statement is equivalent to the `INSERT IGNORE INTO` statement.

Syntax

```
INSERT [IGNORE]
  INTO table_name
  [( column_name [, ...] )]
  [VALUES]
  [(value_list[, ...])]
  [query];
```

Parameters

- `IGNORE` : optional. Specifies that a new record is not inserted if the primary key of the record is the same as that of an existing record.
- `column_name` : optional. The name of the column.
- `query` : inserts one or more records queried from another table to this table.

Precautions

If the column names are not specified, the sequence of the columns in the data to be inserted must be the same as that specified in the CREATE TABLE statement.

Example

Create the customer and courses tables.

```
CREATE TABLE customer (
  customer_id bigint NOT NULL COMMENT 'Customer ID',
  customer_name varchar NOT NULL COMMENT 'Customer name',
  phone_num bigint NOT NULL COMMENT 'Phone number',
  city_name varchar NOT NULL COMMENT 'City',
  sex int NOT NULL COMMENT 'Gender',
  id_number varchar NOT NULL COMMENT 'ID card number',
  home_address varchar NOT NULL COMMENT 'Home address',
  office_address varchar NOT NULL COMMENT 'Office address',
  age int NOT NULL COMMENT 'Age',
  login_time timestamp NOT NULL COMMENT 'Logon time',
  PRIMARY KEY (login_time,customer_id,phone_num)
)DISTRIBUTED BY HASH(customer_id)
PARTITION BY VALUE(DATE_FORMAT(login_time, '%Y%m%d')) LIFECYCLE 30
COMMENT 'Customer information table';
```

```
CREATE TABLE courses(
  id bigint AUTO_INCREMENT PRIMARY KEY,
  name varchar(20) NOT NULL,
  grade varchar(20) default 'Grade 3',
  submission_date timestamp)DISTRIBUTED BY HASH(id)
```

- Insert a record to the customer table.

```
INSERT INTO customer(customer_id,customer_name,phone_num,city_name,sex,id_number,home_address,office_address,age,login_time)
values
(002367,'Alan','13678973421','Hangzhou',0,'987300','West Lake','Cloud Town',23,'2018-03-02 10:00:00');
```

- Insert multiple records to the customer table.

```
INSERT INTO customer(customer_id,customer_name,phone_num,city_name,sex,id_number,home_address,office_address,age,login_time)
values
customer values(002367,'Tom','13678973421','Hangzhou',0,'987300','West Lake','Cloud Town',23,'2018-03-02 10:00:00'),(
002368,'Alex','13878971234','Hangzhou',0,'987300','West Lake','Cloud Town',28,'2018-08-01 11:00:00'),(002369,'Eric','1396
8075284','Hangzhou',1,'987300','West Lake','Cloud Town',35,'2018-09-12 08:11:00');
```

- Insert multiple records to the customer table without specifying the column names.

```
INSERT INTO customer
values(002367,'Tom','13678973421','Hangzhou',0,'987300','West Lake','Cloud Town',23,'2018-03-02 10:00:00'),(002368,'Al
ex','13878971234','Hangzhou',0,'987300','West Lake','Cloud Town',28,'2018-08-01 11:00:00'),(002369,'Eric','13968075284','H
angzhou',1,'987300','West Lake','Cloud Town',35,'2018-09-12 08:11:00');
```

- Insert a record to the courses table.

```
insert into courses (name,submission_date) values("Jams",NOW());
```

14.11.3.2. REPLACE INTO

You can execute the `REPLACE INTO` statement to insert data to a table by overwriting the existing data in real time. The system first checks whether the primary key of a record to be inserted is the same as that of an existing record. If they are the same, the system deletes the existing record and inserts the new record. Otherwise, the system only inserts the new record.

Syntax

```
REPLACE INTO table_name [(column_name,...)] VALUES ((Constant|NULL|DEFAULT),...),(...),...
```

Example

- Insert a record to the customer table by specifying the column names in the `REPLACE INTO` statement.

```
REPLACE INTO customer(customer_id,customer_name,phone_num,city_name,sex,id_number,home_address,office_address,age,login_time) values (002367,'Alan','13678973421','Hangzhou','0','987300','West Lake','Cloud Town','23','2018-03-02 10:00:00');
```

- Insert multiple records to the customer table without specifying the column names.

```
REPLACE INTO customer values(002367,'Tom','13678973421','Hangzhou','0','987300','West Lake','Cloud Town','23','2018-03-02 10:00:00'),(002368,'Alex','13878971234','Hangzhou','0','987300','West Lake','Cloud Town','28','2018-08-01 11:00:00'),(002369,'Eric','13968075284','Hangzhou','1','987300','West Lake','Cloud Town','35','2018-09-12 08:11:00');
```

14.11.3.3. INSERT SELECT FROM

You can execute the `INSERT SELECT FROM` statement to copy records from one table to another.

Syntax

```
INSERT INTO table_name [( column_name [, ...] )]query;
```

Parameters

- `column_name` : the name of the column. If you want to copy data in only some columns of the source table to the target table, the columns specified in the `SELECT` clause must have the same sequence and data types as those specified in the `INSERT` clause.
- `query` : the `SELECT FROM TABLE` or `SELECT FROM VIEW` statement.

Example

- Copy data only in the specified columns of the customer table to the `new_customer` table by specifying the column names.

```
INSERT INTO new_customer (customer_id, customer_name, phone_num)
SELECT customer_id, customer_name, phone_num FROM customer
WHERE customer.customer_name = 'Alan';
```

- Copy data in all columns of the customer table to the `new_customer` table and do not specify the column names.

```
INSERT INTO new_customer
SELECT customer_id,customer_name,phone_num,city_name,sex,id_number,home_address,office_address,age,login_
time)
FROM customer
WHERE customer.customer_name = 'Alan';
```

14.11.3.4. REPLACE SELECT FROM

You can execute the `REPLACE SELECT FROM` statement to copy records from one table to another by overwriting existing data in real time. The system first checks whether the primary key of a record to be inserted is the same as that of an existing record. If they are the same, the system deletes the existing record and inserts the new record. If they are not the same, the system inserts the new record.

Syntax

```
REPLACE INTO table_name [(column_name,...)]query;
```

Parameter

- `query` : the `SELECT FROM TABLE` or `SELECT FROM VIEW` statement.
- `column_name` : the name of the column. If you want to copy data in only some columns of the source table to the target table, the columns specified in the `SELECT` clause must have the same sequence and data types as those specified in the `REPLACE` clause.

Precautions

The target table to which the records are inserted must exist before you execute the `REPLACE SELECT FROM` statement.

Example

Copy data only in the specified columns of the `customer` table to the `new_customer` table by specifying the column names.

```
REPLACE INTO new_customer (customer_id, customer_name, phone_num)
SELECT customer_id, customer_name, phone_num
FROM customer
WHERE customer.customer_name = 'Alan';
```

14.11.3.5. INSERT OVERWRITE INTO SELECT

You can execute the `INSERT OVERWRITE INTO SELECT` statement to insert multiple records to a table at a time.

Syntax

```
INSERT OVERWRITE INTO table_name [(column_name,...)]
SELECT select_statement FROM from_statement
```

Precautions

- The target table to which the records are inserted must exist before you execute the `INSERT OVERWRITE INTO SELECT` statement.

- The existing data in the target table will not change before the `INSERT OVERWRITE INTO SELECT` statement is completed. The system writes data to the target table after the `INSERT OVERWRITE INTO SELECT` statement is completed.
- If you execute the `INSERT OVERWRITE INTO SELECT` statement to insert a primary key that is the same as an existing one in the records, the new record is not inserted.

14.11.3.6. UPDATE

You can execute the `UPDATE` statement to update data in a table.

Syntax

```
UPDATE table_reference
SET assignment_list
[WHERE where_condition]
[ORDER BY ...]
```

Precautions

The table on which you execute the `UPDATE` statement must have a primary key.

Example

Change the name of the customer who has the `customer_id = '2369'` attribute to Claire in the customer table.

```
update customer set customer_name = 'Claire' where customer_id = '2369';
```

14.11.3.7. DELETE

You can execute the `DELETE` statement to delete records from a table.

Syntax

```
DELETE FROM table_name[ WHERE condition ]
```

Precautions

- The table on which you execute the `DELETE` statement must have a primary key.
- You cannot use the alias of a table to execute the `DELETE` statement.
- If you need to delete all records of a table or all partitions, we recommend that you use the `TRUNCATE TABLE` or `TRUNCATE PARTITION` statement, instead of the `DELETE` statement.

Example

- Delete the record where `name` is Alex from the customer table.

```
DELETE FROM customer WHERE customer_name='Alex';
```

- Delete the records where age is less than 18 from the customer table.

```
DELETE FROM customer WHERE age<18;
```

14.11.3.8. TRUNCATE TABLE

You can execute the `TRUNCATE TABLE` statement to clear the data of a table or specified partitions in a table.

Syntax

- Clear the data of a table.

```
TRUNCATE TABLE db_name.table_name;
```

- Clear the data of specified partitions in a table.

```
TRUNCATE TABLE db_name.table_name PARTITION partition_name;
```

- The data type of partition names is `BIGINT`. You can execute the following SQL statement to obtain the names of all partitions in a table:

```
select partition_name from information_schema.partitions where table_name = 'your_table_name' order by partition_name desc limit 100;
```

Precautions

When you execute the `TRUNCATE TABLE` statement to clear the data of a table, the table schema is not deleted.

Example

- Clear the data of the customer table.

```
TRUNCATE TABLE adb_demo.customer;
```

- Clear the data of specified partitions in the customer table.

```
TRUNCATE TABLE adb_demo.customer partition 20170103,20170104,20170108
```

14.11.3.9. KILL PROCESS

You can execute the `KILL PROCESS` statement to terminate a running process.

Syntax

```
KILL PROCESS process_id
```

Parameters

`process_id` : the ID of the process to terminate. You can query the process ID by executing the `SHOW PROCESSLIST` statement and checking the `Processid` field.

Permission

- You can execute the `KILL PROCESS` statement to terminate the running processes under your account.
- The privileged account can grant the `PROCESS` permission to a standard account by using the `GRANT` statement. The standard account can then terminate the running processes of all accounts in the cluster.

```
GRANT process on *.* to account_name;
```

14.11.3.10. SHOW PROCESSLIST

You can execute the `SHOW PROCESSLIST` statement to view the running processes.

Note You can also execute the `INFORMATION_SCHEMA PROCESSLIST` statement to view the running processes.

Syntax

```
SHOW [FULL] PROCESSLIST
```

Response parameters

The following response parameters are returned after you execute the `SHOW FULL PROCESSLIST` or `SHOW PROCESSLIST` statement:

- **Id**: the ID of the process.
- **Processid** : the unique ID of the task. This parameter is required when you execute the `KILL PROCESS` statement.
- **User** : the current account.
- **Host** : the hostname of the client that sends the `SHOW PROCESSLIST` statement, which consists of the IP address and port number.
- **DB** : the database to which the process is connected.
- **Command** : the command that is executed in the current connection. The command types include `sleep` , `query` , and `connect` .
- **Time** : the time when the `Command` is executed. Unit: seconds.
- **State** : the execution status of the SQL statement in the current connection.
- **Info** : the SQL statement.

Note If you do not specify the `FULL` keyword, you can view only the first 100 characters of the `Info` field in each record.

Permission

- You can execute the `SHOW PROCESSLIST` statement to view the processes running under your account.
- A privileged account can grant the `PROCESS` permission to a standard account by using the `GRANT` statement. The standard account can then view the running processes of all accounts in the cluster.

```
GRANT process on *.* to account_name;
```

14.11.4. SELECT

14.11.4.1. Syntax

You can use the `SELECT` statement to query data from one or more tables. The syntax is as follows:

```
[ WITH with_query [, ...] ]
SELECT
[ ALL | DISTINCT ] select_expr [, ...]
[ FROM table_reference [, ...] ]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition]
[ WINDOW window_name AS (window_spec) [, window_name AS (window_spec)] ...]
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
[ ORDER BY {column_name | expr | position} [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

- `table_reference` : the data source from which data is queried. The source can be a table, view, associative table, or subquery.
- Table names and column names are not case-sensitive.
- If a table name or column name contains keywords or space characters, you can quote the table name or column name with backticks (` `).

WHERE

You can enter a `BOOLEAN` expression in the `WHERE` keyword to query data records that meet the specified condition. For example, you can execute the following statement to query the information of the customer whose `customer_id` is 2368.

```
SELECT * FROM CUSTOMER where customer_id=2368;
```

ALL and DISTINCT

You can use the `ALL` and `DISTINCT` keywords to specify whether duplicate rows are retained in the query result. The default value is `ALL`, indicating that all rows are returned. The `DISTINCT` keyword indicates that duplicate rows are deleted in the query result.

```
SELECT col1, col2 FROM t1;SELECT DISTINCT col1, col2 FROM t1;
```

The following methods show you how to use other keywords.

14.11.4.2. WITH

This topic describes how to use the `WITH` clause in a `SELECT` statement.

You can use the `WITH` clause to define subqueries by using common table expressions (CTEs) in a `SELECT` statement. The subqueries defined in the `WITH` clause can be referenced in the `SELECT` statement. The `WITH` clause can be used to flatten nested queries or simplify subqueries. Each subquery is performed only once in the `SELECT` statement. This improves the query performance.

Note

- A CTE is a named temporary result set, and is valid only in a single SQL statement, such as a `SELECT`, `INSERT`, and `DELETE` statement.
- A CTE is valid only during the period when the SQL statement is executing.

Precautions

- A CTE can be followed by a SQL statement, such as a SELECT, INSERT, or UPDATE statement, or other CTEs in the same WITH clause. Separate multiple CTEs with commas (,). Otherwise, the CTEs will not take effect.
- The paging feature is not supported in a CTE.

Use the WITH clause

- The following queries are equivalent:

```
SELECT a, b FROM (SELECT a, MAX(b) AS b FROM t GROUP BY a) AS x;
```

```
WITH x AS (SELECT a, MAX(b) AS b FROM t GROUP BY a) SELECT a, b FROM x;
```

- The WITH clause can be used to define multiple subqueries.

```
WITH
  t1 AS (SELECT a, MAX(b) AS b FROM x GROUP BY a),
  t2 AS (SELECT a, AVG(d) AS d FROM y GROUP BY a)
SELECT t1.*, t2.*
FROM t1 JOIN t2 ON t1.a = t2.a;
```

- After a subquery is defined, other subqueries following this subquery in the same WITH clause can reference this subquery.

```
WITH
  x AS (SELECT a FROM t),
  y AS (SELECT a AS b FROM x),
  z AS (SELECT b AS c FROM y)
SELECT c FROM z;
```

14.11.4.3. GROUP BY

You can use the `GROUP BY` clause to group the query result. You can use the `GROUPING SETS`, `CUBE`, and `ROLLUP` options in the `GROUP BY` clause to display the grouping result in different forms.

Syntax

```
GROUP BY expression [, ...]
```

- **GROUPING SETS**

The `GROUPING SETS` option is used to specify multiple `GROUP BY` conditions for one query, which is equal to the `UNION` of multiple `GROUP BY` clauses.

```
SELECT origin_state, origin_zip, destination_state, sum(package_weight)
FROM shipping
GROUP BY GROUPING SETS (
  (origin_state),
  (origin_state, origin_zip),
  (destination_state));
```

The preceding statement is equal to the following statement:

```
SELECT origin_state, NULL, NULL, sum(package_weight)
FROM shipping GROUP BY origin_state
UNION ALL
SELECT origin_state, origin_zip, NULL, sum(package_weight)
FROM shipping GROUP BY origin_state, origin_zip
UNION ALL
SELECT NULL, NULL, destination_state, sum(package_weight)
FROM shipping GROUP BY destination_state;
```

- **CUBE**

The `CUBE` option is used to list all possible grouping sets.

```
SELECT origin_state, destination_state, sum(package_weight)
FROM shipping
GROUP BY origin_state, destination_state WITH CUBE
```

The preceding statement is equal to the following statement:

```
SELECT origin_state, destination_state, sum(package_weight)
FROM shipping
GROUP BY GROUPING SETS (
  (origin_state, destination_state),
  (origin_state),
  (destination_state),
  ())
```

- **ROLLUP**

The `ROLLUP` option is used to list the grouping sets in a hierarchical manner.

```
SELECT origin_state, origin_zip, sum(package_weight)
FROM shipping
GROUP BY ROLLUP (origin_state, origin_zip)
```

The preceding statement is equal to the following statement:

```
SELECT origin_state, origin_zip, sum(package_weight)
FROM shipping
GROUP BY GROUPING SETS ((origin_state, origin_zip), (origin_state), ())
```

Precautions

- You must use standard aggregate functions such as `SUM`, `AVG`, and `COUNT` to specify the columns that are not used for grouping. Otherwise, the `GROUP BY` clause does not take effect.
- The `GROUP BY` clause must contain all the columns and non-aggregate expressions specified for the `SELECT` statement.

Example

The following statement contains two aggregate expressions. The first aggregate expression uses the `SUM` function, and the second uses the `COUNT` function. The `LISTID` and `EVENTID` columns are the columns that are used for grouping.

```

select listid, eventid, sum(pricepaid) as revenue,
count(qtysold) as numtix
from sales
group by listid, eventid
order by 3, 4, 2, 1
limit 5;
listid | eventid | revenue | numtix
-----+-----+-----+-----
89397 | 47 | 20.00 | 1
106590 | 76 | 20.00 | 1
124683 | 393 | 20.00 | 1
103037 | 403 | 20.00 | 1
47685 | 429 | 20.00 | 1
(5 rows)

```

You can also use the sequence numbers to reference columns in the **GROUP BY** clause.

For example, you can modify the preceding statement as follows:

```

select listid, eventid, sum(pricepaid) as revenue,
count(qtysold) as numtix
from sales
group by 1,2
order by 3, 4, 2, 1
limit 5;
listid | eventid | revenue | numtix
-----+-----+-----+-----
89397 | 47 | 20.00 | 1
106590 | 76 | 20.00 | 1
124683 | 393 | 20.00 | 1
103037 | 403 | 20.00 | 1
147685 | 429 | 20.00 | 1

```

14.11.4.4. HAVING

You can use the **HAVING** clause with the **GROUP BY** clause and aggregate functions to display groups that only meet certain conditions after grouping and aggregation.

Syntax

```
[ HAVING condition ]
```

Precautions

- The columns that you reference in the **HAVING** clause must be used for grouping or contain an aggregate function.
- The **HAVING** clause must be used together with aggregate functions and the **GROUP BY** clause to display groups that only meet certain conditions after grouping based on **GROUP BY**.

Example

Group records in the CUSTOMER table and query the records whose account balance is greater than the specified value.

```
SELECT count(*), mktsegment, nationkey,
       CAST(sum(acctbal) AS bigint) AS totalbal
FROM customer
GROUP BY mktsegment, nationkey
HAVING sum(acctbal) > 5700000
ORDER BY totalbal DESC;
_col0 | mktsegment | nationkey | totalbal
```

```
-----+-----+-----+-----
1272 | AUTOMOBILE | 19 | 5856939
1253 | FURNITURE  | 14 | 5794887
1248 | FURNITURE  | 9  | 5784628
1243 | FURNITURE  | 12 | 5757371
1231 | HOUSEHOLD  | 3  | 5753216
1251 | MACHINERY  | 2  | 5719140
1247 | FURNITURE  | 8  | 5701952
```

14.11.4.5. JOIN

Syntax

```
join_table:
  table_reference [INNER] JOIN table_factor [join_condition]
| table_reference {LEFT|RIGHT|FULL} [OUTER] JOIN table_reference join_condition
| table_reference CROSS JOIN table_reference [join_condition])

table_reference:
  table_factor
| join_table

table_factor:
  tbl_name [alias]
| table_subquery alias
| ( table_references )

join_condition:
  ON expression
```

Example

```

select catgroup1, sold, unsold
from
(select catgroup, sum(qtysold) as sold
from category c, event e, sales s
where c.catid = e.catid and e.eventid = s.eventid
group by catgroup) as a(catgroup1, sold)
join
(select catgroup, sum(numtickets)-sum(qtysold) as unsold
from category c, event e, sales s, listing l
where c.catid = e.catid and e.eventid = s.eventid
and s.listid = l.listid
group by catgroup) as b(catgroup2, unsold)
on a.catgroup1 = b.catgroup2
order by 1;

```

14.11.4.6. LIMIT

You can use the **LIMIT** clause to specify the maximum number of rows to return in a query. Typically, you must specify one or two numbers in the **LIMIT** clause. The first number specifies the number of rows to skip from the beginning, and the second number specifies the maximum number of rows to return.

Example

Query the orders table and limit the number of rows to return to five by using the **LIMIT** clause.

```

SELECT orderdate FROM orders LIMIT 5;
-----
o_orderdate
-----
1996-04-14
1992-01-15
1995-02-01
1995-11-12
1992-04-26

```

Query the customer table and return the information of the third customer to the seventh customer sorted by the creation date.

```

SELECT * FROM customer ORDER BY create_date LIMIT 2,5;

```

14.11.4.7. ORDER BY

You can use the **ORDER BY** clause to sort the query result. Each expression in the **ORDER BY** clause consists of column names or the sequence numbers (starting from 1) of columns.

Syntax

```
[ ORDER BY expression
 [ ASC | DESC ]
 [ NULLS FIRST | NULLS LAST ]
 [ LIMIT { count | ALL } ]
```

14.11.4.8. Subqueries

The following example shows how to query the top ten sellers in ticket sales. The WHERE clause defines a table subquery, and the result of the subquery consists of multiple rows and one column.

 **Note** The result of a table subquery can contain multiple rows and columns.

```
select firstname, lastname, cityname, max(qtysold) as maxsold
from users join sales on users.userid=sales.sellerid
where users.city not in(select venuecity from venue)
group by firstname, lastname, city
order by maxsold desc, city desc
limit 10;
```

```
firstname | lastname | cityname | maxsold
-----+-----+-----+-----
Noah      | Guerrero | Worcester | 8
Isadora   | Moss     | Winooski  | 8
Kieran    | Harrison | Westminster | 8
Heidi     | Davis    | Warwick   | 8
Sara      | Anthony  | Waco      | 8
Bree      | Buck     | Valdez    | 8
Evangeline | Sampson  | Trenton   | 8
Kendall   | Keith    | Stillwater | 8
Bertha    | Bishop   | Stevens Point | 8
Patricia  | Anderson | South Portland | 8
```

14.11.4.9. UNION, INTERSECT, and EXCEPT

You can use the UNION, INTERSECT, and EXCEPT operators to combine multiple query result sets to a single result set.

Syntax

```
query
{ UNION [ ALL ] | INTERSECT | EXCEPT | MINUS }
query
```

Parameters

- **UNION** : returns the union of two result sets.
- **UNION ALL** : returns the union of two result sets where duplicate rows are retained. The **ALL** keyword indicates that the duplicate rows are retained after the **UNION** calculation.

- **INTERSECT** : returns the intersection of two result sets.
- **EXCEPT | MINUS** : returns rows that appear in the first result set, but not in the other result set. The **MINUS** operator is equivalent to the **EXCEPT** operator.

Calculation sequence

- The **UNION** and **EXCEPT** operators are left-associative. That is, if you do not use parentheses () to change the calculation sequence, the calculation starts from left to right.

For example, in the following statement, the **UNION** operator returns the union of T1 and T2. Then the **EXCEPT** operator returns the rows that appear only in the union returned by the **UNION** operator, but not T3.

```
select * from t1
union
select * from t2
except
select * from t3
order by c1;
```

- In the same statement, the **INTERSECT** operator is prioritized over the **UNION** and **EXCEPT** operators.

For example, the following statement finds the intersection of T2 and T3, and then the union of the intersection and T1.

```
select * from t1
union
select * from t2
intersect
select * from t3
order by c1;
```

- You can use parentheses () to change the calculation sequence of these operators.

For example, the following statement finds the union of T1 and T2, and then the intersection of the union and T3.

```
(select * from t1
union
select * from t2)
intersect
(select * from t3)
order by c1;
```

14.11.5. CREATE USER

You can execute the **CREATE USER** statement to create an account.

Syntax

```
CREATE USER
[if not exists] user [auth_option] [, [if not exists] user [auth_option]] ...
```

Precautions

To create an account by executing the CREATE USER statement, you must have the `CREATE_USER` permission.

Examples

Create an account named account2 and set the password to Account2.

```
CREATE USER if not exists 'account2' IDENTIFIED BY 'Account2';
```

14.11.6. GRANT

You can execute the GRANT statement to grant permissions to an account.

Syntax

```
GRANT
  priv_type [(column_list)]
  [, priv_type [(column_list)]] ...
ON priv_level
TO user [auth_option]
[WITH {GRANT OPTION}]
```

Parameters

- `priv_type` : the type of permission to grant.
- `column_list` : optional. If the `priv_type` parameter is set to `SELECT`, you can enter the names of columns to grant the `SELECT` permission on these columns.
- `priv_level` : the level of the permission to grant.
 - `*.*` : the cluster level.
 - `db_name.*` : the database level.
 - `db_name.table_name` or `table_name` : the table level.

Precautions

To grant permissions to the other accounts by executing the `GRANT` statement, you must have the `GRANT OPTION` permission.

Examples

- Grant the cluster-level `all` permission to account2.

```
GRANT all ON *.* TO 'account2';
```

- Grant the database-level `all` permission to account3.

```
GRANT all ON adb_demo.* TO 'account3';
```

- Execute the `GRANT` statement to create and grant permissions to an account. For example, create an account with the cluster-level data manipulation permissions.

```
GRANT insert,select,update,delete on *.* to 'test'@%' identified by 'Testpassword1';
```

Create an account with the database-level data manipulation permissions.

```
GRANT insert,select,update,delete on adb_demo.* to 'test123' identified by 'Testpassword123';
```

- Create an account that has the `SELECT` permission on the specified columns.

```
GRANT select (customer_id, sex) ON customer TO 'test321' identified by 'Testpassword321';
```

14.11.7. REVOKE

You can execute the REVOKE statement to revoke permissions from an account.

Syntax

```
REVOKE
  priv_type [(column_list)]
  [, priv_type [(column_list)]] ...
  ON [object_type] priv_level
  FROM user
```

Parameters

- `priv_type` : the type of permission to revoke.
- `column_list` : optional. If the `priv_type` parameter is set to `SELECT`, you can enter the names of columns to revoke the `SELECT` permission on these columns.
- `priv_level` : the level of the permission to revoke.
 - `*, *` : the cluster level.
 - `db_name.*` : the database level.
 - `db_name.table_name` or `table_name` : the table level.

Precautions

To revoke permissions from other accounts by executing the REVOKE statement, you must have the `GRANT OPTION` permission.

Examples

Revoke the database-level `all` permission of account3.

```
REVOKE all ON adb_demo.* FROM 'account3';
```

14.11.8. Query users

AnalyticDB for MySQL is compatible with MySQL databases. AnalyticDB for MySQL provides a built-in database named MySQL, which stores user information, permission information, and stored procedures of AnalyticDB for MySQL. You can execute a SELECT statement to query user information of AnalyticDB for MySQL.

Precautions

- In AnalyticDB for MySQL, only the privileged account can query user information.
- The privileged account in AnalyticDB for MySQL is equivalent to the root account in MySQL.

Example

```
USE MYSQL;
SELECT User, Host, Password FROM mysql.user;
+-----+-----+-----+
| User | Host| Password |
+-----+-----+-----+
| account1| % | *61f3777f02386598cd**** |
| account2| % | *0fe79c07e168cab99**** |
```

14.11.9. RENAME USER

You can execute the RENAME USER statement to modify the name of an account.

Syntax

```
RENAME USER old_user TO new_user [, old_user TO new_user] ...
```

Examples

```
RENAME USER account2 TO account_2;
SELECT User, Host, Password FROM mysql.user;
+-----+-----+-----+
| User | Host| Password |
+-----+-----+-----+
| account2| % | *61f3777f02386598cda**** |
| account_2| % | *0fe79c07e168cab99b**** |
```

14.11.10. DROP USER

You can execute the DROP USER statement to delete an account.

Syntax

```
DROP USER [if exists] user [, [if exists] user] ...
```

Precautions

To delete an account by executing the DROP USER statement, you must have the `CREATE_USER` permission.

Examples

```
DROP USER account_2;
```

14.11.11. SHOW

SHOW DATABASES

You can execute the SHOW DATABASES statement to view the databases.

Syntax

```
SHOW DATABASES [EXTRA];
```

You can specify the `EXTRA` parameter to view extra information about the databases, such as the IDs of the creators and the connection information.

Examples

```
SHOW DATABASES EXTRA;
+-----+
| Database      |
+-----+
| adb_demo      |
| MYSQL         |
| INFORMATION_SCHEMA |
| adb_demo2     |
```

SHOW TABLES

Syntax

You can execute the `SHOW TABLES` statement to view the tables in the current database.

```
SHOW TABLES [IN db_name];
```

Examples

```
SHOW TABLES IN adb_demo;
+-----+
| Tables_in_adb_demo |
+-----+
| customer           |
| customer2          |
| new_customer       |
| test_table         |
| v                  |
```

SHOW COLUMNS

You can execute the `SHOW COLUMNS` statement to view the columns in a table.

Syntax

```
SHOW COLUMNS IN db_name.table_name;
```

Examples

```
SHOW COLUMNS IN adb_demo.test_table;
```

SHOW CREATE TABLE

You can execute the `SHOW CREATE TABLE` statement to view the statement that is used to create a table.

Syntax

```
SHOW CREATE TABLE db_name.table_name;
```

Examples

```
SHOW CREATE TABLE adb_demo.customer;
```

SHOW GRANTS

You can execute the `SHOW GRANTS` statement to view the permissions that are granted to the current account.

Syntax

```
SHOW GRANTS;
```

14.12. System functions

This topic describes the system functions supported by AnalyticDB for MySQL.

14.12.1. Aggregate functions

The aggregate functions described in this topic use the `testtable` table as test data.

```
create table testtable(a int) distributed by hash(a);
```

```
insert into testtable values (1),(2),(3);
```

14.12.1.1. AVG

This function calculates the average value of all input values.

```
AVG(bigint x)
```

```
AVG(double x)
```

```
AVG(float x)
```

- Return value type: `DOUBLE`.
- Example:

```
select avg(a) from testtable;
+-----+
| avg(a) |
+-----+
|  2.0  |
```

14.12.1.2. BIT_AND

This function returns the result of bitwise `AND` of all bits in the parameter.

```
BIT_AND(float x)
BIT_AND(bigint x)
BIT_AND(double x)
```

- Return value type: BIGINT.
- Example:

```
select bit_and(a) from testtable;
+-----+
| bit_and(a) |
+-----+
|    0 |
```

14.12.1.3. BIT_OR

This function returns the result of bitwise OR of all bits in the parameter.

```
BIT_OR(float x)
BIT_OR(bigint x)
BIT_OR(double x)
```

- Return value type: BIGINT.
- Example:

```
select bit_or(a) from testtable;
+-----+
| bit_or(a) |
+-----+
|    3 |
```

14.12.1.4. BIT_XOR

This function returns the result of bitwise XOR of all bits in the parameter.

```
BIT_XOR(double x)
BIT_XOR(bigint x)
bit_xor(float x)
```

- Return value type: BIGINT.
- Example:

```
select bit_xor(a) from testtable;
+-----+
| bit_xor(a) |
+-----+
|    0 |
```

14.12.1.5. COUNT

This function counts the number of records.

```
COUNT([distinct|all] value x)
```

- **Description:** `distinct` and `all` specify whether to exclude duplicate records in the counting process. The default value is `all`, indicating that all records are counted. If `distinct` is specified, only records with distinct values are counted.
- **Return value type:** `BIGINT`.
- **Example:**

```
select count(distinct a) from testtable;
+-----+
| count(DISTINCT a) |
+-----+
|          3 |
```

14.12.1.6. MAX

This function returns the maximum value of all input values.

```
MAX(value x)
```

- **Description:** value can be of any data type. However, data of the `BOOLEAN` type is not supported in the calculation. If the value for a row in the specified column is null, this row is ignored.
- **Return value type:** `LONG`.
- **Example:**

```
select max(a) from testtable;
+-----+
| max(a) |
+-----+
|    3 |
```

14.12.1.7. MIN

This function returns the minimum value of all input values.

```
MIN(value x)
```

- **Description:** value can be of any data type. However, data of the `BOOLEAN` type is not supported in the calculation. If the value for a row in the specified column is null, this row is ignored.
- **Return value type:** `LONG`.
- **Example:**

```
select min(a) from testtable;
+-----+
| min(a) |
+-----+
|    1 |
```

14.12.1.8. STD/STDDEV

This function returns the sample standard deviation of all input values.

```
STD(double x)
STD(bigint x)
STDDEV(double x)
STDDEV(bigint x)
```

- Return value type: DOUBLE.
- Example:

```
select std(a) from testtable;
+-----+
| std(a) |
+-----+
| 0.816496580927726 |
```

14.12.1.9. STDDEV_POP

This function returns the population standard deviation of all input values.

```
STDDEV_POP(double x)
STDDEV_POP(bigint x)
```

- Return value type: DOUBLE.
- Example:

```
select stddev_pop(a) from testtable;
+-----+
| stddev_pop(a) |
+-----+
| 0.816496580927726 |
```

14.12.1.10. STDDEV_SAMP

This function returns the population standard deviation for a group of integers, decimals, or floating-point numbers.

```
STDDEV_SAMP(double x)
STDDEV_SAMP(bigint x)
```

- Return value type: DOUBLE.
- Example:

```
select stddev_samp(a) from testtable;
+-----+
| stddev_samp(a) |
+-----+
| 1.0            |
```

14.12.1.11. SUM

This function calculates the sum of all input values.

```
SUM(double x)
SUM(float x)
SUM(bigint x)
```

- Return value type: BIGINT.
- Example:

```
select sum(a) from testtable;
+-----+
| sum(a) |
+-----+
| 6      |
```

14.12.1.12. VAR_POP

This function returns the population variance for a group of integers, decimals, or floating-point numbers.

```
VAR_POP(double x)
VAR_POP(bigint x)
```

- Description: You can also use the `VARIANCE()` function, which is equivalent to the `VAR_POP` function. However, the `VARIANCE()` function is not a standard SQL function. If no matches are found, `VAR_POP()` returns the value `NULL`.
- Return value type: DOUBLE.
- Example:

```
select var_pop(a) from testtable;
+-----+
| var_pop(a) |
+-----+
| 0.6666666666666666 |
```

14.12.1.13. VAR_SAMP

This function returns the sample variance for a group of integers, decimals, or floating-point numbers.

```
VAR_SAMP(double x)
VAR_SAMP(bigint x)
```

- Return value type: DOUBLE.
- Example:

```
select var_samp(a) from testtable;
+-----+
|  var_samp(a)  |
+-----+
|      1.0      |
```

14.12.1.14. VARIANCE

This function returns the population variance for a group of integers, decimals, or floating-point numbers.

```
VARIANCE(double x)
VARIANCE(bigint x)
```

- Description: The `VARIANCE()` function is an extension to standard SQL and can be replaced by the standard SQL function `VAR_POP()`. If no matches are found, `VARIANCE()` returns the value NULL.
- Return value type: DOUBLE.
- Example:

```
select variance(a) from testtable;
+-----+
|  variance(a)  |
+-----+
| 0.6666666666666666  |
```

14.12.2. Date and time functions

14.12.2.1. ADDDATE/DATE_ADD

This function adds an interval to a specified date.

```
ADDDATE(date,INTERVAL expr unit)
ADDDATE(expr,days)
```

- Description:
 - Valid values of unit: second, minute, hour, day, month, year, minute_second, hour_second, hour_minute, day_second, day_minute, day_hour, and year_month. Default value of unit: day.
 - days and expr: This function returns the value of expr plus days.
- Parameter types:

```

adddate(date, INTERVAL expr unit)
adddate(timestamp, INTERVAL expr unit)
adddate(datetime, INTERVAL expr unit)
adddate(varchar, INTERVAL expr unit)
adddate(date, varchar)
adddate(date, bigint)
adddate(datetime, bigint)
adddate(datetime, varchar)
adddate(timestamp, varchar)
adddate(timestamp, bigint)
adddate(varchar, bigint)
adddate(varchar, varchar)
    
```

- Return value type: DATE.
- Example:

```

select adddate(date '2001-1-22',interval '3' day);
+-----+
| adddate(DATE '2001-1-22', INTERVAL '3' DAY) |
+-----+
| 2001-01-25          |
    
```

```

select adddate(timestamp '2001-1-22',interval '3' day);
+-----+
| adddate(TIMESTAMP '2001-1-22', INTERVAL '3' DAY) |
+-----+
| 2001-01-25 00:00:00          |
    
```

```

select adddate(datetime '2001-1-22',interval '3' day);
+-----+
| adddate(DATETIME '2001-1-22', INTERVAL '3' DAY) |
+-----+
| 2001-01-25 00:00:00          |
    
```

```

select adddate('2001-1-22',interval '3' day);
+-----+
| adddate('2001-1-22', INTERVAL '3' DAY) |
+-----+
| 2001-01-25          |
    
```

```

select adddate(datetime '2001-1-22',interval '3' second);
+-----+
| adddate(DATETIME '2001-1-22', INTERVAL '3' SECOND) |
+-----+
|                2001-01-22 00:00:03 |
    
```

14.12.2.2. ADDTIME

This function adds time to a specified time. That is, this function returns the result of expr1 plus expr2.

```
ADDTIME(expr1,expr2)
```

- Parameter types:

```
addtime(date,varchar)
addtime(time,varchar)
addtime(datetime,varchar)
addtime(timestamp,varchar)
addtime(varchar,varchar)
```

- Return value type: VARCHAR.

- Example:

```
select addtime(date '1998-01-01','01:01:01');
+-----+
| addtime(DATE '1998-01-01', '01:01:01') |
+-----+
| 1998-01-01 01:01:01          |
```

14.12.2.3. CONVERT_TZ

This function converts the value of dt from from_tz to to_tz and returns the result.

```
CONVERT_TZ(dt,from_tz,to_tz)
```

- Parameter types:

```
convert_tz(varchar, varchar, varchar)
```

- Return value type: DATETIME.

- Example:

```
select convert_tz('2004-01-01 12:00:00','+00:00','+10:00');
+-----+
| convert_tz('2004-01-01 12:00:00', '+00:00', '+10:00') |
+-----+
|                2004-01-01 22:00:00 |
```

14.12.2.4. CURDATE

This function returns the current date.

```
CURDATE()
```

- Return value type: DATE.

- Example:

```
select curdate;
+-----+
| curdate() |
+-----+
| 2019-05-25 |
```

14.12.2.5. CURTIME

This function returns the current time.

```
CURTIME()
```

- Return value type: TIME.
- Example:

```
select curtime();
+-----+
| curtime() |
+-----+
| 14:39:22.109 |
```

14.12.2.6. DATE

This function returns the date part of a date or datetime expression.

```
DATE(expr)
```

- Parameter types:

```
date(timestamp)
date(datetime)
date(varchar)
```

- Return value type: DATE.
- Example:

```
select date(timestamp '2003-12-31 01:02:03');
+-----+
| date(TIMESTAMP '2003-12-31 01:02:03') |
+-----+
| 2003-12-31 |
```

14.12.2.7. DATE_FORMAT

This function returns a date as a string in the specified format.

```
DATE_FORMAT(date,format)
```

- Description: The following table describes the format specifiers.

Specifier	Description
%a	The abbreviation of a day of a week. Valid values: Sun to Sat.
%b	The abbreviation of a month name. Valid values: Jan to Dec.
%c	The month in numeric format. Valid values: 0 to 12.
%d	The day of the month in numeric format. Valid values: 00 to 31.
%e	The day of the month in numeric format. Valid values: 0 to 31.
%f	The microseconds. Valid values: 000000 to 999999.
%H	The hour. Valid values: 00 to 23.
%h	The hour. Valid values: 01 to 12.
%I	The hour. Valid values: 01 to 12.
%i	The minutes in numeric format. Valid values: 00 to 59.
%j	The day of the year. Valid values: 001 to 366.
%k	The hour. Valid values: 0 to 23.
%l	The hour. Valid values: 1 to 12.
%M	The name of the month. Valid values: January to December.
%m	The month in numeric format. Valid values: 00 to 12.
%p	The abbreviation of a time period in 12-hour format. Valid values: AM and PM.
%r	The time in 12-hour format (hh:mm:ss AM or hh:mm:ss PM).
%S	The seconds. Valid values: 00 to 59.
%s	The seconds. Valid values: 00 to 59.
%T	The time in 24-hour format (hh:mm:ss).
%v	The week number. Monday is the first day of a week. This specifier applies to WEEK() mode 3 and is used with %x.
%W	The name of the weekday. Valid values: Sunday to Saturday.
%x	The year of the week in numeric format. Monday is the first day of a week. This specifier is used with %v, and the value contains four digits.
%Y	The year of the week in numeric format. The value contains four digits.

Specifier	Description
%y	The year of the week in numeric format. The value contains two digits.
%%	The percent sign (%).
%x	x, for any "x" not listed above.

- Parameter types:

```
date_format(timestamp, varchar)
date_format(varchar, varchar)
date_format(datetime, varchar)
date_format(date, varchar)
```

- Return value type: VARCHAR.
- Example:

```
select date_format(timestamp '2019-05-27 13:23:00', '%W %M %Y')as result;
+-----+
| result |
+-----+
| Monday May 2019 |
```

14.12.2.8. SUBDATE/DATE_SUB

This function returns the value of date minus the specified INTERVAL.

```
DATE_SUB(date,INTERVAL expr unit)
```

- Description: Valid values of unit: second, minute, hour, day, month, year, minute_second, hour_second, hour_minute, day_second, day_minute, day_hour, and year_month. Default value of unit: day.
- Parameter types:

```
subdate(date, INTERVAL expr unit)
subdate(timestamp, INTERVAL expr unit)
subdate(datetime, INTERVAL expr unit)
subdate(varchar, INTERVAL expr unit)
subdate(date, bigint)
subdate(date, varchar)
subdate(datetime, bigint)
subdate(datetime, varchar)
subdate(timestamp, bigint)
subdate(timestamp, varchar)
subdate(varchar, bigint)
subdate(varchar, varchar)
```

- Return value type: DATE.
- Example:

```
select date_sub(date '2001-1-22',interval '3' day);
+-----+
| date_sub(DATE '2001-1-22', INTERVAL '3' DAY) |
+-----+
| 2001-01-19          |
```

14.12.2.9. DATEDIFF

This function returns the number of days between expr1 and expr2.

```
DATEDIFF(expr1,expr2)
```

- Parameter types:

```
datediff(varchar, varchar)
datediff(datetime, varchar)
datediff(varchar, datetime)
datediff(datetime, datetime)
datediff(varchar, timestamp)
datediff(timestamp, timestamp)
datediff(timestamp, varchar)
datediff(date, date)
datediff(date, varchar)
datediff(varchar, date)
```

- Return value type: BIGINT.

- Example:

```
select datediff('2007-12-31 23:59:59','2007-12-30');
+-----+
| datediff('2007-12-31 23:59:59', '2007-12-30') |
+-----+
|                1 |
```

14.12.2.10. DAY/DAYOFMONTH

This function returns the day in date. Valid values: [1,31] .

```
DAY(date)
DAYOFMONTH(date)
```

- Parameter types:

```
dayofmonth(timestamp)
dayofmonth(datetime)
dayofmonth(date)
dayofmonth(time)
dayofmonth(varchar)
```

- Return value type: BIGINT.

- Example:

```
select dayofmonth(timestamp '2007-02-03 12:23:09');
+-----+
| dayofmonth(TIMESTAMP '2007-02-03 12:23:09') |
+-----+
|                3 |
```

14.12.2.11. DAYNAME

This function returns the day of the week for a date, such as Monday.

```
DAYNAME(date)
```

- Parameter types:

```
dayname(timestamp)
dayname(datetime)
dayname(date)
dayname(varchar)
```

- Return value type: VARCHAR.

- Example:

```
select dayname(timestamp '2007-02-03 00:00:00');
+-----+
| dayname(TIMESTAMP '2007-02-03 00:00:00') |
+-----+
| Saturday                |
```

14.12.2.12. DAYOFWEEK

This function returns the weekday index for a date. For example, the weekday indexes for Sunday, Monday, and Saturday are 1, 2, and 7, respectively.

```
DAYOFWEEK(date)
```

- Parameter types:

```
dayofweek(timestamp)
dayofweek(datetime)
dayofweek(date)
dayofweek(varchar)
```

- Return value type: BIGINT.

- Example:

```

select dayofweek(timestamp '2007-02-03 00:00:00');
+-----+
| dayofweek(TIMESTAMP '2007-02-03 00:00:00') |
+-----+
|                7 |

```

14.12.2.13. DAYOFYEAR

This function returns the day of the year for a date. Valid values: [1,366] .

```
DAYOFYEAR(date)
```

- Parameter types:

```

dayofyear(timestamp)
dayofyear(datetime)
dayofyear(date)
dayofyear(varchar)

```

- Return value type: BIGINT.

- Example:

```

select dayofyear(timestamp '2007-02-03 00:12:12');
+-----+
| dayofyear(TIMESTAMP '2007-02-03 00:12:12') |
+-----+
|                34 |

```

14.12.2.14. EXTRACT

This function returns a part of a date or time, which is specified by unit. For example, this function can return the year, month, day, hour, or minute of a date or time.

```
EXTRACT(unit FROM date)
```

- Valid values of unit: second, minute, hour, day, month, year, minute_second, hour_second, hour_minute, day_second, day_minute, day_hour, and year_month.
- Supported parameter types: VARCHAR, TIMESTAMP, DATETIME, and TIME.
- Return value type: BIGINT.
- Example:

```

select extract(second from '2019-07-02 00:12:34');
+-----+
| _col0 |
+-----+
|    34 |

```

14.12.2.15. FROM_DAYS

This function returns a DATE value based on the parameter N indicating the number of days.

FROM_DAYS(N)

- Parameter types:

```
from_days(varchar)
from_days(bigint)
```

- Return value type: DATE.
- Example:

```
select from_days(730669);
+-----+
| from_days(730669) |
+-----+
| 2000-07-03      |
```

14.12.2.16. FROM_UNIXTIME

This function returns a Unix timestamp in the specified format.

FROM_UNIXTIME(unix_timestamp[,format])

- Description: The format parameter conforms to the format in the DATE_FORMAT function.
- Parameter types:

```
from_unixtime(varchar, varchar)
from_unixtime(varchar)
from_unixtime(double, varchar)
from_unixtime(double)
```

- Return value type: DATETIME.
- Example:

```
select from_unixtime('1447430881','%Y %M %h:%i:%s %x');
+-----+
| from_unixtime('1447430881','%Y %M %h:%i:%s %x') |
+-----+
| 2015 November 12:08:01 2015      |
```

14.12.2.17. HOUR

This function returns the hour part of a specified time.

HOUR(time)

- Parameter types:

```
hour(timestamp)
hour(datetime)
hour(date)
hour(time)
hour(varchar)
```

- Return value type: BIGINT.
- Example:

```
select hour(timestamp '2019-12-07 10:05:03');
+-----+
| hour(TIMESTAMP '2019-12-07 10:05:03') |
+-----+
|                10 |
```

14.12.2.18. LAST_DAY

This function returns the last day of the month for a date or datetime.

```
LAST_DAY(date)
```

- Parameter types:

```
last_day(varchar)
last_day(timestamp)
last_day(datetime)
last_day(date)
```

- Return value type: DATE.
- Example:

```
select last_day('2003-02-05');
+-----+
| last_day('2003-02-05') |
+-----+
| 2003-02-28           |
```

14.12.2.19. LOCALTIME/LOCALTIMESTAMP/NOW

This function returns the current timestamp.

```
localtime
localtime()
localtimestamp
localtimestamp()
now()
```

- Return value type: DATETIME.
- Example:

```
select now();
+-----+
| now() |
+-----+
| 2019-05-25 00:28:37 |
```

14.12.2.20. MAKEDATE

This function returns a date based on the year and dayofyear parameters.

```
MAKEDATE(year,dayofyear)
```

- Parameter types:

```
makedate(bigint, bigint)
makedate(varchar, varchar)
```

- Return value type: DATE.
- Example:

```
select makedate(2011,31), makedate(2011,32);
+-----+-----+
| makedate(2011, 31) | makedate(2011, 32) |
+-----+-----+
| 2011-01-31 | 2011-02-01 |
```

14.12.2.21. MAKETIME

This function returns a time based on the hour, minute, and second parameters.

```
MAKETIME(hour,minute,second)
```

- Parameter types:

```
maketime(bigint, bigint, bigint)
maketime(varchar, varchar, varchar)
```

- Return value type: TIME.
- Example:

```
select maketime(12,15,30);
+-----+
| maketime(12, 15, 30) |
+-----+
| 12:15:30 |
```

14.12.2.22. MINUTE

This function returns the minute part of a specified time.

```
MINUTE(time)
```

- Parameter types:

```
minute(timestamp)
minute(datetime)
minute(date)
minute(time)
minute(varchar)
```

- Return value type: BIGINT.
- Example:

```
select minute(timestamp '2008-02-03 10:05:03');
+-----+
| minute(TIMESTAMP '2008-02-03 10:05:03') |
+-----+
|                5 |
```

14.12.2.23. MONTH

This function returns the month for a date.

```
MONTH(date)
```

- Parameter types:

```
month(timestamp)
month(datetime)
month(date)
month(time)
month(varchar)
```

- Return value type: BIGINT.
- Example:

```
select month(timestamp '2008-02-03 00:00:00');
+-----+
| month(TIMESTAMP '2008-02-03 00:00:00') |
+-----+
|                2 |
```

14.12.2.24. MONTHNAME

This function returns the full name of the month for a date.

```
MONTHNAME(date)
```

- Parameter types:

```
monthname(timestamp)
monthname(datetime)
monthname(date)
monthname(varchar)
```

- Return value type: VARCHAR.
- Example:

```
select monthname(datetime '2008-02-03 00:00:00');
+-----+
| monthname(DATETIME '2008-02-03 00:00:00') |
+-----+
| February          |
```

14.12.2.25. PERIOD_ADD

This function adds N months to period P.

```
PERIOD_ADD(P,N)
```

- Parameter types:

```
period_add(bigint, bigint)
period_add(varchar, varchar)
period_add(varchar, bigint)
```

- Return value type: BIGINT.
- Example:

```
select period_add(200801,2);
+-----+
| period_add(200801, 2) |
+-----+
|          200803 |
```

14.12.2.26. PERIOD_DIFF

This function returns the number of months between periods P1 and P2.

```
PERIOD_DIFF(P1,P2)
```

- Parameter types:

```
period_diff(bigint, bigint)
period_diff(varchar, varchar)
```

- Return value type: BIGINT.
- Example:

```

select period_diff(200802,200703);
+-----+
| period_diff(200802, 200703) |
+-----+
|          11 |

```

14.12.2.27. QUARTER

This function returns the quarter of the year for a date. Valid values: [1,4] .

QUARTER(date)

- Parameter types:

```

quarter(datetime)
quarter(varchar)
quarter(timestamp)
quarter(date)

```

- Return value type: BIGINT.

- Example:

```

select quarter(datetime '2008-04-01 12:12:12');
+-----+
| quarter(DATETIME '2008-04-01 12:12:12') |
+-----+
|          2 |

```

14.12.2.28. SEC_TO_TIME

This function converts seconds to time.

SEC_TO_TIME(seconds)

- Parameter types:

```

sec_to_time(bigint)
sec_to_time(varchar)

```

- Return value type: TIME.

- Example:

```

select sec_to_time(2378);
+-----+
| sec_to_time(2378) |
+-----+
| 00:39:38 |

```

14.12.2.29. SECOND

This function returns the second part of a specified time. Valid values: [0,59] .

```
SECOND(time)
```

- Parameter types:

```
second(timestamp)
second(datetime)
second(date)
second(time)
second(varchar)
```

- Return value type: BIGINT.

- Example:

```
select second(timestamp '2019-03-12 12:13:14');
+-----+
| second(TIMESTAMP '2019-03-12 12:13:14') |
+-----+
|                14 |
```

14.12.2.30. STR_TO_DATE

This function converts a string to a date or datetime in the specified format.

```
STR_TO_DATE(str,format)
```

- Parameter types:

```
str_to_date(varchar, varchar)
```

- Return value type: DATETIME.

- Example:

```
select str_to_date('2017-01-06 10:20:30','%Y-%m-%d %H:%i:%s') as result;
+-----+
| result      |
+-----+
| 2017-01-06 10:20:30 |
```

14.12.2.31. SUBTIME

This function subtracts expr2 from expr1.

```
SUBTIME(expr1,expr2)
```

- Parameter types:

```
subtime(date, varchar)
subtime(datetime, varchar)
subtime(timestamp, varchar)
subtime(time, varchar)
subtime(varchar, varchar)
```

- Return value type: DATETIME.
- Example:

```
select subtime(date '2018-10-31','0:1:1');
+-----+
| subtime(DATE '2018-10-31', '0:1:1') |
+-----+
|          2018-10-30 23:58:59 |
```

14.12.2.32. SYSDATE

This function returns the system time.

```
SYSDATE()
```

- Return value type: DATETIME.
- Example:

```
select sysdate();
+-----+
| sysdate() |
+-----+
| 2019-05-26 00:47:21 |
```

14.12.2.33. TIME

This function returns the time in expr as a string.

```
TIME(expr)
```

- Parameter types:

```
time(varchar)
time(datetime)
time(timestamp)
```

- Return value type: VARCHAR.
- Example:

```
select time('2003-12-31 01:02:03');
+-----+
| time('2003-12-31 01:02:03') |
+-----+
| 01:02:03 |
```

14.12.2.34. TIME_FORMAT

This function returns time as a string in the specified format.

```
TIME_FORMAT(time,format)
```

- Description: The format parameter conforms to the format in the [DATE_FORMAT](#) function.
- Parameter types:

```
time_format(varchar, varchar)
time_format(timestamp, varchar)
time_format(datetime, varchar)
time_format(time, varchar)
time_format(date, varchar)
```

- Return value type: VARCHAR.
- Example:

```
select time_format('12:00:00', '%H %k %h %l %I');
+-----+
| time_format('12:00:00', '%H %k %h %l %I') |
+-----+
| 12 12 12 12 12 |
```

14.12.2.35. TIME_TO_SEC

This function converts time to seconds.

```
TIME_TO_SEC(time)
```

- Parameter types:

```
time_to_sec(varchar)
time_to_sec(datetime)
time_to_sec(timestamp)
time_to_sec(date)
time_to_sec(time)
```

- Return value type: BIGINT.
- Example:

```
select time_to_sec(datetime '2009-12-12 22:23:00');
+-----+
| time_to_sec(DATETIME '2009-12-12 22:23:00') |
+-----+
| 80580 |
```

14.12.2.36. TIMEDIFF

This function subtracts expr2 from expr1.

```
TIMEDIFF(expr1,expr2)
```

- Parameter types:

```
timediff(time, varchar)
timediff(time, time)
timediff(varchar, varchar)
```

- Return value type: DATETIME.
- Example:

```
select timediff(time '12:00:00','10:00:00');
+-----+
| timediff(TIME '12:00:00', '10:00:00') |
+-----+
| 02:00:00          |
```

14.12.2.37. TIMESTAMP

This function returns expr as a datetime value.

```
TIMESTAMP(expr)
```

- Parameter types:

```
timestamp(date)
timestamp(varchar)
```

- Return value type: DATETIME.
- Example:

```
select timestamp(date '2019-05-27');
+-----+
| timestamp(DATE '2019-05-27') |
+-----+
| 2019-05-27 00:00:00          |
```

14.12.2.38. TIMESTAMPADD

This function adds interval to datetime_expr.

```
TIMESTAMPADD(unit,interval,datetime_expr)
```

- Description: unit specifies the unit of interval. Valid values of unit: second, minute, hour, day, week, month, quarter, and year.
- Parameter types:

```
timestampadd(vchar, varchar, timestamp)
timestampadd(vchar, bigint, timestamp)
timestampadd(vchar, varchar, date)
timestampadd(vchar, bigint, date)
timestampadd(vchar, varchar, datetime)
timestampadd(vchar, bigint, datetime)
timestampadd(vchar, varchar, varchar)
timestampadd(vchar, bigint, varchar)
```

- Return value type: DATETIME.
- Example:

```
select timestampadd(second,'1',timestamp '2003-01-02 12:12:12')as result;
+-----+
| result |
+-----+
| 2003-01-02 12:12:13 |
```

14.12.2.39. TIMESTAMPDIFF

This function subtracts datetime_expr2 from datetime_expr1. unit specifies the unit of the result.

```
TIMESTAMPDIFF(unit,datetime_expr1,datetime_expr2)
```

- Description: Valid values of unit: second, minute, hour, day, week, month, quarter, and year. Use this function in the same way as the TIMESTAMPADD function.
- Parameter types:

```
timestampdiff(vchar, timestamp, timestamp)
timestampdiff(vchar, date, date)
timestampdiff(vchar, datetime, datetime)
timestampdiff(vchar, varchar, varchar)
```

- Return value type: BIGINT.
- Example:

```
select timestampdiff(second,datetime '2003-02-01 10:12:13',datetime '2003-05-01 10:12:13')as result;
+-----+
| result |
+-----+
| 7689600 |
```

14.12.2.40. TO_DAYS

This function returns the number of days since year 0 based on date.

```
TO_DAYS(date)
```

- Parameter types:

```
to_days(date)
to_days(time)
to_days(varchar)
to_days(timestamp)
to_days(datetime)
```

- Return value type: BIGINT.
- Example:

```
select to_days(date '2018-12-12');
+-----+
| to_days(DATE '2018-12-12') |
+-----+
|           737405 |
```

14.12.2.41. TO_SECONDS

This function returns the number of seconds since year 0 based on expr.

```
TO_SECONDS(expr)
```

- Parameter types:

```
to_seconds(date)
to_seconds(datetime)
to_seconds(timestamp)
to_seconds(varchar)
to_seconds(time)
```

- Return value type: BIGINT.
- Example:

```
select to_seconds(date '2019-09-08');
+-----+
| to_seconds(DATE '2019-09-08') |
+-----+
|           63735120000 |
```

14.12.2.42. UNIX_TIMESTAMP

UNIX_TIMESTAMP returns the Unix timestamp for the current time in seconds since '1970-01-01 00:00:00' UTC.

```
UNIX_TIMESTAMP([date])
```

- Description: UNIX_TIMESTAMP(date) returns the Unix timestamp for the date parameter in seconds since '1970-01-01 00:00:00' UTC.
- Parameter types:

```

unix_timestamp()
unix_timestamp(varchar)
unix_timestamp(timestamp)
unix_timestamp(date)
unix_timestamp(datetime)

```

- Return value type: BIGINT.
- Example:

```

select unix_timestamp();
+-----+
| unix_timestamp() |
+-----+
| 1558935850 |

```

14.12.2.43. UTC_DATE

This function returns the UTC date.

```
UTC_DATE()
```

- Return value type: VARCHAR.
- Example:

```

select utc_date();
+-----+
| utc_date() |
+-----+
| 2019-05-27 |

```

14.12.2.44. UTC_TIME

This function returns the UTC time.

```
UTC_TIME()
```

- Return value type: VARCHAR.
- Example:

```

select utc_time();
+-----+
| utc_time() |
+-----+
| 05:53:19 |

```

14.12.2.45. UTC_TIMESTAMP

This function returns the UTC timestamp.

```
utc_timestamp()
```

- Return value type: VARCHAR.
- Example:

```
select utc_timestamp();
+-----+
| utc_timestamp() |
+-----+
| 2019-05-27 05:55:15 |
```

14.12.2.46. WEEK

This function returns the week number for date, which is the week to which date belongs in the year.

```
WEEK(date[,mode])
```

- Description:
 - date specifies the date for which you want to obtain the week number.
 - mode is an optional parameter that specifies the logic for calculating the week number. It allows you to specify whether the week starts from Monday or Sunday. The returned value ranges from 0 to 52 or from 1 to 53. The following table describes the formats that mode supports.

Mode	First day of the week	Range
0	Sunday	0 to 53
1	Monday	0 to 53
2	Sunday	1 to 53
3	Monday	1 to 53
4	Sunday	0 to 53
5	Monday	0 to 53
6	Sunday	1 to 53
7	Monday	1 to 53

- Parameter types:

```
week(varchar)
week(varchar, bigint)
week(date)
week(date, bigint)
week(datetime)
week(datetime, bigint)
week(timestamp)
week(timestamp, bigint)
```

- Return value type: BIGINT.
- Example:

```
select week('2019-05-27');
+-----+
| week('2019-05-27') |
+-----+
|          21 |
```

14.12.2.47. WEEKDAY

This function returns the weekday for date. The result is an integer indicating the weekday. The mapping is as follows: 0 = Monday , 1 = Tuesday , and 6 = Sunday .

WEEKDAY(date)

- Parameter types:

```
weekday(timestamp)
weekday(datetime)
weekday(date)
weekday(varchar)
```

- Return value type: BIGINT.

- Example:

```
select weekday(timestamp '2019-05-27 00:09:00');
+-----+
| weekday(TIMESTAMP '2019-05-27 00:09:00') |
+-----+
|          0 |
```

14.12.2.48. WEEKOFYEAR

This function returns the calendar week for date. Valid values: [1, 53] .

WEEKOFYEAR(date)

- Parameter types:

```
weekofyear(timestamp)
weekofyear(datetime)
weekofyear(date)
weekofyear(varchar)
```

- Return value type: BIGINT.

- Example:

```
select weekofyear(timestamp '2019-05-27 09:00:00');
+-----+
| weekofyear(TIMESTAMP '2019-05-27 09:00:00') |
+-----+
|          22 |
```

14.12.2.49. YEAR

This function returns the year for date.

```
YEAR(date)
```

- Parameter types:

```
year(timestamp)
year(datetime)
year(date)
year(time)
year(varchar)
```

- Return value type: BIGINT.

- Example:

```
select year(timestamp '2019-05-27 00:00:00');
+-----+
| year(TIMESTAMP '2019-05-27 00:00:00') |
+-----+
|                2019 |
```

14.12.2.50. YEARWEEK

This function returns the year and week for a date.

```
YEARWEEK(date)
YEARWEEK(date,mode)
```

- Description: The year in the result may be different from the year in the date parameter for the first and the last week of the year.

mode works in the same way as mode in the [WEEK](#) function. For the single-parameter syntax, the value of mode is 0.

- Parameter types:

```
yearweek(timestamp)
yearweek(timestamp, bigint)
yearweek(datetime)
yearweek(datetime, bigint)
yearweek(date, bigint)
yearweek(date)
yearweek(varchar)
yearweek(varchar, bigint)
```

- Return value type: BIGINT.

- Example:

```
select yearweek(timestamp '2019-05-27 00:00:00');
+-----+
| yearweek(TIMESTAMP '2019-05-27 00:00:00') |
+-----+
|                201921 |
```

14.12.3. String functions

14.12.3.1. ASCII

This function returns the decimal ASCII code of the str character or the leftmost character in the string str.

```
ASCII(varchar str)
```

- Return value type: BIGINT.
- Example:

```
select ascii('2');
+-----+
| ascii('2') |
+-----+
|      50 |
```

14.12.3.2. BIN

This function returns the binary string of the integer N.

```
BIN(bigint N)
```

- Description: If N is null, null is returned.
- Return value type: VARCHAR.
- Example:

```
select bin(12);
+-----+
| bin(12) |
+-----+
| 1100 |
```

14.12.3.3. BIT_LENGTH

This function returns the length of the string str, measured in bits.

```
BIT_LENGTH(varchar str)
```

- Return value type: BIGINT.
- Example:

```
select bit_length('text');
+-----+
| bit_length('text') |
+-----+
|          32 |
```

14.12.3.4. CHAR

This function returns a string of decimal ASCII codes of the integers N1, N2, and so on.

```
CHAR(bigint N1, bigint N2...)
```

- Return value type: VARBINARY.
- Example:

```
select char(97,110,97,108,121,116,105,99,100,98);
+-----+
| char(97, 110, 97, 108, 121, 116, 105, 99, 100, 98) |
+-----+
| analyticdb |
```

14.12.3.5. CHAR_LENGTH/CHARACTER_LENGTH

This function returns the length of the string str, measured in characters.

```
CHAR_LENGTH(varchar str)
```

- Return value type: BIGINT.
- Example:

```
select char_length('China');
+-----+
| char_length('China') |
+-----+
|          2 |
```

14.12.3.6. CONCAT

This function concatenates strings. If any of the parameters is null, null is returned.

```
concat(varchar str1, ..., varchar strn)
```

- Return value type: VARCHAR.
- Example:

```
select concat('aliyun', ',', 'analyticdb');
+-----+
| concat('aliyun', ',', 'analyticdb') |
+-----+
| aliyun, analyticdb          |
```

14.12.3.7. CONCAT_WS

This function concatenates strings and separates them with delimiters. The separator parameter specifies the delimiter. Parameters whose value is null are not included in the final string.

```
concat_ws(vchar separator, varchar str1, ..., varchar strn)
```

- Return value type: VARCHAR.
- Example:

```
select concat_ws(',', 'First name', 'Second name', 'Last Name') as result;
+-----+
| result          |
+-----+
| First name,Second name,Last Name |
```

14.12.3.8. ELT

This function returns the string specified by the integer N.

```
ELT(bigint N, varchar str1, varchar str2, varchar str3,...)
```

- If N is less than 1 or greater than the number of the specified strings, null is returned.
- Return value type: VARCHAR.
- Example:

```
select elt(4, 'Aa', 'Bb', 'Cc', 'Dd');
+-----+
| elt(4, 'Aa', 'Bb', 'Cc', 'Dd') |
+-----+
| Dd          |
```

14.12.3.9. EXPORT_SET

This function returns a string in which the strings on and off are placed based on the bits 0 or 1 from right to left (from low-order to high-order) in the binary value of the bits integer.

```
EXPORT_SET(bigint bits, varchar on, varchar off[, varchar separator[, bigint number_of_bits]])EXPORT_SET(bigint bits, v
archar on, varchar off[, varchar separator[, bigint number_of_bits]])
```

- Description: The string on is placed for bit 1, and the string off is placed for bit 0. These strings are separated by delimiters. The default delimiter is a comma (.). The number_of_bits parameter specifies the number of bits that are checked. Default value: 64.

If the value of the number_of_bits parameter is greater than 64, the parameter is silently clipped to 64.

The same value is returned when the `number_of_bits` parameter is set to -1 or 64.

- Return value type: VARCHAR.
- Example:

```
select export_set(5,'1','0','','2);
+-----+
| export_set(5, '1', '0', '', 2) |
+-----+
| 1,0          |
```

14.12.3.10. FIELD

This function returns the position of the string `str` in the strings such as `str1`, `str2`, and `str3`. If the string `str` is not found, 0 is returned.

```
field(varchar str, varchar str1, varchar str2, varchar str3,...)
```

- Return value type: BIGINT.
- Example:

```
select field('Bb', 'Aa', 'Bb', 'Cc', 'Dd', 'Ff');
+-----+
| field('Bb', 'Aa', 'Bb', 'Cc', 'Dd', 'Ff') |
+-----+
|                2 |
```

14.12.3.11. FIND_IN_SET

This function returns the position of the string `str` in the string list `strlist`.

```
FIND_IN_SET(varchar str, varchar strlist)
```

- Description: If `str` is not found in `strlist` or `strlist` is empty, 0 is returned.
If either `str` or `strlist` is null, null is returned.
- Return value type: BIGINT.
- Example:

```
select find_in_set('b','a,b,c,d');
+-----+
| find_in_set('b', 'a,b,c,d') |
+-----+
|                2 |
```

14.12.3.12. FORMAT

This function formats the integer `X` to the `###,###.##` format rounded to `D` decimal places, and returns the result as a string.

```
format(double X, bigint D)
```

- Description: If D is 0, the result has no decimal point or fractional part.
- Return value type: BIGINT.
- Example:

```
select format(12332.123456, 4)as result1, format(12332.1,4)as result2, format(12332.2,0)as result3;
+-----+-----+-----+
| result1 | result2 | result3 |
+-----+-----+-----+
| 12,332.1235 | 12,332.1000 | 12,332 |
```

14.12.3.13. HEX

This function returns the hexadecimal string of the integer N, or returns a string consisting of the hexadecimal values of all characters in the string str.

```
HEX(bigint N)
HEX(varchar str)
```

- Return value type: VARCHAR.
- Example:

```
select hex(16);
+-----+
| hex(16) |
+-----+
| 10      |
```

14.12.3.14. INSTR

This function returns the position of the first occurrence of the substring substr in the string str.

```
INSTR(varchar str, varchar substr)
```

- Return value type: BIGINT.
- Example:

```
select instr('foobarbar', 'bar');
+-----+
| instr('foobarbar', 'bar') |
+-----+
|                4 |
```

14.12.3.15. LEFT

This function returns the leftmost len characters of the string str.

```
LEFT(varchar str, bigint len)
```

- Description: If str or len is null, null is returned.
- Return value type: VARCHAR.

- Example:

```
select left('foobarbar', 5);
+-----+
| left('foobarbar', 5) |
+-----+
| fooba                |
```

14.12.3.16. LENGTH/OCTET_LENGTH

This function returns the length of the string str.

```
length(varchar str)
```

- Return value type: BIGINT.
- Example:

```
select length('aliyun');
+-----+
| length('aliyun') |
+-----+
|          6      |
```

14.12.3.17. LIKE

The LIKE operator is used to match the string expression with the string pattern. If the two strings match, 1 is returned. Otherwise, 0 is returned.

```
expression [ NOT ] LIKE pattern [ESCAPE 'escape_char']
```

- Description: The string pattern can contain the following wildcard characters:
 - %: matches strings in any length.
 - _: matches a single character.

escape_char: escapes the percent signs (%) and underscores (_) in the string pattern. The percent signs (%) and underscores (_) following the escape character are not used as wildcard characters.
- Return value type: BIGINT.
- Example:

```
select 'David!' like 'David_' as result1, 'David!' not like 'David_' as result2, 'David!' like '%D%v%' as result3;
+-----+-----+-----+
| result1 | result2 | result3 |
+-----+-----+-----+
| 1      | 0      | 1      |
```

14.12.3.18. LOCATE

This function returns the position of the first occurrence of substring substr in string str, or returns the position of the first occurrence of the substring substr in the string str, starting at position pos.

```
LOCATE(varchar substr, varchar str)
LOCATE(varchar substr, varchar str, bigint pos)
```

- **Description:** If substr is not found in str, 0 is returned.
If substr or str is null, null is returned.
- **Return value type:** BIGINT.
- **Example:**

```
select locate('bar', 'foobarbar');
+-----+
| locate('bar', 'foobarbar') |
+-----+
|                4 |
```

14.12.3.19. LOWER/LCASE

This function converts the string str to lowercase.

```
lower(varchar str)
```

- **Return value type:** VARCHAR.
- **Example:**

```
select lower('Aliyun');
+-----+
| lower('Aliyun') |
+-----+
| aliyun      |
```

14.12.3.20. LPAD

This function returns the string str, left-padded with the string padstr to a length of len characters.

```
lpad(varchar str, bigint len, varchar padstr)
```

- **Description:** If str is longer than len, the return value is shortened to len characters.
- **Return value type:** VARCHAR.
- **Example:**

```
select lpad('Aliyun',9,'#');
+-----+
| lpad('Aliyun', 9, '#') |
+-----+
| ###Aliyun      |
```

14.12.3.21. LTRIM

This function removes the leading space characters of the string str.

```
LTRIM(varchar str)
```

- Return value type: VARCHAR.
- Example:

```
select ltrim(' abc');
+-----+
| ltrim(' abc') |
+-----+
| abc      |
```

14.12.3.22. MAKE_SET

This function returns a set value (a string containing substrings separated by delimiters) consisting of the string that have the corresponding bit in the bits set.

```
MAKE_SET(bits, str1, str2,...)
```

- Description: The string str1 corresponds to bit 0 and the string str2 corresponds to bit 1. The same rule applies to the rest numbers. The strings whose value is null are not included in the final string.
- Return value type: VARCHAR.
- Example:

```
select make_set(5,'hello','nice','world');
+-----+
| make_set(5, 'hello', 'nice', 'world') |
+-----+
| hello,world      |
```

14.12.3.23. MID

This function returns a substring that contains len characters in length from the string str, starting from the pos position.

```
MID(varchar str, bigint pos, bigint len)
```

- Return value type: VARCHAR.
- Example:

```
select mid('Quadratically',5,6);
+-----+
| mid('Quadratically', 5, 6) |
+-----+
| ratica      |
```

14.12.3.24. OCT

This function returns the octal string of the integer N.

```
OCT(bigint N)
```

- Description: If N is null, null is returned.
- Return value type: VARCHAR.
- Example:

```
select oct(12);
+-----+
| oct(12) |
+-----+
| 14    |
```

14.12.3.25. POSITION

This function returns the position of the first occurrence of the substring `substr` in the string `str`, starting from position 1. If `substr` is not found in `str`, 0 is returned.

```
POSITION(varchar substr IN varchar str)
```

- Return value type: BIGINT.
- Example:

```
select position('bar' in 'foobarbar');
+-----+
| locate('bar', 'foobarbar') |
+-----+
|                4 |
```

14.12.3.26. REPEAT

This function repeats the string `str` for `count` times.

```
REPEAT(varchar str, bigint count)
```

- If `count < 1`, an empty string is returned.
If `str` or `count` is null, null is returned.
- Return value type: VARCHAR.
- Example:

```
select repeat('a', 3);
+-----+
| repeat('a', 3) |
+-----+
| aaa          |
```

14.12.3.27. REPLACE

This function replaces all occurrences of the string `from_str` in the string `str` with the string `to_str`.

```
REPLACE(varchar str, varchar from_str, varchar to_str)
```

- Return value type: VARCHAR.

- Example:

```
select replace('WWW.aliyun.com', 'W', 'w');
+-----+
| replace('WWW.aliyun.com', 'W', 'w') |
+-----+
| www.aliyun.com           |
```

14.12.3.28. REVERSE

This function returns the string str with the order of the characters reversed.

```
REVERSE(varchar str)
```

- Return value type: VARCHAR.
- Example:

```
select reverse('123456');
+-----+
| reverse('123456') |
+-----+
| 654321           |
```

14.12.3.29. RIGHT

This function returns the rightmost len characters from the string str.

```
RIGHT(varchar str, bigint len)
```

- Description: If str or len is null, null is returned.
- Return value type: VARCHAR.
- Example:

```
select right('abc',3);
+-----+
| presto_right('abc', 3) |
+-----+
| abc                   |
```

14.12.3.30. RLIKE/REGEXP

This function matches the expression string with the pattern string, which is a regular expression. If the two strings match, 1 is returned. Otherwise, 0 is returned.

```
expression RLIKE pattern
expression REGEXP pattern
```

- If expression or pattern is null, null is returned.
- Return value type: BIGINT.
- Example:

```

select 'Michael!' regexp '.*';
+-----+
| regexp_like('Michael!', '.*') |
+-----+
|                1 |

```

14.12.3.31. RPAD

This function returns the string `str`, right-padded with the string `padstr` to a length of `len` characters.

```
rpad(vvarchar str, bigint len, varchar padstr)
```

- If `str` is longer than `len`, the return value is shortened to `len` characters.
- Return value type: VARCHAR.
- Example:

```

select rpad('Aliyun',9,'#');
+-----+
| rpad('Aliyun', 9, '#') |
+-----+
| Aliyun###           |

```

14.12.3.32. RTRIM

This function removes the trailing space characters of the string `str`.

```
RTRIM(vvarchar str)
```

- Return value type: VARCHAR.
- Example:

```

select rtrim('barbar ');
+-----+
| rtrim('barbar ') |
+-----+
| barbar          |

```

14.12.3.33. SPACE

This function returns a string consisting of a specified number of space characters.

```
SPACE(bigint N)
```

- Return value type: VARCHAR.
- Example:

```
select concat("#", space(6), "#");
+-----+
| concat('#', space(6), '#') |
+-----+
| # #           |
```

14.12.3.34. STRCMP

If the string str1 is the same as the string str2, 0 is returned. If the string str1 is smaller than the string str2 based on the current sort order, -1 is returned. Otherwise, 1 is returned.

```
STRCMP(varchar str1, varchar str2)
```

- Return value type: BIGINT.
- Example:

```
select strcmp('text', 'text2');
+-----+
| strcmp('text', 'text2') |
+-----+
|           -1 |
```

14.12.3.35. SUBSTR/SUBSTRING

```
SUBSTRING(varchar str, bigint pos)
SUBSTRING(varchar str FROM pos)
SUBSTRING(varchar str, bigint pos, bigint len)
SUBSTRING(varchar str FROM pos FOR len)
```

- Description:
 - SUBSTRING(varchar str, bigint pos) or SUBSTRING(varchar str FROM pos): returns the substring starting from the pos position to the end of the string. If pos < 0, the beginning of the substring is pos characters from the end of the string.
 - SUBSTRING(varchar str, bigint pos, bigint len) or SUBSTRING(varchar str FROM pos FOR len): returns a substring that contains len characters in length from the string, starting from the pos position. If pos < 0, the beginning of the substring is pos characters from the end of the string.
- Return value type: VARCHAR.
- Example:

```
select substr('helloworld', 6);
+-----+
| substr('helloworld', 6) |
+-----+
| world           |
```

14.12.3.36. SUBSTRING_INDEX

This function returns the substring before the last occurrence of the delim delimiter in the str string.

```
SUBSTRING_INDEX(varchar str, varchar delim, bigint count)
```

- **Description:**
 - If count > 0, this function returns all the characters to the left of the last delim delimiter.
 - If count < 0, this function returns all the characters to the right of the last delim delimiter.
 - The SUBSTRING_INDEX function performs a case-sensitive match when searching for the delim delimiter.
- **Return value type:** VARCHAR.
- **Example:**

```
select substring_index('www.aliyun.com', '.', 2);
+-----+
| substring_index('www.aliyun.com', '.', 2) |
+-----+
| www.aliyun                               |
```

14.12.3.37. TRIM

This function removes the leading and trailing space characters or other specified characters from a string.

```
TRIM([remstr FROM] str)
TRIM([{BOTH | LEADING | TRAILING} [remstr] FROM] str)
```

- **Return value type:** VARCHAR.
- **Example:**

```
select trim(' bar ');
+-----+
| trim(' bar ') |
+-----+
| bar          |
```

14.12.3.38. UPPER/UCASE

This function converts the string str to uppercase.

```
upper(varchar str)
```

- **Return value type:** VARCHAR.
- **Example:**

```
select upper('Aliyun');
+-----+
| upper('Aliyun') |
+-----+
| ALIYUN          |
```

14.12.4. Numeric functions

14.12.4.1. ABS

This function returns the absolute value of *x*.

```
abs(tinyint x)
abs(smallint x)
abs(int x)
abs(bigint x)
abs(float x)
abs(double x)
abs(decimal x)
```

- Return value types: LONG, DECIMAL, and DOUBLE.
- Example:

```
select abs(4.5);
+-----+
| abs(4.5) |
+-----+
| 4.5 |
```

14.12.4.2. ROUND

This function rounds the parameter *x*. *d* specifies the number of decimal places. The default value of *d* is 0. The rounding algorithm changes based on the data type of *x*.

```
round(tinyint x)
round(smallint x)
round(int x)
round(bigint x)
round(float x)
round(double x)
round(x, d)
```

- Description:
 - If *x* is null, null is returned.
 - If *d* > 0, the parameter is rounded to the specified decimal place.
 - If *d* = 0, the parameter is rounded to the nearest integer.
 - If *d* < 0, digits to the left of the decimal point of the parameter are rounded.
- Return value types: LONG, DECIMAL, and DOUBLE.
- Example:

```
select round(4);
+-----+
| round(4) |
+-----+
| 4 |
```

14.12.4.3. SQRT

This function returns the square root of x.

```
sqrt(double x)
```

- Return value type: DOUBLE.
- Example:

```
select sqrt(4.222);
+-----+
| sqrt(4.222) |
+-----+
| 2.054750593137766 |
```

14.12.4.4. LN

This function returns the natural logarithm of x.

```
ln(double x)
```

- Return value type: DOUBLE.
- Example:

```
select ln(2.718281828459045);
+-----+
| ln(2.718281828459045) |
+-----+
| 1.0 |
```

14.12.4.5. LOG

If called with one parameter, this function returns the natural logarithm of x. If called with two parameters, this function returns the logarithm of y to the base x.

- Return value type: DOUBLE.
- Example:

```
select log(16);
+-----+
| log(16) |
+-----+
| 2.772588722239781 |
```

14.12.4.6. LOG2

This function returns the base-2 logarithm of the parameter.

- Return value type: DOUBLE.
- Example:

```
select log2(8.7654);
+-----+
| log2(8.7654) |
+-----+
| 3.131819928389146 |
```

14.12.4.7. PI

This function returns the value of Pi.

```
pi()
```

- Return value type: DOUBLE.
- Example:

```
select pi();
+-----+
| pi() |
+-----+
| 3.141592653589793 |
```

14.12.4.8. LOG10

This function returns the base-10 logarithm of the parameter.

```
log10(double x)
```

- Return value type: DOUBLE.
- Example:

```
select log10(100.876);
+-----+
| log10(100.876) |
+-----+
| 2.0037878529824615 |
```

14.12.4.9. POWER/POW

This function returns the value of x raised to the power of y.

```
power(double x, double y)
pow(double x, double y)
```

- Return value type: DOUBLE.
- Example:

```
select power(1.2,3.4);
+-----+
| power(1.2, 3.4) |
+-----+
| 1.858729691979481 |
```

14.12.4.10. RADIANS

This function converts degrees to radians.

```
radians(double x)
```

- Return value type: DOUBLE.
- Example:

```
select radians(60.0);
+-----+
| radians(60.0) |
+-----+
| 1.0471975511965976 |
```

14.12.4.11. DEGREES

This function converts radians to degrees.

```
degrees(double x)
```

- Return value type: DOUBLE.
- Example:

```
select degrees(1.3);
+-----+
| degrees(1.3) |
+-----+
| 74.48451336700703 |
```

14.12.4.12. SIGN

This function returns the sign of x.

```
sign(smallint x)
sign(tinyint x)
sign(int x)
sign(bigint x)
sign(float x)
sign(double x)
sign(decimal x)
```

- Return value type: LONG.

- Example:

```
select sign(12);
+-----+
| sign(12) |
+-----+
|    1 |
```

14.12.4.13. CEILING/CEIL

This function returns the minimum integer value that is greater than the value of x.

```
ceiling(tinyint x)
ceiling(smallint x)
ceiling(int x)
ceiling(bigint x)
ceiling(float x)
ceiling(double x)
```

- Return value types: LONG, DECIMAL, and DOUBLE.
- Example:

```
select ceiling(4);
+-----+
| ceiling(4) |
+-----+
|    4 |
```

14.12.4.14. FLOOR

This function returns the maximum integer value that is less than the value of x.

```
floor(tinyint x)
floor(smallint x)
floor(int x)
floor(bigint x)
floor(float x)
floor(double x)
```

- Return value types: LONG, DECIMAL, and DOUBLE.
- Example:

```
select floor(4.5);
+-----+
| floor(4.5) |
+-----+
|    4.0 |
```

14.12.4.15. EXP

This function returns the value of e (the base of natural logarithms) raised to the power of x.

```
exp(double x)
```

- Return value type: DOUBLE.
- Example:

```
select exp(4.5);
+-----+
| exp(4.5) |
+-----+
| 90.01713130052181 |
```

14.12.4.16. COS

This function returns the cosine of x.

```
cos(double x)
```

- Return value type: DOUBLE.
- Example:

```
select cos(1.3);
+-----+
| cos(1.3) |
+-----+
| 0.26749882862458735 |
```

14.12.4.17. ACOS

This function returns the arc cosine of x.

```
acos(double x)
```

- Description: If $x > 1$ or $x < -1$, null is returned.
- Return value type: DOUBLE.
- Example:

```
select acos(0.5);
+-----+
| acos(0.5) |
+-----+
| 1.0471975511965979 |
```

14.12.4.18. TAN

This function returns the tangent of x.

```
tan(double x)
```

- Return value type: DOUBLE.

- Example:

```
select tan(8);
+-----+
| tan(8) |
+-----+
| -6.799711455220379 |
```

14.12.4.19. ATAN

This function returns the arc tangent of x.

```
atan(double x)
```

- Return value type: DOUBLE.
- Example:

```
select atan(0.5);
+-----+
| atan(0.5) |
+-----+
| 0.4636476090008061 |
```

14.12.4.20. ATAN2

This function returns the arc tangent of the result of x divided by y.

```
atan2(double x, double y)
atan(double x, double y)
```

- Return value type: DOUBLE.
- Example:

```
select atan2(0.5,0.3);
+-----+
| atan2(0.5, 0.3) |
+-----+
| 1.0303768265243125 |
```

14.12.4.21. COT

This function returns the cotangent of x.

```
cot(double x)
```

- Return value type: DOUBLE.
- Example:

```

select cot(1.234);
+-----+
| cot(1.234) |
+-----+
| 0.35013639786701445 |

```

14.12.4.22. ASIN

This function returns the arc sine of x.

```
asin(double x)
```

- Return value type: DOUBLE.
- Example:

```

select asin(0.5);
+-----+
| asin(0.5) |
+-----+
| 0.5235987755982989 |

```

14.12.4.23. SIN

This function returns the sine of x.

```
sin(double x)
```

- Return value type: DOUBLE.
- Example:

```

select sin(1.234);
+-----+
| sin(1.234) |
+-----+
| 0.9438182093746337 |

```

14.12.5. Arithmetic operators

+

- Description: This operator is used for addition.
- Example:

```

select 3+5;
+-----+
| _col0 |
+-----+
| 8 |

```

-

- **Description:** This operator is used for subtraction.
- **Example:**

```
select 3-5;
+-----+
|_col0|
+-----+
|  -2|
```

*

- **Description:** This operator is used for multiplication.
- **Example:**

```
select 3*pi();
+-----+
|_col0|
+-----+
| 9.42477796076938|
```

/

- **Description:** This operator is used for division.
- **Example:**

```
select 3/pi();
+-----+
|_col0|
+-----+
| 0.954929658551372|
```

DIV

- **Description:** This operator is used for division. The decimal part of the result is discarded.
- **Example:**

```
select 3 div pi();
+-----+
|_col0|
+-----+
|  0|
```

% or MOD

- **Description:** This operator returns the remainder of one argument divided by the other argument.
- **Example:**

```
select 3 mod pi();
+-----+
|_col0|
+-----+
|  3.0|
```

-

- **Description:** This operator converts a positive number to a negative number or a negative number to a positive number.
- **Example:**

```
select - 2;
+-----+
| _col0 |
+-----+
|  -2 |
```

14.12.6. Bit functions and operators

BIT_COUNT

- **Description:** This function converts the parameter to a binary value, and then returns the number of bits that are set to 1 in the value.
- **Return value type:** BIGINT.
- **Example:**

```
select bit_count(2);
+-----+
| bit_count(2) |
+-----+
|      1 |
```

&

- **Description:** This function is used for bitwise AND.
- **Return value type:** BIGINT.
- **Example:**

```
select 12 & 15;
+-----+
| bitwise_and(12, 15) |
+-----+
|      12 |
```

~

- **Description:** This function inverts all bits.
- **Return value type:** BIGINT.
- **Example:**

```
select 2 & ~1;
+-----+
| bitwise_and(2, bitwise_not(1)) |
+-----+
|      2 |
```

|

- Description: This function is used for bitwise OR.
- Return value type: BIGINT.
- Example:

```
select 29 | 15;
+-----+
| bitwise_or(29, 15) |
+-----+
|          31 |
```

^

- Description: This function is used for bitwise XOR.
- Return value type: BIGINT.
- Example:

```
select 1 ^ 10;
+-----+
| bitwise_xor(1, 10) |
+-----+
|          11 |
```

>>(BITWISE_RIGHT_SHIFT)

```
bitwise_right_shift(double x, double y)
bitwise_right_shift(varchar x, varchar y)
bitwise_right_shift(bigint x, bigint y)
```

- Description: This function shifts a value to the right.
- Return value type: BIGINT.
- Example:

```
select 3 >> 2;
+-----+
| bitwise_right_shift(3, 2) |
+-----+
|          0 |
```

<<(BITWISE_LEFT_SHIFT)

```
bitwise_left_shift(double x, double y)
bitwise_left_shift(varchar x, varchar y)
bitwise_left_shift(bigint x, bigint y)
```

- Description: This function shifts a value to the left.
- Return value type: BIGINT.
- Example:

```

SELECT 3 << 2;
+-----+
| bitwise_left_shift(3, 2) |
+-----+
|          12 |

```

14.12.7. Control flow functions

The control flow functions described in this topic use the conditiontest table as test data.

```
create table conditiontest(a int) distributed by hash(a);
```

```
insert into conditiontest values (1),(2),(3);
```

```

SELECT * FROM conditiontest;
+----+
| a |
+----+
| 2 |
| 1 |
| 3 |

```

CASE

```

CASE expression
  WHEN value THEN result
  [ WHEN ... ]
  [ ELSE result ]
END

```

- **Description:** A simple CASE expression checks each value from left to right, and returns the result for the first value that is equal to the expression. If no match is found, result is returned.
- **Example:**

```

SELECT a,
  CASE a
    WHEN 1 THEN 'one'
    WHEN 2 THEN 'two'
    ELSE 'three'
  END as caseresult
FROM conditiontest;
+----+-----+
| a | caseresult |
+----+-----+
| 2 | two      |
| 1 | one      |
| 3 | three    |

```

```

CASE
  WHEN condition THEN result
  [ WHEN ... ]
  [ ELSE result ]
END

```

- **Description:** An advanced CASE expression calculates conditions from left to right, and returns the result for the first condition that is TRUE. If no condition is TRUE, result is returned.
- **Example:**

```

SELECT a,
       CASE a
         WHEN a=1 THEN 'one1'
         WHEN a=2 THEN 'two2'
         ELSE 'three3'
       END as caseresult
FROM conditiontest;
+---+-----+
| a | caseresult |
+---+-----+
| 1 | one1      |
| 3 | three3    |
| 2 | three3    |

```

IF

```
if(condition, true_value)
```

- **Description:** If the condition is true, true_value is returned. Otherwise, null is returned.
- **Example:**

```

SELECT IF((2+3)>4,5);
+-----+
| _col0 |
+-----+
| 5 |

```

```
if(condition, true_value, false_value)
```

- **Description:** If the condition is true, true_value is returned. Otherwise, false_value is returned.
- **Example:**

```

SELECT IF((2+3)<5,5,6);
+-----+
| _col0 |
+-----+
| 6 |

```

IFNULL

```
IFNULL(expr1,expr2)
```

- **Description:** If expr1 is not null, expr1 is returned. Otherwise, expr2 is returned.
- **Example:**

```
SELECT IFNULL(NULL,2);
+-----+
|_col0|
+-----+
|  2 |
SELECT IFNULL(1,0);
+-----+
|_col0|
+-----+
|  1 |
```

NULLIF

```
NULLIF(expr1,expr2)
```

- **Description:** If expr1 is equal to expr2, null is returned. Otherwise, expr1 is returned.
- **Example:**

```
SELECT NULLIF (2,1);
+-----+
|_col0|
+-----+
|  2 |
SELECT NULLIF (2,2);
+-----+
|_col0|
+-----+
| NULL |
```

15. AnalyticDB for PostgreSQL

15.1. What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) is a distributed cloud database service that uses multiple compute nodes to provide massively parallel processing (MPP) data warehousing services.

AnalyticDB for PostgreSQL is developed based on the open source Greenplum database project and enhanced by Alibaba Cloud. This service has the following features:

- Compatible with Greenplum and all tools that support it.
- Supports OSS, JSON, and HyperLogLog, a probability cardinality estimation algorithm.
- Supports SQL:2003-compliant syntax and OLAP aggregate functions to provide flexible hybrid analysis.
- Supports both row store and column store to enhance analytics performance.
- Leverages data compression technologies to reduce storage costs.
- Provides online expansion and performance monitoring services to enable DBAs, developers, and data analysts to focus on improving enterprise productivity and creating core business value instead of managing and maintaining large numbers of MPP clusters.

15.2. Quick start

15.2.1. Overview

This topic provides a quick start guide about how to perform management tasks for AnalyticDB for PostgreSQL instances such as creating an instance and logging on to a database.

- [Log on to the AnalyticDB for PostgreSQL console](#)

This topic describes how to log on to the AnalyticDB for PostgreSQL console.

- [Create an instance](#)

You can create an instance in the console and then manage the instance.

- [Configure a whitelist](#)

To ensure a secure and stable database, you must add IP addresses or CIDR blocks that are allowed to access the database to a whitelist of the instance before you use the AnalyticDB for PostgreSQL instance.

- [Create an initial account](#)

After you create an instance, you must create an initial account to log on to the database.

- [Connect to a database](#)

You can use a client that supports PostgreSQL or Greenplum to connect to the database.

15.2.2. Log on to the AnalyticDB for PostgreSQL console

This topic describes how to log on to the AnalyticDB for PostgreSQL console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > AnalyticDB for PostgreSQL**.

15.2.3. Create an instance

You can create an instance in the console and then manage the instance.

1. **Log on to the AnalyticDB for PostgreSQL console.**
2. In the upper-right corner of the page, click **Create Instance**.
3. On the **AnalyticDB for PostgreSQL buy page**, configure the following parameters.

Section	Parameter	Description
Region	Organization	The organization to which the instance belongs.
	Resource Set	The resource set to which the instance belongs.
	Region	The region of the instance. Note If you need to access the AnalyticDB for PostgreSQL instance from an ECS instance over VPC, you must deploy the instance in the same region and zone as those of the ECS instance.
	Zone	The zone of the instance.
Basic Settings	Engine	Currently, only the integrated computing and storage version is supported.
	Engine Version	The engine version of the instance.
	Node Type	The unit of computing resources. Different group types have different storage capacities and computing capabilities.
	Nodes	The number of compute nodes. An instance must contain at least two compute nodes. The performance of an instance scales linearly with the number of compute nodes.
	Network Type	Valid values: <ul style="list-style-type: none"> ◦ <i>Classic Network</i>: Cloud services on a classic network are not isolated from each other. Unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. ◦ <i>VPC</i>: A VPC helps you to build an isolated network environment in Alibaba Cloud. You can customize the route table, IP address range, and gateway in a VPC. We recommend that you select VPC for enhanced security. You can create a VPC in advance, or change the network type to VPC after instance creation.

Network Section	Parameter	Description
	VPC	The VPC where the AnalyticDB for PostgreSQL instance is located.  Note Virtual Private Cloud (VPC): You can use a VPC to build an isolated network environment in Alibaba Cloud. You can customize the route table, IP address range, and gateway in a VPC.
	VSwitch	The VSwitch where the AnalyticDB for PostgreSQL instance is located.
	IP Whitelist	The IP addresses that are allowed to access the instance.

- After you have configured the preceding parameters, click **Submit**.

15.2.4. Configure a whitelist

To ensure a secure and stable database, you must add IP addresses or CIDR blocks that are allowed to access the database to a whitelist.

- Log on to the [AnalyticDB for PostgreSQL console](#).
- Find the target instance and click its ID. The **Basic Information** page appears.
- In the left-side navigation pane, click **Security Controls**. The **Security Controls** page appears.
- On the **Whitelist Settings** tab, click **Modify** corresponding to the *default* whitelist. The **Modify Group** page appears.

 **Note** You can also click **Clear** corresponding to the *default* whitelist to delete the IP addresses of the default whitelist, and then click **Add Group** to create a new whitelist.

- Delete 127.0.0.1 from the *default* whitelist and enter your IP addresses in the whitelist. The following table lists the parameters.

Parameter	Description
Whitelist Name	Specify the name of the whitelist. The whitelist name must be 2 to 32 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a letter or digit. The default whitelist cannot be modified or deleted.
IP Addresses	Enter the CIDR blocks or IP addresses that are allowed to access the database. Use commas (,) to separate multiple CIDR blocks or IP addresses. <ul style="list-style-type: none"> A whitelist can contain IP addresses such as 10.10.10.1 and CIDR blocks such as 10.10.10.0/24. This CIDR block indicates that any IP addresses in the 10.10.10.X format have access to the database. The percent sign (%) or 0.0.0.0/0 indicates that any IP addresses are allowed to access the database.  Notice This configuration is not recommended because it reduces the security of the database. <ul style="list-style-type: none"> Default whitelists of new instances contain the loopback address 127.0.0.1. This configuration allows no access from external IP addresses. You can add up to 999 IP addresses or CIDR blocks to a whitelist group.

- Click **OK** to create a whitelist.

What's next

- We recommend that you regularly maintain the whitelist to ensure secure access for AnalyticDB for PostgreSQL.
- You can click **Modify** or **Delete** to modify or delete custom whitelists.

15.2.5. Create an initial account

After you create an instance, you must create an initial account to log on to the database.

1. [Log on to the AnalyticDB for PostgreSQL console](#).
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Account Management**. The **Account Management** page appears.
4. In the upper-right corner of the page, click **Create Account**. The **Create Account** page appears.
5. Enter the database account and password, and click **OK**.

Parameter	Description
Account	The name of the account must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a letter and end with a letter or digit.
New Password	The password must be 8 to 32 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
Password	Enter the password again.

15.2.6. Obtain the client tool

The interface protocol of AnalyticDB for PostgreSQL is compatible with Greenplum Community Edition and PostgreSQL 8.2. Because of this, you can use the Greenplum or PostgreSQL client to connect to AnalyticDB for PostgreSQL.

Note

Apsara Stack is an isolated environment. You must deploy software installation packages to the internal environment.

Graphical client tools

AnalyticDB for PostgreSQL users can directly use client tools that support Greenplum, such as [SQL Workbench](#), [Navicat Premium](#), [Navicat for PostgreSQL](#), and [pgAdmin III \(1.6.3\)](#).

Command-line client psql (for RHEL 6, RHEL 7, CentOS 6, and CentOS 7)

For Red Hat Enterprise Linux (RHEL) and CentOS 6 or 7, you can download the tools from the following addresses and decompress the packages to use them:

- For RHEL 6 or CentOS 6, click [hybriddb_client_package_el6](#).
- For RHEL version 7 or CentOS version 7, click [hybriddb_client_package_el7](#).

Command-line client psql (for other Linux systems)

The compilation methods for client tools applicable to other Linux systems are as follows:

1. Obtain the source code by using one of the following methods:
 - Obtain the git directory. You must first install the git tool.

```
git clone https://github.com/greenplum-db/gpdb.git
cd gpdb
git checkout 5d870156
```

- Download the code.

```
wget https://github.com/greenplum-db/gpdb/archive/5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
unzip 5d87015609abd330c68a5402c1267fc86cbc9e1f.zip
cd gpdb-5d87015609abd330c68a5402c1267fc86cbc9e1f
```

2. Use gcc and other compilers.

```
./configure
make -j32
make install
```

3. Use psql and pg_dump. The paths of the two tools are as follows:

```
psql: /usr/local/pgsql/bin/psql
```

```
pg_dump: /usr/local/pgsql/bin/pg_dump
```

Command-line client psql (for Windows and other systems)

For client tools for Windows and other systems, go to the Pivotal website to download [HybridDB Client](#)

15.2.7. Connect to a database

The Greenplum Database and AnalyticDB for PostgreSQL are both developed based on PostgreSQL 8.2 and fully compatible with its messaging protocol. AnalyticDB for PostgreSQL users can use tools that support the PostgreSQL 8.2 message protocol, such as libpq, JDBC, ODBC, psycopg2, and pgAdmin III.

Context

AnalyticDB for PostgreSQL provides psql, a binary program of Red Hat. For more information about the download link, see [Obtain the client tool](#). The Greenplum official website provides an easy-to-install installation package that includes JDBC, ODBC, and libpq. For more information, see [Greenplum official documentation](#).

Note

- Apsara Stack is an isolated environment. To access Apsara Stack, you must prepare the necessary software installation packages in advance.
- AnalyticDB for PostgreSQL instances can only be accessed by clients deployed on ECS instances within the same region and zone.

psql

psql is a common tool used together with Greenplum, and provides a variety of command functions. Its binary files are located in the *bin* directory of Greenplum. The procedure is as follows:

1. Connect to AnalyticDB for PostgreSQL by using one of the following methods:

- Connection string

```
psql "host=yourgpdbaddress.gpdb.rds.aliyuncs.com port=3432 dbname=postgres user=gpdbaccount password=gpdbpassword"
```

- Specified parameters

```
psql -h yourgpdbaddress.gp.aliyun-inc.com -p 3432 -d postgres -U gpdbaccount
```

Parameters:

- -h: specifies the host address.
- -p: specifies the port number.
- -d: specifies the database. The default database is postgres.
- -U: specifies the user to connect to the database.

In psql, you can run the `psql --help` command to view more options. You can run the `\?` command to view the commands supported in psql.

2. Enter the password to go to the psql shell interface.

```
postgres=>
```

References

- For more information about the Greenplum psql usage, see [psql](#).
- AnalyticDB for PostgreSQL also supports psql statements of PostgreSQL. Pay attention to the differences between Greenplum psql and PostgreSQL psql. For more information, see [PostgreSQL 8.3.23 Documentation - psql](#).

pgAdmin III

pgAdmin III is a PostgreSQL graphical client and can be directly used to connect to AnalyticDB for PostgreSQL. For more information, click [here](#). For more information about other graphical clients, see [Obtain the client tool](#).

1. Download pgAdmin III 1.6.3 or earlier versions.

You can download pgAdmin III 1.6.3 from the [PostgreSQL website](#). PgAdmin III 1.6.3 supports various operating systems, such as Windows, macOS, and Linux.

 **Note** AnalyticDB for PostgreSQL is compatible with PostgreSQL 8.2. Therefore, you must use pgAdmin III 1.6.3 or earlier to connect to AnalyticDB for PostgreSQL. pgAdmin 4 and later versions are not supported.

2. Choose **File > Add Server**.
3. In the New Server Registration dialog box that appears, enter the configuration information.
4. Click **OK** to connect to AnalyticDB for PostgreSQL.

JDBC

JDBC uses the interface provided by PostgreSQL. The download methods are as follows:

- Click [PostgreSQL JDBC Driver](#) to download the official JDBC of PostgreSQL, and then add it to the environment variables.
- Use the tools provided by the Greenplum website. For more information, see [Greenplum Database 4.3 Connectivity Tools for UNIX](#).

The sample code is as follows:

```
import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement; public class gp_conn { public static void main(String[] args) { try { Class.forName("org.postgresql.Driver"); Connection db = DriverManager.getConnection("jdbc:postgresql://mygpdbpub.gpdb.rds.aliyuncs.com:3432/postgres","mygpdb","mygpdb"); Statement st = db.createStatement(); ResultSet rs = st.executeQuery("select * from gp_segment_configuration;"); while (rs.next()) { System.out.print(rs.getString(1)); System.out.print(" | "); System.out.print(rs.getString(2)); System.out.print(" | "); System.out.print(rs.getString(3)); System.out.print(" | "); System.out.print(rs.getString(4)); System.out.print(" | "); System.out.print(rs.getString(5)); System.out.print(" | "); System.out.print(rs.getString(6)); System.out.print(" | "); System.out.print(rs.getString(7)); System.out.print(" | "); System.out.print(rs.getString(8)); System.out.print(" | "); System.out.print(rs.getString(9)); System.out.print(" | "); System.out.print(rs.getString(10)); System.out.print(" | "); System.out.println(rs.getString(11)); } rs.close(); st.close(); } catch (ClassNotFoundException) { e.printStackTrace(); } catch (SQLException e) { e.printStackTrace(); }}
```

Python

Python uses `psycopg2` to connect to Greenplum and PostgreSQL. Procedure:

1. Install `psycopg2`. There are three installation methods in CentOS:

- Method 1: Run the `yum -y install python-psycopg2` command.
- Method 2: Run the `pip install psycopg2` command.
- Method 3: Run the source code:

```
yum install -y postgresql-devel*
wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.tar.gz
tar xf psycopg2-2.6.tar.gz
cd psycopg2-2.6
python setup.py build
sudo python setup.py install
```

2. Run the following commands to set `PYTHONPATH` and reference it:

```
import psycopg2
sql = 'select * from gp_segment_configuration;'
conn = psycopg2.connect(database='gpdb', user='mygpdb', password='mygpdb', host='mygpdbpub.gpdb.rds.aliyuncs.com', port=3432)
conn.autocommit = True
cursor = conn.cursor()
cursor.execute(sql)
rows = cursor.fetchall()
for row in rows:
    print row
conn.commit()
conn.close()
```

A similar output is displayed:

```
(1, -1, 'p', 'p', 's', 'u', 3022, '192.168.2.158', '192.168.2.158', None, None)(6, -1, 'm', 'm', 's', 'u', 3019, '192.168.2.47', '192.168.2.47', None, None)(2, 0, 'p', 'p', 's', 'u', 3025, '192.168.2.148', '192.168.2.148', 3525, None)(4, 0, 'm', 'm', 's', 'u', 3024, '192.168.2.158', '192.168.2.158', 3524, None)(3, 1, 'p', 'p', 's', 'u', 3023, '192.168.2.158', '192.168.2.158', 3523, None)(5, 1, 'm', 'm', 's', 'u', 3026, '192.168.2.148', '192.168.2.148', 3526, None)
```

libpq

libpq is the C language interface to AnalyticDB for PostgreSQL. You can use the libpq library to access and manage PostgreSQL databases in a C program. You can locate its static and dynamic libraries under the lib directory.

For the example programs, visit [Example Programs](#).

For more information about libpq, see [PostgreSQL 9.4.17 Documentation - Chapter 31. libpq - C Library](#).

ODBC

PostgreSQL ODBC is an open-source version based on the GNU Lesser General Public License (LGPL) protocol. You can download it from the [PostgreSQL website](#).

1. Install the driver.

```
yum install -y unixODBC.x86_64
yum install -y postgresql-odbc.x86_64
```

2. View the driver configuration.

```
cat /etc/odbcinst.ini
# Example driver definitions
# Driver from the postgresql-odbc package
# Setup from the unixODBC package
[PostgreSQL]
Description = ODBC for PostgreSQL
Driver = /usr/lib/psqlodbcw.so
Setup = /usr/lib/libodbcpsqlS.so
Driver64 = /usr/lib64/psqlodbcw.so
Setup64 = /usr/lib64/libodbcpsqlS.so
FileUsage = 1
# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
Description = ODBC for MySQL
Driver = /usr/lib/libmyodbc5.so
Setup = /usr/lib/libodbcmyS.so
Driver64 = /usr/lib64/libmyodbc5.so
Setup64 = /usr/lib64/libodbcmyS.so
FileUsage = 1
```

3. Configure the DSN. Replace the `****` in the following code with the corresponding connection information.

```
[mygpdb]
Description = Test to gp
Driver = PostgreSQL
Database = ****
Servername = ****.gpdb.rds.aliyuncs.com
UserName = ****
Password = ****
Port = ****
ReadOnly = 0
```

4. Test connectivity.

```

echo "select count(*) from pg_class" | isql mygpdb
+-----+
| Connected!          |
|                    |
| sql-statement      |
| help [tablename]   |
| quit               |
|                    |
+-----+
SQL> select count(*) from pg_class
+-----+
| count  |
+-----+
| 388    |
+-----+
SQLRowCount returns 1
1 rows fetched

```

5. After ODBC is connected to the instance, connect the application to ODBC. For more information, see [PostgreSQL ODBC Driver](#) and [psqlODBC HOWTO - C#](#).

References

- [Pivotal Greenplum documentation](#)
- [PostgreSQL psqlODBC](#)
- [Compiling psqlODBC on Unix](#)
- [Download ODBC connectors](#)
- [Download JDBC connectors](#)
- [The PostgreSQL JDBC Interface](#)

15.3. Instances

15.3.1. Reset the password

If you forget the password of your database account, you can reset the password in the AnalyticDB for PostgreSQL console.

 **Note** We recommend that you change your password periodically to ensure data security.

1. [Log on to the AnalyticDB for PostgreSQL console](#).
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Account Management**. The **Account Management** page appears.
4. Click **Reset Password** in the corresponding **Actions** column of the account. The **Reset Account Password** page appears.
5. After you enter and confirm the new password, click **OK**.

 **Note** The password must be 8 to 32 characters in length. It must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. We recommend that you do not use a previously used password.

15.3.2. View monitoring information

You can go to the monitoring information page in the console to view the operation status of an instance.

1. [Log on to the AnalyticDB for PostgreSQL console.](#)
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Monitoring and Alarms**. The **Monitoring and Alarms** page appears. Specify a duration of time n up to seven days in length to view the metrics for that last n period.

15.3.3. Switch the network type of an instance

The default network type of an instance is Virtual Private Cloud (VPC). After an instance is created, you can switch its network type between classic network and VPC as needed.

Context

AnalyticDB for PostgreSQL supports two network types: classic network and VPC. Both network types use BGP connections, and are independent of the public network of your service provider. These network types only differ in functions, and you can choose a network type based on your requirements. The two network types are applicable to different scenarios:

- **Classic network:** IP addresses are allocated by Alibaba Cloud. Classic networks are easy to configure and use. This network type is suitable for users who do not need to perform complex operations, or who only require short deployment cycles.
- **VPC:** a logically isolated private network. You can customize the network topology and IP addresses and connect through a leased line. This network type is suitable for advanced users.

 **Warning** Switching the network type will cause the database service to stop. Proceed with caution.

1. [Log on to the AnalyticDB for PostgreSQL console.](#)
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the left-side navigation pane, click **Database Connection**. The **Database Connection** page appears.
4. In the upper-right corner of the page, click **Switch to Classic Network** or **Switch to VPC**.
5. If you click **Switch to VPC**, you must select the destination VPC and VSwitch. Click **OK**.

 **Note** To switch the network type to VPC, a VPC and a VSwitch must exist or be created in the zone where the instance is located.

6. If you click **Switch to Classic Network**, click **OK** in the displayed message.

 **Note** After you switch the network type, it takes 3 to 30 minutes for the instance to enter the running state.

15.3.4. Restart an instance

To better meet your needs, AnalyticDB for PostgreSQL automatically updates the database kernel version. When you create an instance, the latest database kernel is used by default. After a new version is released, you can restart your instance to update the database kernel and use its extended features. This topic describes how to restart an instance.

 **Warning** Restarting an instance will cause the database service to stop. Proceed with caution.

1. [Log on to the AnalyticDB for PostgreSQL console](#).
2. Find the target instance and click its ID. The **Basic Information** page appears.
3. In the upper-right corner of the page, click **Restart Instance**.

 **Note** The restart process typically takes from 3 to 30 minutes. During the restart period, the instance cannot provide external services. We recommend that you take precautionary measures before restarting instances. After the instance has been restarted and enters the running state, you can access the database.

15.3.5. Import data

15.3.5.1. Import or export data from or to OSS in parallel

AnalyticDB for PostgreSQL can import or export data from or to OSS tables in parallel by using the OSS external table feature, `gpossex`. AnalyticDB for PostgreSQL also supports GZIP compression for OSS external tables to reduce file size and storage costs. `gpossex` can read from and write to TEXT and CSV files, even when they are compressed in GZIP packages.

- Create an OSS external table extension (`oss_ext`)

To use an OSS external table, you must first create an OSS external table extension in AnalyticDB for PostgreSQL. You must create an extension for each database that you need to access.

- Creation statement: `CREATE EXTENSION IF NOT EXISTS oss_ext;`
- Deletion statement: `DROP EXTENSION IF EXISTS oss_ext;`

- Import data in parallel

- i. Distribute data evenly among multiple OSS files for storage. We recommend that you set the number of OSS files to an integer that is the multiple of the number of compute nodes in AnalyticDB for PostgreSQL.
- ii. Create a `READABLE` external table in AnalyticDB for PostgreSQL.
- iii. Execute the following statement to import data in parallel:

```
INSERT INTO <destination table> SELECT * FROM <external table>
```

 **Note**

- The data import performance depends on the OSS performance and resources of the AnalyticDB for PostgreSQL instance, such as CPU, I/O, memory, and network resources. To ensure the best import performance, we recommend that you use column store and compression when you create a table. For example, you can specify the following clause: `WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576)`. For more information, see [Greenplum Database official documentation on database table creation syntax](#).
- We recommend that you configure OSS and AnalyticDB for PostgreSQL instances within the same region to implement the best import performance.

- Export data in parallel

- i. Create a `WRITABLE` external table in AnalyticDB for PostgreSQL.
- ii. Execute the following statement to export data to OSS in parallel:

```
INSERT INTO <external table> SELECT * FROM <source table>
```

- Create OSS external tables

 **Note** The syntax to create and use external tables is the same as that of Greenplum Database, except for the syntax of location-related parameters.

```

CREATE [READABLE] EXTERNAL TABLE tablename
( columnname datatype [, ...] | LIKE othertable )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
  (( [HEADER]
    [DELIMITER [AS] 'delimiter' | 'OFF']
    [NULL [AS] 'null string']
    [ESCAPE [AS] 'escape' | 'OFF']
    [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
    [FILL MISSING FIELDS] ))
| 'CSV'
  (( [HEADER]
    [QUOTE [AS] 'quote']
    [DELIMITER [AS] 'delimiter']
    [NULL [AS] 'null string']
    [FORCE NOT NULL column [, ...]]
    [ESCAPE [AS] 'escape']
    [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
    [FILL MISSING FIELDS] ))
[ ENCODING 'encoding' ]
[ [LOG ERRORS [INTO error_table]] SEGMENT REJECT LIMIT count
  [ROWS | PERCENT] ]
CREATE WRITABLE EXTERNAL TABLE table_name
( column_name data_type [, ...] | LIKE other_table )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
  (( [DELIMITER [AS] 'delimiter']
    [NULL [AS] 'null string']
    [ESCAPE [AS] 'escape' | 'OFF'] ))
| 'CSV'
  (([QUOTE [AS] 'quote']
    [DELIMITER [AS] 'delimiter']
    [NULL [AS] 'null string']
    [FORCE QUOTE column [, ...]] ]
    [ESCAPE [AS] 'escape'] ))
[ ENCODING 'encoding' ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
ossprotocol:
  oss://oss_endpoint prefix=prefix_name
  id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
  oss://oss_endpoint dir=[folder/[folder/]...]/file_name
  id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
  oss://oss_endpoint filepath=[folder/[folder/]...]/file_name
  id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]

```

Parameters

Common parameters

Parameter	Description
Protocol and endpoint	<p>It is in the <code>protocol name://oss_endpoint</code> format. The protocol name is <code>oss</code>. <code>oss_endpoint</code> is the domain name used by users to access OSS in a region.</p> <p> Note You can access the database from a VPC host by using an internal endpoint containing "internal" in the name to avoid generating public traffic.</p>
id	The AccessKey ID of the OSS account.
key	The AccessKey secret of the OSS account.
bucket	The bucket where the data file is located. You must use OSS to create the bucket before data import.
prefix	<p>The prefix of the path name corresponding to the data file. Prefixes are directly matched and cannot be controlled by regular expressions. The <code>prefix</code>, <code>filepath</code>, and <code>dir</code> parameters are mutually exclusive and only one parameter can be specified at a time.</p> <ul style="list-style-type: none"> • If you create a <code>READABLE</code> external table for data import, all OSS files that contain the specified prefix will be imported. <ul style="list-style-type: none"> ◦ If you set <code>prefix</code> to <code>test/filename</code>, the following files will be imported: <ul style="list-style-type: none"> ▪ <code>test/filename</code> ▪ <code>test/filenameexxx</code> ▪ <code>test/filename/aa</code> ▪ <code>test/filenameyyy/aa</code> ▪ <code>test/filenameyyy/bb/aa</code> ◦ If you set <code>prefix</code> to <code>test/filename/</code>, only the following file out of the preceding files will be imported: <ul style="list-style-type: none"> <code>test/filename/aa</code> • if you create a <code>WRITABLE</code> external table for data export, each exported file will have a unique name based on this parameter. <p> Note One or more files can be exported for each compute node. The names of exported files are in the <code>prefix_tablename_uuid.x</code> format. <code>uuid</code> indicates a timestamp in microseconds as an <code>int64</code> value. <code>x</code> indicates the node ID. You can use an external table for multiple export operations. Each export operation is assigned to a <code>uuid</code> value. The files exported during each operation share a <code>uuid</code> value.</p>

Parameter	Description
dir	<p>The virtual folder path in OSS. The prefix, filepath, and dir parameters are mutually exclusive and only one parameter can be specified at a time.</p> <ul style="list-style-type: none"> A folder path must end with a forward slash (/) such as <code>test/mydir/</code>. If you use this parameter when creating an external table for data import, all files under the specified virtual directory (except for its subdirectories and contained files) will be imported. Unlike filepath, dir does not require you to specify the names of files in the directory. If this parameter is used in creating an external table for data export, all data will be exported to multiple files within the specified directory. The names of exported files are in the <code>filename.x</code> format, where x is a digit. The values of x may not be consecutive.
filepath	<p>The file name that contains a path in OSS. The prefix, filepath, and dir parameters are mutually exclusive and only one parameter can be specified at a time. You can only specify the filepath parameter when you create a READABLE external table for data import.</p> <ul style="list-style-type: none"> The file name includes the file path, but not the bucket name. The filename specified for data import must be in the <code>filename</code> or <code>filename.x</code> format. The values of x must be consecutive digits starting from 1. <p>For example, if filepath is set to filename and OSS contains the following files, the imported files include filename, filename.1, and filename.2, but filename.4 is not imported because filename.3 does not exist.</p> <pre>filename filename.1 filename.2 filename.4</pre>

Import mode parameters

Parameter	Description
async	<p>Specifies whether to load data asynchronously.</p> <ul style="list-style-type: none"> Asynchronous data import is enabled by default. You can set <code>async</code> to <code>false</code> or <code>f</code> to disable asynchronous data import. Enables the worker thread to load data from OSS to accelerate the import performance. The default import mode is asynchronous mode. Asynchronous data import consumes more hardware resources than normal data import.
compressiontype	<p>The compression format of the imported file. Valid values:</p> <ul style="list-style-type: none"> <code>none</code>: specifies to import files without compressing them. This is the default value. <code>gzip</code>: specifies compress imported files in the GZIP format. Only the GZIP format is supported.
compressionlevel	<p>The compression level of the files written to OSS. Valid values: 1 to 9. Default value: 6.</p>

Export mode parameters

Parameter	Description
oss_flush_block_size	The size of each data block written to OSS. Valid values: 1 MB to 128 MB. Default value: 32 MB.
oss_file_max_size	The maximum size of each file written to OSS. If the limit is exceeded, subsequent data is written to another file. Valid values: 8 MB to 4000 MB. Default value: 1024 MB.
num_parallel_worker	The number of parallel compression threads for data written to OSS. Valid values: 1 to 8. Default value: 3.

Additionally, you must pay attention to the following items for the export mode:

- **WRITABLE** is the keyword of the external table for data export. You must specify this keyword when creating an external table.
- Only the **prefix** and **dir** parameters are supported for data export. The **filepath** parameter is not supported.
- You can use the **DISTRIBUTED BY** clause to write data from compute nodes to OSS based on the specified distribution keys.

Other common parameters

The following error-tolerance parameters can be used for data import and export:

Error-tolerance parameters

Parameter	Description
oss_connect_timeout	The connection timeout period. Unit: seconds. Default value: 10.
oss_dns_cache_timeout	The DNS timeout period. Unit: seconds. Default value: 60.
oss_speed_limit	The minimum rate tolerated. Default value: 1024 bit/s (1 Kbit/s).
oss_speed_time	The maximum amount of time tolerated. Unit: seconds. Default value: 15.

If the default values are used for the preceding parameters, a timeout will occur when the transmission rate is lower than 1 Kbit/s for 15 consecutive seconds. For more information, see [Troubleshooting in OSS SDK reference](#).

The other parameters are compatible with the original external table syntax of Greenplum Database. For more information about the syntax, see [Greenplum Database official documentation on external table syntax](#). These parameters include:

- **FORMAT**: indicates the supported file format, such as **TEXT** and **CSV**.
- **ENCODING**: indicates the data encoding format of a file, such as **UTF-8**.
- **LOG ERRORS**: indicates that the clause can ignore imported erroneous data and write the data to `error_table`. You can also use the **count** parameter to specify the error reporting threshold.

Examples

```
# Create a READABLE external table of OSS.
create readable external table ossexample
  (date text, time text, open float, high float,
  low float, volume int)
  location('oss://oss-cn-hangzhou.aliyuncs.com
  prefix=osstest/example id=XXX
  key=XXX bucket=testbucket compressiontype=gzip')
```

```

FORMAT 'csv' (QUOTE '"' DELIMITER E'\t')
ENCODING 'utf8'
LOG ERRORS INTO my_error_rows SEGMENT REJECT LIMIT 5;
create readable external table ossexample
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
dir=osstest/ id=XXX
key=XXX bucket=testbucket')
FORMAT 'csv'
LOG ERRORS SEGMENT REJECT LIMIT 5;
create readable external table ossexample
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
filepath=osstest/example.csv id=XXX
key=XXX bucket=testbucket')
FORMAT 'csv'
LOG ERRORS SEGMENT REJECT LIMIT 5;
# Create a WRITABLE external table of OSS.
create WRITABLE external table ossexample_exp
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
prefix=osstest/exp/outfromhdb id=XXX
key=XXX bucket=testbucket') FORMAT 'csv'
DISTRIBUTED BY (date);
create WRITABLE external table ossexample_exp
(date text, time text, open float, high float,
low float, volume int)
location('oss://oss-cn-hangzhou.aliyuncs.com
dir=osstest/exp/ id=XXX
key=XXX bucket=testbucket') FORMAT 'csv'
DISTRIBUTED BY (date);
# Create a heap table named example to which you want to import data.
create table example
(date text, time text, open float,
high float, low float, volume int)
DISTRIBUTED BY (date);
# Import data to the example heap table from the ossexample table in parallel.
insert into example select * from ossexample;
# Export data from example to OSS in parallel
insert into ossexample_exp select * from example;
# As shown in the following execution plan, all compute nodes are involved in the task.
# All compute nodes read data from OSS in parallel. AnalyticDB for PostgreSQL performs a redistribution motion operation to compute the data by using a hash algorithm, and then distributes the data to its compute nodes after computing. After a compute node receives data, it performs an insert operation to add the data to AnalyticDB for PostgreSQL.

```

```

explain insert into example select * from ossexample;
          QUERY PLAN
-----
Insert (slice0; segments: 4) (rows=250000 width=92)
-> Redistribute Motion 4:4 (slice1; segments: 4) (cost=0.00..11000.00 rows=250000 width=92)
    Hash Key: ossexample.date
    -> External Scan on ossexample (cost=0.00..11000.00 rows=250000 width=92)
(4 rows)
# As shown in the following query plan, each compute node exports local data directly to OSS without redistributing the data.
explain insert into ossexample_exp select * from example;
          QUERY PLAN
-----
Insert (slice0; segments: 3) (rows=1 width=92)
-> Seq Scan on example (cost=0.00..0.00 rows=1 width=92)
(2 rows)
    
```

TEXT and CSV format description

The following parameters specify the formats of files read from and written to OSS. You can specify the parameters in the external DDL parameters.

- `\n`: a line delimiter or line break for TEXT and CSV files.
- **DELIMITER**: specifies the delimiter of columns.
 - If the **DELIMITER** parameter is specified, the **QUOTE** parameter must also be specified.
 - Recommended column delimiters include commas (`,`), vertical bars (`|`), `\t`, and other special characters.
- **QUOTE**: encloses user data that contains special characters by column.
 - Strings that contain special characters will be enclosed by **QUOTE** to differentiate user data from the control characters.
 - To optimize the efficiency, it is unnecessary to enclose data such as integers in **QUOTE** characters.
 - **QUOTE** cannot be the same string as specified in **DELIMITER**. The default value of **QUOTE** is double quotation marks (`"`).
 - User data that contains **QUOTE** characters must also contain **ESCAPE** characters to differentiate user data from machine code.
- **ESCAPE**: specifies the escape character.
 - Place an escape character before a special character that needs to be escaped to indicate that it is not a special character.
 - If **ESCAPE** is not specified, the default value is the same as **QUOTE**.
 - You can also use other characters as **ESCAPE** characters such as backslashes (`\`), which is used by MySQL.

Default control characters for TEXT and CSV files

Default control characters for TEXT and CSV files

Control character	TEXT	CSV
DELIMITER	<code>\t</code> (tab)	<code>,</code> (comma)
QUOTE	<code>"</code> (double quotation mark)	<code>"</code> (double quotation mark)
ESCAPE	N/A	Same as QUOTE

Control character	TEXT	CSV
NULL	\N (backslash-N)	Empty string without quotation marks

 **Note** All control characters must be single-byte characters.

SDK troubleshooting

The following **Error log information** table lists the error logs generated when an error occurs during the import or export process.

Error log information

Keyword	Description
code	The HTTP status code of the error request.
error_code	The error code returned by OSS.
error_msg	The error message returned by OSS.
req_id	The UUID used to identify the request. If you require assistance in solving a problem, you can submit a ticket containing the req_id of the failed request to OSS developers.

For more information, see . You can handle timeout-related errors by using parameters related to oss_ext.

References

- [Greenplum Database official documentation on external table syntax](#)
- [Greenplum Database official documentation on table creation syntax](#)

15.3.5.2. Import data from MySQL

You can use the mysql2pgsql tool to migrate tables from MySQL to AnalyticDB for PostgreSQL, Greenplum Database, PostgreSQL, or PPAS.

Background information

mysql2pgsql connects a source MySQL database to a destination AnalyticDB for PostgreSQL database, queries data to be exported from the MySQL database, and then imports the data to the destination database by using the \COPY statement. The tool supports multi-thread import. Each worker thread imports a part of database tables.

To download the binary installation package of mysql2pgsql, click [here](#).

To view instructions on source code compilation of mysql2pgsql, click [here](#).

Procedure

1. Modify the my.cfg configuration file to configure the connection information of source and destination databases.

i. Modify the connection information of the source MySQL database.

 **Note** You must have the read permissions on all user tables.

```
[src.mysql]
host = "192.168.1.1"
port = "3306"
user = "test"
password = "test"
db = "test"
encodingdir = "share"
encoding = "utf8"
```

ii. Modify the connection information of the destination PostgreSQL, PPAS, or AnalyticDB for PostgreSQL database.

 **Note** You must have the write permissions on the destination table.

```
[desc.pgsql]
connect_string = "host=192.168.1.2 dbname=test port=3432 user=test password=pgsql"
```

2. Import data by using mysql2pgsql.

```
./mysql2pgsql -l <tables_list_file> -d -n -j <number of threads> -s <schema of target table>
```

Parameters

Parameter	Description
-l	Optional. Used to specify a text file that contains tables to be synchronized. If you do not specify this parameter, all the tables in the database that is specified in the configuration file will be synchronized. <tables_list_file> is the name of a file that contains a collection of tables to be synchronized and conditions for table queries. The content format is as follows: <pre>table1 : select * from table_big where column1 < '2016-08-05'</pre> <pre>table2 :</pre> <pre>table3</pre> <pre>table4: select column1, column2 from tableX where column1 != 10</pre> <pre>table5: select * from table_big where column1 >= '2016-08-05'</pre>
-d	Optional. Indicates the table creation DDL statement that creates the destination table but does not synchronize data.
-n	Optional. Must be used along with -d to specify that the table partition definition is not included in the DDL statement.

Parameter	Description
-j	Optional. Used to specify the number of threads used for data synchronization. If you do not specify this parameter, five concurrent threads will be used by default.
-s	Optional. Used to specify the schema of the destination table. Only one schema at a time can be specified by the command. If you do not specify the parameter, the data is imported into the table under the public schema.

Typical usage

Full database migration

1. Obtain the DDL statements of the corresponding destination table by running the following command:

```
./mysql2pgsql -d
```

2. Create a table in the destination database based on these DDL statements with the distribution key information added.
3. Run the following command to synchronize all tables:

```
./mysql2pgsql
```

This command will migrate the data from all MySQL tables in the database that is specified in the configuration file to the destination database. By default, five concurrent threads are used to read and import data from involved tables.

Partial table migration

1. Create a new file tab_list.txt and enter the following content:

```
t1
t2 : select * from t2 where c1 > 138888
```

2. Run the following command to synchronize the specified t1 and t2 tables (note that for the t2 table, only data that meets the c1 > 138888 condition is migrated):

```
./mysql2pgsql -l tab_list.txt
```

15.3.5.3. Import data from PostgreSQL

You can use the pgsq2pgsql tool to migrate tables across AnalyticDB for PostgreSQL, Greenplum Database, PostgreSQL, and PPAS.

Context

pgsq2pgsql supports the following features:

- Full migration across PostgreSQL, PPAS, Greenplum Database, and AnalyticDB for PostgreSQL.
- Full migration and incremental migration from PostgreSQL or PPAS (version 9.4 or later) to AnalyticDB for PostgreSQL or ApsaraDB RDS for PPAS.

You can download the software packages from the [dbsync project](#) library.

- To download the binary installation package of pgsq2pgsql, click [here](#).
- To view instructions on source code compilation of pgsq2pgsql, click [here](#).

Procedure

1. Modify the my.cfg configuration file to configure the connection information of source and destination databases.

- i. Modify the connection information of the source PostgreSQL database.

 **Note** In the connection information of the source PostgreSQL database, we recommend that you set the user to the owner of the source database.

```
[src.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=3432 user=test password=pgsql"
```

- ii. Modify the connection information of the local temporary PostgreSQL database.

```
[local.pgsql]
connect_string = "host=192.168.1.2 dbname=test port=3432 user=test2 password=pgsql"
```

- iii. Modify the connection information of the destination PostgreSQL database.

 **Note** You must have the write permissions on the destination table.

```
[desc.pgsql]
connect_string = "host=192.168.1.2 dbname=test port=3432 user=test3 password=pgsql"
```

 **Note**

- If you need to synchronize incremental data, you must have the permissions to create replication slots in the source database.
- PostgreSQL versions 9.4 and later support logic flow replication, meaning that source databases of the versions support incremental migration. The kernel supports logic flow replication only if you configure the following kernel parameters:

```
wal_level = logical
max_wal_senders = 6
max_replication_slots = 6
```

2. Use `pgsql2pgsql` to perform full database migration. `./pgsql2pgsql`

By default, the migration program migrates the table data of all users from the source PostgreSQL database to the destination PostgreSQL database.

3. View the status information.

You can view the status information in a single migration process by connecting to the local temporary database. The information is stored in the `db_sync_status` table, including the start and end time of the full migration, the start time of the incremental migration, and the status of incremental synchronization.

15.3.5.4. Import data by using the `\COPY` statement

You can use the `\COPY` statement to import the data of local text files into AnalyticDB for PostgreSQL databases. The local text files must be formatted, such as files that use commas (,), semicolons (;), or special characters as delimiters.

Context

- Parallel writing of large amounts of data is not available because the `\COPY` statement writes data in serial using the coordinator node. If you need to import a large amount of data in parallel, you can use the OSS-based data import method.
- The `\COPY` statement is a psql instruction. If you use the database statement `COPY` instead of the `\COPY` statement, you must note that only `stdin` is supported. This `COPY` statement does not support file because

the root user does not have the superuser permissions to perform operations on files.

- AnalyticDB for PostgreSQL also allows you to use JDBC to execute the COPY statement. The CopyIn method is encapsulated within JDBC. For more information, see [Interface CopyIn](#).
- For more information about how to use the COPY statement, see [COPY](#).

Procedure

1. Import data by using the following sample code:

```
\COPY table [(column [, ...])] FROM {'file' | STDIN}
[ [WITH]
  [OIDS]
  [HEADER]
  [DELIMITER [ AS ] 'delimiter']
  [NULL [ AS ] 'null string']
  [ESCAPE [ AS ] 'escape' | 'OFF']
  [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
  [CSV [QUOTE [ AS ] 'quote']
    [FORCE NOT NULL column [, ...]]
  [FILL MISSING FIELDS]
  [[LOG ERRORS [INTO error_table] [KEEP]
  SEGMENT REJECT LIMIT count [ROWS | PERCENT] ]
\COPY {table [(column [, ...])] | (query)} TO {'file' | STDOUT}
[ [WITH]
  [OIDS]
  [HEADER]
  [DELIMITER [ AS ] 'delimiter']
  [NULL [ AS ] 'null string']
  [ESCAPE [ AS ] 'escape' | 'OFF']
  [CSV [QUOTE [ AS ] 'quote']
    [FORCE QUOTE column [, ...]] ]
[IGNORE EXTERNAL PARTITIONS ]
```

15.4. Databases

15.4.1. Overview

The operations based on the Greenplum Database in AnalyticDB for PostgreSQL are the same as those in the Greenplum Database, including schema, supported data types, and user permissions. Except for certain operations exclusive to the Greenplum Database (such as the partition keys and AO tables), you can refer to PostgreSQL for other operations.

References

- [Pivotal Greenplum Official Documentation](#)
- [Greenplum 4.3 Best Practices](#)
- [Golden Rules of Greenplum Data Distribution](#)

15.4.2. Create a database

After you log on to the AnalyticDB for PostgreSQL instance, you can execute SQL statements to create databases.

Similar to PostgreSQL, in AnalyticDB for PostgreSQL you can execute SQL statements to create databases. For example, after psql is connected to Greenplum, execute the following statements:

```
=> create database mygpdb;
CREATE DATABASE
=> \c mygpdb
psql (9.4.4, server 8.3devel)
You are now connected to database "mygpdb" as user "mygpdb".
```

15.4.3. Create a partition key

AnalyticDB for PostgreSQL is a distributed database and data is distributed across all the data nodes. You must create partition keys to distribute the data. The partition keys are vital to query performance. Partition keys are used to ensure even data distribution. Proper selection of keys can significantly improve query performance.

Specify a partition key

In AnalyticDB for PostgreSQL, tables can be distributed across all compute nodes in either hash or random mode. You must specify the partition key when creating a table. Imported data will be distributed to the specific compute node based on the hash value calculated by the partition key.

```
=> create table vtbl(id serial, key integer, value text, shape cuboid, location geometry, comment text) distributed by (key);
CREATE TABLE
```

If you do not specify the partition key (that means a statement without the `distributed by (key)` field), AnalyticDB for PostgreSQL will randomly allocate the ID field by using the round-robin algorithm.

Rules for selecting the partition key

- Select evenly distributed columns or multiple columns to prevent data skew.
- Select fields commonly used for connection operations, especially for highly concurrent statements.
- Select the condition columns that feature high concurrency queries and high filterability.
- Do not use random distribution.

15.4.4. Construct data

In some test scenarios, you must construct data to fill the database.

1. Create a function that generates random strings.

```
CREATE OR REPLACE FUNCTION random_string(integer) RETURNS text AS $body$
SELECT array_to_string(array
    (SELECT substr('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
        FROM (ceil(random()*62))::int
        FOR 1)
    FROM generate_series(1, $1)), '');
$body$
LANGUAGE SQL VOLATILE;
```

2. Create a partition key.

```
CREATE TABLE tbl(id serial, KEY integer, locate geometry, COMMENT text) distributed by (key);
```

3. Construct data.

```
INSERT INTO tbl(KEY, COMMENT, locate)
SELECT
  KEY,
  COMMENT,
  ST_GeomFromText(locate) AS locate
FROM
  (SELECT
    (a + 1) AS KEY,
    random_string(ceil(random() * 24)::integer) AS COMMENT,
    'POINT(' || ceil(random() * 36 + 99) || ' ' || ceil(random() * 24 + 50) || ')' AS locate
  FROM
    generate_series(0, 99999) AS a)
AS t;
```

15.4.5. Query data

This topic describes the query statements and how to view the query plans.

Query statement sample

```
=> select * from tbl where key = 751;
| id | key | value | shape | locate | comment |
+----+----+-----+-----+-----+-----+
| 751 | 751 | red | 010100000000000000000000C05B40000000000004A40 | B9hPhjeNWPqV |
(1 row)
Time: 513.101 ms
```

View a query plan

```
=> explain select * from tbl where key = 751;
Gather Motion 1:1 (slice1; segments: 1) (cost=0.00..1519.28 rows=1 width=53)
-> Seq Scan on tbl (cost=0.00..1519.28 rows=1 width=53)
    Filter: key = 751
Settings: effective_cache_size=8GB; gp_statistics_use_fkeys=on
Optimizer status: legacy query optimizer
```

15.4.6. Manage extensions

You can use extensions to expand database features. AnalyticDB for PostgreSQL enables you to manage extensions.

Extension types

AnalyticDB for PostgreSQL supports the following extensions:

- PostGIS: supports geographic information data.
- MADlib: supports the machine learning function library.
- fuzzystrmatch: supports the fuzzy matching of strings.
- orafunc: compatible with some Oracle functions.
- oss_ext: supports reading data from OSS.
- hll: collects statistics by using the HyperLogLog algorithm.
- pljava: supports compiling user-defined functions (UDF) in the PL/Java language.
- pgcrypto: supports cryptographic hash functions.
- intarray: supports integer array-related functions, operators, and indexes.

Create an extension

Execute the following statements to create an extension:

```
CREATE EXTENSION <extension name>;
CREATE SCHEMA <schema name>;
CREATE EXTENSION IF NOT EXISTS <extension name> WITH SCHEMA <schema name>;
```

Note

Before you create the MADlib extension, you must create the plpythonu extension first.

```
CREATE EXTENSION plpythonu;
CREATE EXTENSION madlib;
```

Delete an extension

Execute the following statements to delete an extension:

```
DROP EXTENSION <extension name>;
DROP EXTENSION IF EXISTS <extension name> CASCADE;
```

 **Note** If there are objects dependent on the extension, you must add the CASCADE keyword to delete all dependent objects.

15.4.7. Manage users and permissions

This topic describes how to manage users and permissions in AnalyticDB for PostgreSQL.

Manage users

The system prompts you to specify an initial username and password when you create an instance. This initial user is the root user. After the instance is created, you can use the root user account to connect to the database. The system also creates superusers such as aurora and replicator for internal management.

You can run the `\du+` command to view the information of all the users after you connect to the database by using the client tool of PostgreSQL or Greenplum. Example:

```
postgres=> \du+
                List of roles
Role name | Attributes | Member of | Description
-----+-----+-----+-----
root_user |           | rds_superuser
...
```

AnalyticDB for PostgreSQL does not provide superuser permissions, but offers a similar role, `RDS_SUPERUSER`, which is consistent with the permission system of ApsaraDB RDS for PostgreSQL. The root user (such as `root_user` in the preceding example) has the permissions of the `RDS_SUPERUSER` role. You can only identify this permission attribute by viewing the user description.

The root user has the following permissions:

- Can create databases and users and perform actions such as `LOGIN`, excluding the `SUPERUSER` permissions.
- Can view and modify the data tables of other users and perform actions such as `SELECT`, `UPDATE`, `DELETE`, and changing owners.
- Can view the connection information of other users, cancel their SQL statements, and kill their connections.
- Can create and delete extensions.
- Can create other users with `RDS_SUPERUSER` permissions. Example:

```
CREATE ROLE root_user2 RDS_SUPERUSER LOGIN PASSWORD 'xyz' ;
```

Manage permissions

You can manage permissions at the database, schema, and table levels. For example, if you want to grant read permissions on a table to a user and revoke their write permissions, you can execute the following statements:

```
GRANT SELECT ON TABLE t1 TO normal_user1;
REVOKE UPDATE ON TABLE t1 FROM normal_user1;
REVOKE DELETE ON TABLE t1 FROM normal_user1;
```

15.4.8. Manage JSON data

JavaScript Object Notation (JSON) has become a basic data type in the Internet and IoT fields. For more information about JSON, visit [JSON official website](#). PostgreSQL support for JSON has been well developed. Optimized by Alibaba Cloud, AnalyticDB for PostgreSQL supports the JSON type based on the PostgreSQL syntax.

Check whether the current version supports JSON

Execute the following statement to check whether the current version supports JSON:

```
=> SELECT ''::json;
```

If the following output is displayed, it indicates the JSON type is supported and the instance is ready for use. If the operation fails, restart the instance.

```
json
-----
""
(1 row)
```

If the following output is displayed, it indicates the JSON type is not supported.

```
ERROR: type "json" does not exist
LINE 1: SELECT ""::json;
          ^
```

The preceding command converts data from the string type to the JSON type. PostgreSQL supports operations on JSON data based on this conversion.

JSON conversion in the database

Database operations include reading and writing. The written data is typically converted from the string type to the JSON type. The contents of a string must meet the JSON standard, such as strings, digits, arrays, and objects. Example:

String

```
=> SELECT "hjson"::json;
 json
-----
 "hjson"
(1 row)
```

`::` is used for explicit type conversion in PostgreSQL, Greenplum, and AnalyticDB for PostgreSQL. The database calls the input function in JSON type during the conversion. Therefore, the JSON format check is performed as follows:

```
=> SELECT '{hjson:1024}'::json;
ERROR: invalid input syntax for type json
LINE 1: SELECT '{hjson:1024}'::json;
          ^
DETAIL:  Token "hjson" is invalid.
CONTEXT:  JSON data, line 1: {hjson...
=>
```

In the preceding example, `hjson` must be enclosed in double quotation marks (`" "`) because JSON requires the KEY value to be a string. A syntax error is returned when `{hjson:1024}` is entered.

Apart from explicit type conversion, database records can also be converted to JSON.

Typically, JSON is not used for a string or a digit, but an object that contains one or more key-value pairs. AnalyticDB for PostgreSQL can support most JSON scenarios after data is converted from the string type to objects. Example:

```
=> select row_to_json(row({'a':"a"}, 'b'));
 row_to_json
-----
 {"f1":{"a":"a"},"f2":"b"}
(1 row)
=> select row_to_json(row({'a':"a"}::json, 'b'));
 row_to_json
-----
 {"f1":{"a":"a"},"f2":"b"}
(1 row)
```

You can see the differences between the string and JSON here. The whole record is conveniently converted into the JSON type.

JSON data types

- Object

The object is the most frequently used data type in JSON. Example:

```
=> select '{"key":"value"}::json;
      json
-----
{"key":"value"}
(1 row)
```

- Integer and floating point number

JSON only supports three data types for numeric values: integer, floating point number, and constant expression. AnalyticDB for PostgreSQL supports all three types.

```
=> SELECT '1024'::json;
      json
-----
1024
(1 row)

=> SELECT '0.1'::json;
      json
-----
0.1
(1 row)
```

The following information is required in some special situations:

```
=> SELECT '1e100'::json;
      json
-----
1e100
(1 row)

=> SELECT '{"f":1e100}'::json;
      json
-----
{"f":1e100}
(1 row)
```

Extra-long numbers are also supported. Example:

```
=> SELECT '9223372036854775808'::json;
      json
-----
9223372036854775808
(1 row)
```

- Array

```
=> SELECT '[[1,2], [3,4,5]]':json;
      json
-----
 [[1,2], [3,4,5]]
(1 row)
```

Operators

Operators supported by JSON

```
=> select oprname,oprcode from pg_operator where oprleft = 3114;
oprname |      oprcode
-----+-----
-> | json_object_field
->> | json_object_field_text
-> | json_array_element
->> | json_array_element_text
#> | json_extract_path_op
#>> | json_extract_path_text_op
(6 rows)
```

Basic usage

```
=> SELECT '{"f":"1e100"}':json -> 'f';
? column?
-----
"1e100"
(1 row)
=> SELECT '{"f":"1e100"}':json ->> 'f';
? column?
-----
1e100
(1 row)
=> select '{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}':json#>array['f4','f6'];
? column?
-----
"stringy"
(1 row)
=> select '{"f2":{"f3":1},"f4":{"f5":99,"f6":"stringy"}}':json#>'f4,f6';
? column?
-----
"stringy"
(1 row)
=> select '{"f2":["f3",1],"f4":{"f5":99,"f6":"stringy"}}':json#>>'f2,0';
? column?
-----
f3
(1 row)
```

JSON functions

Supported JSON functions

```
postgres=# \df *json*
```

Schema	Name	Result data type	Argument data types	Type
pg_catalog	array_to_json	json	anyarray	normal
pg_catalog	array_to_json	json	anyarray, boolean	normal
pg_catalog	json_array_element	json	from_json json, element_index integer	normal
pg_catalog	json_array_element_text	text	from_json json, element_index integer	normal
pg_catalog	json_array_elements	SETOF json	from_json json, OUT value json	normal
pg_catalog	json_array_length	integer	json	normal
pg_catalog	json_each	SETOF record	from_json json, OUT key text, OUT value json	normal
pg_catalog	json_each_text	SETOF record	from_json json, OUT key text, OUT value text	normal
pg_catalog	json_extract_path	json	from_json json, VARIADIC path_elems text[]	normal
pg_catalog	json_extract_path_op	json	from_json json, path_elems text[]	normal
pg_catalog	json_extract_path_text	text	from_json json, VARIADIC path_elems text[]	normal
pg_catalog	json_extract_path_text_op	text	from_json json, path_elems text[]	normal
pg_catalog	json_in	json	cstring	normal
pg_catalog	json_object_field	json	from_json json, field_name text	normal
pg_catalog	json_object_field_text	text	from_json json, field_name text	normal
pg_catalog	json_object_keys	SETOF text	json	normal
pg_catalog	json_out	cstring	json	normal
pg_catalog	json_populate_record	anyelement	base anyelement, from_json json, use_json_as_text boolean	normal
pg_catalog	json_populate_recordset	SETOF anyelement	base anyelement, from_json json, use_json_as_text boolean	normal
pg_catalog	json_recv	json	internal	normal
pg_catalog	json_send	bytea	json	normal
pg_catalog	row_to_json	json	record	normal
pg_catalog	row_to_json	json	record, boolean	normal
pg_catalog	to_json	json	anyelement	normal

(24 rows)

Basic usage

```
=> SELECT array_to_json('{{1,5},{99,100}}':int[]);
```

```
array_to_json
```

```
-----  
[[1,5],[99,100]]
```

```
(1 row)
```

```
=> SELECT row_to_json(row(1,'foo'));
```

```
row_to_json
```

```
-----  
{"f1":1,"f2":"foo"}
```

```
(1 row)
```

```

=> SELECT json_array_length('[1,2,3,{"f1":1,"f2":[5,6]},4]');
 json_array_length
-----
          5
(1 row)
=> select * from json_each('{"f1":[1,2,3],"f2":{"f3":1},"f4":null,"f5":99,"f6":"stringy"}') q;
 key | value
-----+-----
 f1 | [1,2,3]
 f2 | {"f3":1}
 f4 | null
 f5 | 99
 f6 | "stringy"
(5 rows)
=> select json_each_text('{"f1":[1,2,3],"f2":{"f3":1},"f4":null,"f5":"null"}');
 json_each_text
-----
(f1,[1,2,3])
(f2,{"f3":1})
(f4,)
(f5,null)
(4 rows)
=> select json_array_elements('[1,true,[1,2,3],null,{"f1":1,"f2":[7,8,9]},false]');
 json_array_elements
-----
 1
 true
 [1,2,3]
 null
 {"f1":1,"f2":[7,8,9]}
 false
(6 rows)
create type jpop as (a text, b int, c timestamp);
=> select * from json_populate_record(null::jpop, '{"a":"blurfl","x":43.2}', false) q;
  a  | b | c
-----+-----
 blurfl |  |
(1 row)
=> select * from json_populate_recordset(null::jpop, [{"a":"blurfl","x":43.2}, {"b":3,"c":"2012-01-20 10:42:53"}], false) q;
  a  | b | c
-----+-----
 blurfl |  |
      | 3 | Fri Jan 20 10:42:53 2012
(2 rows)

```

Code examples

Create a table

```

create table tj(id serial, ary int[], obj json, num integer);
=> insert into tj(ary, obj, num) values('{1,5}::int[], '{"obj":1}', 5);
INSERT 0 1
=> select row_to_json(q) from (select id, ary, obj, num from tj) as q;
      row_to_json
-----
{"f1":1,"f2":[1,5],"f3":{"obj":1},"f4":5}
(1 row)
=> insert into tj(ary, obj, num) values('{2,5}::int[], '{"obj":2}', 5);
INSERT 0 1
=> select row_to_json(q) from (select id, ary, obj, num from tj) as q;
      row_to_json
-----
{"f1":1,"f2":[1,5],"f3":{"obj":1},"f4":5}
{"f1":2,"f2":[2,5],"f3":{"obj":2},"f4":5}
(2 rows)

```

Join multiple tables

```

create table tj2(id serial, ary int[], obj json, num integer);
=> insert into tj2(ary, obj, num) values('{2,5}::int[], '{"obj":2}', 5);
INSERT 0 1
=> select * from tj, tj2 where tj.obj->>'obj' = tj2.obj->>'obj';
 id | ary | obj | num | id | ary | obj | num
-----+-----+-----+-----+-----+-----+-----+-----
  2 | {2,5} | {"obj":2} | 5 | 1 | {2,5} | {"obj":2} | 5
(1 row)
=> select * from tj, tj2 where json_object_field_text(tj.obj, 'obj') = json_object_field_text(tj2.obj, 'obj');
 id | ary | obj | num | id | ary | obj | num
-----+-----+-----+-----+-----+-----+-----+-----
  2 | {2,5} | {"obj":2} | 5 | 1 | {2,5} | {"obj":2} | 5
(1 row)

```

Use the JSON function index

```

CREATE TEMP TABLE test_json (
  json_type text,
  obj json
);
=> insert into test_json values('aa', '{"f2":{"f3":1},"f4":{"f5":99,"f6":"foo"}}');
INSERT 0 1
=> insert into test_json values('cc', '{"f7":{"f3":1},"f8":{"f5":99,"f6":"foo"}}');
INSERT 0 1
=> select obj->'f2' from test_json where json_type = 'aa';
? column?
-----
{"f3":1}
(1 row)
=> create index i on test_json (json_extract_path_text(obj, 'f4'));
CREATE INDEX
=> select * from test_json where json_extract_path_text(obj, 'f4') = '{"f5":99,"f6":"foo"}';
 json_type |          obj
-----+-----
aa        | {"f2":{"f3":1},"f4":{"f5":99,"f6":"foo"}}
(1 row)

```

Note

JSON data cannot be used as the partition key and does not support JSON aggregate functions.

Example of using Python to access the database:

```

#!/bin/env python
import time
import json
import psycopg2
def gpquery(sql):
    conn = None
    try:
        conn = psycopg2.connect("dbname=sanity1x2")
        conn.autocommit = True
        cur = conn.cursor()
        cur.execute(sql)
        return cur.fetchall()
    except Exception as e:
        if conn:
            try:
                conn.close()
            except:
                pass
            time.sleep(10)
        print e
    return None
def main(_):
    sql = "select obj from tj;"
    #rows = Connection(host, port, user, pwd, dbname).query(sql)
    rows = gpquery(sql)
    for row in rows:
        print json.loads(row[0])
if __name__ == '__main__':
    main()

```

15.4.9. Use HyperLogLog

AnalyticDB for PostgreSQL is highly optimized by Alibaba Cloud, and not only has the features of Greenplum Database, but also supports HyperLogLog. It is suitable for industries such as Internet advertising and estimation analysis that require quick estimation of business metrics such as PV and UV.

Create a HyperLogLog extension

You can execute the following statement to create a HyperLogLog extension:

```
CREATE EXTENSION hll;
```

Basic types

- Execute the following statement to create a table containing the hll field:

```
create table agg (id int primary key,userids hll);
```

- Execute the following statement to convert int to hll_hashval:

```
select 1::hll_hashval;
```

Basic operators

- The hll type supports =, !=, <>, ||, and #.

```
select hll_add_agg(1::hll_hashval) = hll_add_agg(2::hll_hashval);
select hll_add_agg(1::hll_hashval) || hll_add_agg(2::hll_hashval);
select #hll_add_agg(1::hll_hashval);
```

- The hll_hashval type supports =, !=, and <>.

```
select 1::hll_hashval = 2::hll_hashval;
select 1::hll_hashval <> 2::hll_hashval;
```

Basic functions

- Hash functions such as Hll_hash_boolean, hll_hash_smallint, and hll_hash_bigint.

```
select hll_hash_boolean(true);
select hll_hash_integer(1);
```

- hll_add_agg: converts the int format to the hll format.

```
select hll_add_agg(1::hll_hashval);
```

- hll_union: aggregates the hll fields.

```
select hll_union(hll_add_agg(1::hll_hashval),hll_add_agg(2::hll_hashval));
```

- hll_set_defaults: sets the precision.

```
select hll_set_defaults(15,5,-1,1);
```

- hll_print: displays debug information.

```
select hll_print(hll_add_agg(1::hll_hashval));
```

Examples

```

create table access_date (acc_date date unique, userids hll);
insert into access_date select current_date, hll_add_agg(hll_hash_integer(user_id)) from generate_series(1,10000) t(u
ser_id);
insert into access_date select current_date-1, hll_add_agg(hll_hash_integer(user_id)) from generate_series(5000,2000
0) t(user_id);
insert into access_date select current_date-2, hll_add_agg(hll_hash_integer(user_id)) from generate_series(9000,4000
0) t(user_id);
postgres=# select #userids from access_date where acc_date=current_date;
? column?
-----
9725.85273370708
(1 row)
postgres=# select #userids from access_date where acc_date=current_date-1;
? column?
-----
14968.6596883279
(1 row)
postgres=# select #userids from access_date where acc_date=current_date-2;
? column?
-----
29361.5209149911
(1 row)

```

15.4.10. Use the CREATE LIBRARY statement

AnalyticDB for PostgreSQL introduces the CREATE LIBRARY and DROP LIBRARY statements to allow you to import custom software packages.

Syntax

```

CREATE LIBRARY library_name LANGUAGE [JAVA] FROM oss_location OWNER ownername
CREATE LIBRARY library_name LANGUAGE [JAVA] VALUES file_content_hex OWNER ownername
DROP LIBRARY library_name

```

Parameters

Parameter	Description
library_name	The name of the library to be installed. If the library to be installed has the same name as an existing library, you must delete the existing library before installing the new one.
LANGUAGE [JAVA]	The programming language to be used. Only PL/Java is supported.

Parameter	Description
oss_location	The location of the package. You can specify the OSS bucket and object names. Only one object can be specified and the specified object cannot be a compressed file. The format is as follows: <pre>oss://oss_endpoint filepath=[folder/[folder/]...]/file_name id=userossid key=userosskey bucket=ossbucket</pre>
file_content_hex	The content of the file. The byte stream is in hexadecimal notation. For example, 73656c6563742031 indicates the hexadecimal byte stream of "select 1". You can use this syntax to import packages without using OSS.
ownername	Specifies the user.
DROP LIBRARY	Deletes a library.

Examples

- Example 1: Install a JAR package named analytics.jar.

```
create library example language java from 'oss://oss-cn-hangzhou.aliyuncs.com filepath=analytics.jar id=xxx key=yy bucket=zzz';
```

- Example 2: Import the file content with the byte stream in hexadecimal notation.

```
create library pglib LANGUAGE java VALUES '73656c6563742031' OWNER "myuser";
```

- Example 3: Delete a library.

```
drop library example;
```

- Example 4: View installed libraries.

```
select name, lanname from pg_library;
```

15.4.11. Create and use the PL/Java UDF

AnalyticDB for PostgreSQL allows you to compile and upload JAR software packages written in PL/Java language, and use these JAR packages to create user-defined functions (UDFs). The PL/Java language supported by AnalyticDB for PostgreSQL is Community Edition PL/Java 1.5.0 and the JVM version is 1.8. This topic describes how to create a PL/Java UDF. For more information about PL/Java examples, see [PL/Java code](#). For more information about the compiling method, see [PL/Java documentation](#).

Procedure

1. In AnalyticDB for PostgreSQL, execute the following statement to create a PL/Java extension. You only need to execute the statement once for each database.

```
create extension pljava;
```

2. Compile the UDF based on your business needs. For example, you can use the following code to compile the Test.java file:

```
public class Test
{
    public static String substring(String text, int beginIndex,
        int endIndex)
    {
        try {
            Process process = null;
            process = Runtime.getRuntime().exec("ech Test running");
        } catch (Exception e) {
            return "" + e;
        }
        return text.substring(beginIndex, endIndex);
    }
}
```

3. Compile the manifest.txt file:

```
Manifest-Version: 1.0
Main-Class: Test
Specification-Title: "Test"
Specification-Version: "1.0"
Created-By: 1.7.0_99
Build-Date: 01/20/2016 21:00 AM
```

4. Run the following commands to compile and package the program:

```
javac Test.java
jar cfm analytics.jar manifest.txt Test.class
```

5. Upload the analytics.jar file generated in step 4 to OSS by using the following OSS console command. `osscli`

```
md put analytics.jar oss://zzz
```

6. In AnalyticDB for PostgreSQL, execute the CREATE LIBRARY statement to import the file to AnalyticDB for PostgreSQL:

```
create library example language java from 'oss://oss-cn-hangzhou.aliyuncs.com filepath=analytics.jar id=xxx key=yyy bucket=zzz';
```

 **Note** You can only use the filepath variable in the CREATE LIBRARY statement to import files one at a time. Additionally, the CREATE LIBRARY statement also supports byte streams to import files without using OSS. For more information, see [Use the CREATE LIBRARY statement](#).

7. In AnalyticDB for PostgreSQL, execute the following statements to create and use the UDF.

```
create table temp (a varchar) distributed randomly;
insert into temp values ('my string');
create or replace function java_substring(varchar, int, int) returns varchar as 'Test.substring' language java;
select java_substring(a, 1, 5) from temp;
```

15.5. Table

15.5.1. Create a table

You can create tables within your databases.

Syntax

The complete syntax for creating a table is as follows. Depending on your business needs, not all clauses will be required. Use the clauses that can fulfill your business needs.

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name (
  [ { column_namedata_type [ DEFAULT default_expr ]
    [column_constraint [ ... ]
  [ ENCODING ( storage_directive [,...] ) ]
]
  | table_constraint
  | LIKE other_table [{INCLUDING | EXCLUDING}
    {DEFAULTS | CONSTRAINTS}] ...}
[, ... ]
)
[ INHERITS ( parent_table [, ... ] ) ]
[ WITH ( storage_parameter=value [, ... ] ) ]
[ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
[ TABLESPACE tablespace ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
[ PARTITION BY partition_type (column)
  [ SUBPARTITION BY partition_type (column) ]
  [ SUBPARTITION TEMPLATE ( template_spec ) ]
  [...]
( partition_spec )
  | [ SUBPARTITION BY partition_type (column) ]
  [...]
( partition_spec
  [ ( subpartition_spec
    [(...)]
  ) ]
)
)
```

The column_constraint clause can be defined as follows:

```
[CONSTRAINT constraint_name]
  NOT NULL | NULL
| UNIQUE [USING INDEX TABLESPACE tablespace]
  [WITH ( FILLFACTOR = value )]
| PRIMARY KEY [USING INDEX TABLESPACE tablespace]
  [WITH ( FILLFACTOR = value )]
| CHECK ( expression )
| REFERENCES table_name [ ( column_name [, ... ] ) ]
  [ key_match_type ]
  [ key_action ]
```

The `storage_directive` clause of columns can be defined as follows:

```
COMPRESSTYPE={ZLIB | QUICKLZ | RLE_TYPE | NONE}
[COMPRESSLEVEL={0-9} ]
[BLOCKSIZE={8192-2097152} ]
```

The `storage_parameter` clause of tables can be defined as follows:

```
APPENDONLY={TRUE|FALSE}
BLOCKSIZE={8192-2097152}
ORIENTATION={COLUMN|ROW}
CHECKSUM={TRUE|FALSE}
COMPRESSTYPE={ZLIB|QUICKLZ|RLE_TYPE|NONE}
COMPRESSLEVEL={0-9}
FILLFACTOR={10-100}
OIDS={TRUE|FALSE}
```

The `table_constraint` clause can be defined as follows:

```
[CONSTRAINT constraint_name]
  UNIQUE ( column_name [, ... ] )
    [USING INDEX TABLESPACE tablespace]
    [WITH ( FILLFACTOR=value )]
| PRIMARY KEY ( column_name [, ... ] )
    [USING INDEX TABLESPACE tablespace]
    [WITH ( FILLFACTOR=value )]
| CHECK ( expression )
| FOREIGN KEY ( column_name [, ... ] )
    REFERENCES table_name [ ( column_name [, ... ] ) ]
    [ key_match_type ]
    [ key_action ]
    [ key_checking_mode ]
```

Valid values of `key_match_type`:

```
MATCH FULL
| SIMPLE
```

Valid values of `key_action`:

```
ON DELETE
| ON UPDATE
| NO ACTION
| RESTRICT
| CASCADE
| SET NULL
| SET DEFAULT
```

Valid values of `key_checking_mode`:

```
DEFERRABLE
| NOT DEFERRABLE
| INITIALLY DEFERRED
| INITIALLY IMMEDIATE
```

Valid values of `partition_type`:

```
LIST
| RANGE
```

The `partition_specification` clause can be defined as follows:

```
partition_element [, ...]
```

The `partition_element` clause can be defined as follows:

```
DEFAULT PARTITION name
| [PARTITION name] VALUES (list_value [,...])
| [PARTITION name]
  START ([datatype] 'start_value') [INCLUSIVE | EXCLUSIVE]
  [ END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE] ]
  [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
| [PARTITION name]
  END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE]
  [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
[ WITH ( partition_storage_parameter=value [, ... ] ) ]
[ TABLESPACE tablespace ]
```

The `subpartition_spec` or `template_spec` clause can be defined as follows:

```
subpartition_element [, ...]
```

The `subpartition_element` clause can be defined as follows:

```

DEFAULT SUBPARTITION name
| [SUBPARTITION name] VALUES (list_value [,...])
| [SUBPARTITION name]
  START ([datatype] 'start_value') [INCLUSIVE | EXCLUSIVE]
  [ END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE] ]
  [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
| [SUBPARTITION name]
  END ([datatype] 'end_value') [INCLUSIVE | EXCLUSIVE]
  [ EVERY ([datatype] [number | INTERVAL] 'interval_value') ]
[ WITH ( partition_storage_parameter=value [, ... ] ) ]
[ TABLESPACE tablespace ]

```

The `storage_parameter` clause of partitions can be defined as follows:

```

APPENDONLY={TRUE|FALSE}
BLOCKSIZE={8192-2097152}
ORIENTATION={COLUMN|ROW}
CHECKSUM={TRUE|FALSE}
COMPRESSTYPE={ZLIB|QUICKLZ|RLE_TYPE|NONE}
COMPRESLEVEL={1-9}
FILLFACTOR={10-100}
OIDS={TRUE|FALSE}

```

Parameters

The [Table creation parameters](#) table describes the key parameters for creating a table.

Table creation parameters

Parameter	Description
TABLE_NAME	The name of the table to be created.
column_name	The name of a column to be created in the new table.
data_type	The data type of the column. For columns that contain textual data, set the data type to VARCHAR or TEXT. We do not recommend the CHAR type.
DEFAULT default_expr	Specifies a default value for the column. The system will assign this default value to all columns that do not have a value. The default values can be any variable-free expression. Subqueries or cross-references to other columns in the table are not allowed. The data type of the default expression must match the data type of the column. If a column does not have a default value, the default value is null.
ENCODING storage_directive	Specifies the type of compression and block size for the column data. This clause is valid only for append-optimized, column-oriented tables. Column compression settings are inherited from the table level to the partition level to the sub-partition level. The lowest-level settings have priority over inherited settings.

Parameter	Description
INHERITS	Specifies that all columns in the new table automatically inherit a parent table. You can use INHERITS to create a persistent relationship between the new child table and its parent table. Schema modifications to the parent table are applied to the child table as well. When the parent table is also scanned, the data of the child table is scanned as well.
LIKE other_table	Specifies a table from which the new table automatically copies all column names, data types, NOT NULL constraints, and distribution policies. Storage properties such as append-optimized or partition structure are not copied. Unlike INHERITS, the new table is completely decoupled from the original table after the new table is created.
CONSTRAINT constraint_name	Configures a column or table constraint. When a constraint is violated, the constraint name will be displayed in the error message. Constraint names can be used to communicate helpful information to client applications. Constraint names that contain spaces must be enclosed by double quotation marks ("").
WITH (storage_option=value)	Configures storage options for the table or its indexes.
ON COMMIT	The operation that the system performs on the temporary tables at the end of a transaction. Valid values: <ul style="list-style-type: none"> • PRESERVE ROWS: No special action is taken. The data will be retained after the transaction is complete. The data will only be released when the session is disconnected. • DELETE ROWS: All rows in the temporary table are deleted. • DROP: The temporary table is deleted.
TABLESPACE tablespace	Specifies the name of the tablespace in which the new table is to be created. If not specified, the default tablespace of the database is used.
DISTRIBUTED BY	Specifies the distribution policy for the database. <ul style="list-style-type: none"> • DISTRIBUTED BY (column, [...]): specifies the partition key. The system uses hash distribution based on the distribution key. To evenly distribute data, you must set the partition key to the primary key of the table or a unique column or a set of columns. • DISTRIBUTED RANDOMLY: distributes data randomly. <div style="background-color: #e0f2f7; padding: 5px; margin-top: 10px;"> ? Note We recommend that you do not use random distribution. </div>
PARTITION BY	Configures the partition key to partition the table. Partitioning large tables improves data access efficiency. To partition a table is to create a top-level (parent) table and multiple lower-level (child) tables. A parent table is always empty when the partition table is created. Data is stored in the lowest-level child tables. In a multi-level partition table, data is only stored in the lowest-level sub-partitions. Valid values: RANGE, LIST, and a combination of the two.
SUBPARTITION BY	Configures a multi-level partitioned table.

Parameter	Description
SUBPARTITION TEMPLATE	You can specify a sub-partition template to create sub-partitions (lower-level child tables). This sub-partition template is applied to all parent partitions to ensure the same sub-partition structure.

Examples

Create a table and configure the partition key. The primary key is the default partition key in AnalyticDB for PostgreSQL.

```
CREATE TABLE films (
code    char(5) CONSTRAINT firstkey PRIMARY KEY,
title   varchar(40) NOT NULL,
did     integer NOT NULL,
date_prod date,
kind    varchar(10),
len     interval hour to minute
);

CREATE TABLE distributors (
did     integer PRIMARY KEY DEFAULT nextval('serial'),
name    varchar(40) NOT NULL CHECK (name <> '')
);
```

Create a compressed table and configure the partition key.

```
CREATE TABLE sales (txn_id int, qty int, date date)
WITH (appendonly=true, compresslevel=5)
DISTRIBUTED BY (txn_id);
```

Use sub-partition templates of each level and the default partition to create a three-level partition table.

```
CREATE TABLE sales (id int, year int, month int, day int,
region text)
DISTRIBUTED BY (id)
PARTITION BY RANGE (year)

SUBPARTITION BY RANGE (month)
SUBPARTITION TEMPLATE (
START (1) END (13) EVERY (1),
DEFAULT SUBPARTITION other_months )

SUBPARTITION BY LIST (region)
SUBPARTITION TEMPLATE (
SUBPARTITION usa VALUES ('usa'),
SUBPARTITION europe VALUES ('europe'),
SUBPARTITION asia VALUES ('asia'),
DEFAULT SUBPARTITION other_regions)

( START (2008) END (2016) EVERY (1),
DEFAULT PARTITION outlying_years);
```

15.5.2. Principles and scenarios of row store, column store, heap tables, and AO tables

AnalyticDB for PostgreSQL supports row store, column store, heap tables, and AO tables. This topic describes their principles and scenarios.

Row store and column store

Comparison

Dimension	Row store	Column store
Definition	Row store stores data in the form of rows. Each row is a tuple. To read a column, you must deform all of the columns that precede the target column. Because of this, the costs for accessing the first and the last columns are different.	Column store stores data as columns corresponding to a file or a batch of files. The cost of reading any column is the same. However, if you need to read multiple columns, you must access multiple files. The more columns you access, the higher the overheads are.
Compression ratio	Low.	High.
Cost of reading any column	Columns with larger column numbers cost more.	Same.
Vector computing and JIT architecture	Not suitable. Not suitable for batch computation.	Suitable. More efficient when accessing and obtaining statistics of a batch of data.

Dimension	Row store	Column store
Scenarios	<p>If you need to perform a large number of update and delete operations due to OLTP requirements such as when querying table details where multiple columns are returned, you can use row store.</p> <p>You can use partition tables if you have diversified requirements. For example, if you need to partition the data based on time, you can use row store to query the details of recent data and use column store to obtain more statistics from historical data.</p>	<p>You can use column store if you need data statistics because of the OLAP requirements.</p> <p>If you need a higher compression ratio, you can use column store.</p>

Heap tables

A heap table is heap storage. All changes to the heap table generate redo logs that can be used to restore data by time point. However, heap tables cannot implement logical incremental backup because any data block in the table may be changed and it is not convenient to record the position by using the heap storage.

Commit and redo logs are used to ensure reliability when transactions are finished. You can also implement redundancy by building secondary nodes through redo logs.

Append-optimized (AO) tables

AO tables are used to append data for storage. When you delete the updated data, you can use another bitmap file to mark the row to be deleted and use the bit and offset to determine whether a row is deleted.

When the transaction is finished, you must call the fsync function to record the offset of the data block that performs the last write operation. Even if the data block only contains one record, a new data block will be appended for the next transaction. The data block is synchronized to the secondary node for data redundancy.

AO tables are not suitable for small transactions because the fsync function is called at the end of each transaction, and this data block will not be reused even if there is space left.

AO tables are suitable for OLAP scenarios, batch data writing, high compression ratio, and logical backup that supports incremental backup. During backup, you only need to record the offset from the backup and the bitmap deletion mark for each full backup.

Usage scenarios of heap tables

- When multiple small transactions are handled, use a heap table.
- When you need to restore data by time point, use a heap table.

Usage scenarios of AO tables

- When you need to use column store, use an AO table.
- When data is written in batches, use an AO table.

15.5.3. Enable the column store and compression features

If you want to improve performance, speed up data import, or reduce costs for tables with infrequent updates and multiple fields, we recommend that you use column store and compression. This will increase the compression ratio threefold to ensure faster performance and import speed.

To enable the column store and compression features, you must specify the column store and compression options when creating a table. For example, you can add the following clause to the CREATE statement to enable the two features. For more information about the table creation syntax, see [Create a table](#).

with (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576, OIDS=false)

 **Note** AnalyticDB for PostgreSQL only supports zlib and RLE_TYPE compression algorithms. If you specify the quicklz algorithm, it is automatically converted to zlib.

15.5.4. Add a field to a column store table and set the default value

This topic describes how to add a field to a column store table and set the default value for the field, and how to use the ANALYZE statement to view the impact of updated data on the size of the column store table.

Context

In a column store table, each column is stored as a file, and two columns in the same row correspond to each other by using the offset. For example, if you add two fields of the INT8 type, you can quickly locate column B from column A by using the offset.

When you add the field, AO tables are not rewritten. If an AO table contains the records of deleted data, the added field must be filled with the deleted records before using the offset.

Procedure

1. Create three AO column store tables.

```
postgres=# create table tbl1 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
```

NOTICE: Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.

HINT: The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.

CREATE TABLE

```
postgres=# create table tbl2 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
```

NOTICE: Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.

HINT: The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.

CREATE TABLE

```
postgres=# create table tbl3 (id int, info text) with (appendonly=true, blocksize=8192, compresstype=none, orientation=column);
```

NOTICE: Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'id' as the Greenplum Database data distribution key for this table.

HINT: The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.

CREATE TABLE

2. Insert 10 million entries to the first two tables and 20 million entries to the third one.

```

postgres=# insert into tbl1 select generate_series(1,10000000),'test';
INSERT 0 10000000
postgres=# insert into tbl2 select generate_series(1,10000000),'test';
INSERT 0 10000000
postgres=# insert into tbl3 select generate_series(1,20000000),'test';
INSERT 0 20000000

```

3. Analyze the tables and display their sizes.

```

postgres=# analyze tbl1;
ANALYZE
postgres=# analyze tbl2;
ANALYZE
postgres=# analyze tbl3;
ANALYZE

postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
pg_size_pretty
-----
88 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl2'));
pg_size_pretty
-----
88 MB
(1 row)
postgres=# select pg_size_pretty(pg_relation_size('tbl3'));
pg_size_pretty
-----
173 MB
(1 row)

```

4. Update all the data in the first table. Display the table size after the update. The size is twice as large as the size before the update.

```

postgres=# update tbl1 set info='test';
UPDATE 10000000
postgres=# analyze tbl1;
ANALYZE
postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
pg_size_pretty
-----
173 MB
(1 row)

```

5. Add fields to the three tables and set the default values.

```
postgres=# alter table tbl1 add column c1 int8 default 1;
ALTER TABLE
postgres=# alter table tbl2 add column c1 int8 default 1;
ALTER TABLE
postgres=# alter table tbl3 add column c1 int8 default 1;
ALTER TABLE
```

6. Analyze the tables and view the table sizes.

```
postgres=# analyze tbl1;
ANALYZE
postgres=# analyze tbl2;
ANALYZE
postgres=# analyze tbl3;
ANALYZE

postgres=# select pg_size_pretty(pg_relation_size('tbl1'));
pg_size_pretty
-----
325 MB
(1 row)

postgres=# select pg_size_pretty(pg_relation_size('tbl2'));
pg_size_pretty
-----
163 MB
(1 row)

postgres=# select pg_size_pretty(pg_relation_size('tbl3'));
pg_size_pretty
-----
325 MB
(1 row)
```

When you add fields to the AO tables, the number of entries in the existing files will prevail. Even if all the entries are deleted, you must initialize the original data in the newly added fields.

15.5.5. Configure the table partition

For fact tables or large-sized tables in the database, we recommend that you configure table partitions.

Configure the table partition

You can use the table partitioning feature to delete data by using the `ALTER TABLE DROP PARTITION` statement to delete all the data in a partition, and import data by using the `ALTER TABLE EXCHANGE PARTITION` statement to add a new data partition on a regular basis.

AnalyticDB for PostgreSQL supports range partitioning, list partitioning, and composite partitioning. Range partitioning only supports partitioning by fields of the numeric or datetime data types.

The following example shows a table that uses range partitioning.

```

CREATE TABLE LINEITEM (
L_ORDERKEY          BIGINT NOT NULL,
L_PARTKEY           BIGINT NOT NULL,
L_SUPPKEY           BIGINT NOT NULL,
L_LINENUMBER       INTEGER,
L_QUANTITY          FLOAT8,
L_EXTENDEDPRICE    FLOAT8,
L_DISCOUNT        FLOAT8,
L_TAX              FLOAT8,
L_RETURNFLAG       CHAR(1),
L_LINESTATUS       CHAR(1),
L_SHIPDATE         DATE,
L_COMMITDATE       DATE,
L_RECEIPTDATE      DATE,
L_SHIPINSTRUCT     CHAR(25),
L_SHIPMODE         CHAR(10),
L_COMMENT          VARCHAR(44)
) WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576, OIDS
=false) DISTRIBUTED BY (L_orderkey)
PARTITION BY RANGE (L_SHIPDATE) (START (date '1992-01-01') INCLUSIVE END (date '2000-01-01') EXCLUSIVE EVERY (INT
ERVAL '1 month' ));

```

Principles of table partitioning

The purpose of partitioning is to minimize the amount of data that needs to be scanned during a query, so partitions must be associated with the query conditions.

- Principle 1: Select the fields related to the query conditions to configure partitions and reduce the amount of data to be scanned.
- Principle 2: When multiple query conditions exist, configure sub-partitions to further reduce the amount of data to be scanned.

15.5.6. Configure the sort key

A sort key is an attribute of a table. Data on disks is stored in the order of the sort key.

Context

Sort keys have two major advantages:

- Speed up and optimize column-store operations. The min and max meta information the system collects seldom overlaps with each other, which features good filterability.
- Eliminate the need to perform ORDER BY and GROUP BY operations. The data directly read from the disk is ordered as required by the sorting conditions.

Create a table

Command: CREATE TABLE

Description: define a new table

Syntax:

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name (
[ { column_name data_type [ DEFAULT default_expr ] [column_constraint [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
| table_constraint
| LIKE other_table [{INCLUDING | EXCLUDING}
{DEFAULTS | CONSTRAINTS}] ...}
[, ... ] ]
[column_reference_storage_directive [, ] ]
)
[ INHERITS ( parent_table [, ... ] ) ]
[ WITH ( storage_parameter=value [, ... ] )
[ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
[ TABLESPACE tablespace ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
[ SORTKEY (column, [ ... ] ) ]
[ PARTITION BY partition_type (column)
[ SUBPARTITION BY partition_type (column) ]
[ SUBPARTITION TEMPLATE ( template_spec ) ]
[... ]
( partition_spec )
| [ SUBPARTITION BY partition_type (column) ]
[... ]
( partition_spec
[ ( subpartition_spec
[ (... ) ]
) ]
) ]
)
```

Examples:

```
create table test(date text, time text, open float, high float, low float, volume int) with(APPENDONLY=true,ORIENTATIO
N=column) sortkey (volume);
```

Sort the table

```
VACUUM SORT ONLY [tablename]
```

Modify the sort key

This statement only modifies the catalog and does not sort data. You must execute the `VACUUM SORT ONLY` statement to sort the table.

```
ALTER [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name SET SORTKEY (column, [ ... ] )
```

Examples:

```
alter table test set sortkey (high,low);
```

15.6. Best practices

15.6.1. Configure memory and load parameters

You must configure memory and load parameters to improve database stability.

Background information

AnalyticDB for PostgreSQL is an MPP database with high computational and resource requirements. It consumes all of the resources provided to it, allowing AnalyticDB for PostgreSQL to have higher processing speeds but making it very easy to reach its limits.

The worst-case scenario in the event of the CPU, network, or hard disk exceeding its limits is a hardware bottleneck. However, in the event that memory is completely consumed, the database may crash.

How to avoid OOM errors

Out of memory (OOM) indicates that the system is unable to provide sufficient memory requested by a process. The following prompt appears when OOM errors occur:

```
Out of memory (seg27 host.example.com pid=47093) VM Protect failed to allocate 4096 bytes, 0 MB available
```

Causes

Possible causes of the OOM error include:

- The memory of the database node is insufficient.
- Kernel parameters related to the memory of the operating system are incorrectly configured.
- Data skew has occurred, causing a compute node to request a large amount of memory.
- Query skew has occurred. For example, if the grouping fields of some aggregate or window functions are not distribution keys, the data must be redistributed. After redistribution, data will be skewed in a certain computer node and result in the node requesting a large amount of memory.

Solutions

1. Modify the queries to request less memory.
2. Use the resource queue provided by AnalyticDB for PostgreSQL to limit the number of concurrent queries. Reduce the number of queries executed within the cluster at the same time to reduce the overall memory requested by the system.
3. Reduce the number of compute nodes deployed on a host. For example, deploy 8 compute nodes instead of 16 compute nodes on a host with 128 GB of memory. This allows each compute node to use twice the amount of memory compared with the latter.
4. Increase the memory of a host.
5. Set the `gp_vmem_protect_limit` parameter to limit the maximum VMEM that can be used by a single compute node. The memory size of a single host and the number of compute nodes deployed on the host determine the maximum memory size that a single compute node can use on average.
6. For SQL statements that have unpredictable memory usage, you can set the `statement_mem` parameter in the session to limit the memory usage of a single SQL statement, so as to prevent a single SQL statement from consuming all available memory.
7. Set the `statement_mem` parameter at the database level to apply to all the sessions in the database.
8. Use the resource queue to limit the maximum memory usage of the resource group. Add database users to the resource group to limit the overall memory used by these users.

Configure memory-related parameters

Properly configuring the operating system, database parameters, and resource queue can effectively reduce the probability of OOM.

When calculating the average memory usage of a single compute node on a single host, you must consider both the primary and secondary compute nodes. When the cluster encounters a host failure, the system will switch the service from primary compute nodes to the corresponding secondary compute nodes. During this time, the number of compute nodes on the host will be greater than usual. Therefore, you must consider the number of resources that will be occupied by the secondary compute nodes during failover.

The following tables describe how to configure parameters of the operating system kernel and database to avoid OOM.

The following **Operating system kernel parameters** table describes the parameter configuration of the operating system kernel.

Operating system kernel parameters

Parameter	Description
huge page	Do not configure the huge page parameter of the system. AnalyticDB for PostgreSQL does not support the latest version of PostgreSQL and therefore does not support the huge page feature. The huge page parameter locks a part of the allocated memory. Database nodes will not be able to use this part of the memory.
vm.overcommit_memory	<p>If you use the swap space, set this parameter to 2. If you do not use the swap space, set this parameter to 0.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0: The requested memory space cannot exceed the difference between the total memory and the resident set size (RSS). An error is returned only when the memory has been exceeded. 1: Most processes use the malloc function to apply for the memory, but do not use all the memory applied. When this parameter is set to 1, the memory requested by the malloc function will be allocated under any circumstances unless there is not sufficient memory. 2: The swap space is also considered when the system calculates the memory space that can be applied for. You can apply for a large amount of memory even if the swap space is triggered.
overcommit_ratio	<p>The larger the value, the more memory that process can apply for and the less that will be reserved for the operating system. For the formula used to calculate the memory parameters, see Examples to calculate the memory parameters.</p> <p>When this parameter is set to 2, the memory address that can be applied for cannot exceed $\text{swap} + \text{memory} \times \text{overcommit_ratio}$.</p>

The following **Database parameters** table describes the parameter configuration of the database.

Database parameters

Parameter	Description
gp_vmem_protect_limit	Specifies the maximum amount of memory that all processes can apply for on each node. If the value is too large, it may result in a system OOM error or even more serious problems. If the value is too small, SQL statements may not be executed even when the system has enough memory.

Parameter	Description
runaway_detector_activation_percent	<p>Default value: 90. This value is specified as a percentage. When the memory used by any compute node exceeds $\text{runaway_detector_activation_percent} \times \text{gp_vmem_protect_limit}/100$, the query is terminated to prevent OOM.</p> <p>The termination starts from the query that occupies the maximum memory until the memory reaches a value lower than $\text{runaway_detector_activation_percent} \times \text{gp_vmem_protect_limit}/100$.</p> <p>You can use the <code>gp_toolkit.session_level_memory_consumption</code> view to observe the memory usage of each session and runaway information.</p>
statement_mem	<p>Specifies the maximum amount of memory that a single SQL statement can apply. When the maximum memory is exceeded, spill files are created. Default value: 125. Unit: MB.</p> <p>We recommend that you set this parameter according to the following formula:</p> <div style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> $(\text{gp_vmem_protect_limit} \times 0.9) / \text{max_expected_concurrent_queries}$ </div> <div style="background-color: #e0f2f7; padding: 10px; margin: 10px 0;"> <p> Note</p> <ul style="list-style-type: none"> You can specify the <code>statement_mem</code> parameter in a session. If the current concurrency is low and a session needs to run a query that requires a large amount of memory, you must specify this parameter in the session. <code>Statement_mem</code> is suitable for limiting memory usage in low concurrency scenarios. If you use <code>statement_mem</code> to limit the memory for high concurrency scenarios, each query is allocated with a very small amount of memory. As a result, the performance of a small number of queries with high memory requirements in high concurrency scenarios is affected. We recommend that you use the resource queue to limit the maximum memory usage in high concurrency scenarios. </div>
gp_workfile_limit_files_per_query	<p>Specifies the maximum number of spill files that can be created by each query. When the memory requested by the query exceeds the <code>statement_mem</code> limit, spill files (also known as work files) are created, which is similar to the swap space of the operating system. When the number of spill files used exceeds the limit, the query will be terminated.</p> <p>Default value: 0, which indicates that an unlimited number of spill files can be created.</p>
gp_workfile_compress_algorithm	<p>Specifies the compression algorithm for spill files. Valid values: none and zlib.</p> <p>Specifies the compression algorithm. The values optimize storage space or I/O by sacrificing CPU. You can set this parameter when the disk is insufficient or the spill files meet a write bottleneck.</p>

Examples to calculate the memory parameters

The environment is as follows:

- Host configuration:

Total RAM = 256 GB

SWAP = 64 GB

- Four hosts, each deployed with eight primary compute nodes and eight secondary compute nodes.

When a host fails, the eight primary compute nodes are distributed to the remaining three hosts. A single host can be deployed with at most three extra primary compute nodes from the failed host. A single host can be deployed with at most 11 primary compute nodes.

1. Calculate the total memory allocated to AnalyticDB for PostgreSQL by the operating system.

Reserve 7.5 GB and 5% of memory for the operating system and calculate the available memory for all applications, and divide the available memory by the empirical coefficient of 1.7.

```
gp_vmem = ((SWAP + RAM) - (7.5 GB + 0.05 × RAM))/1.7
         = ((64 + 256) - (7.5 + 0.05 × 256))/1.7
         = 176
```

2. Use the empirical coefficient of 0.026 to calculate `overcommit_ratio`.

```
vm.overcommit_ratio = (RAM - (0.026 × gp_vmem))/RAM
                    = (256 - (0.026 × 176))/256
                    = .982
```

Set `vm.overcommit_ratio` to 98.

3. Calculate `gp_vmem_protect_limit` (the protection parameter of the maximum memory usage for each compute node), and divide `gp_vmem` by `maximum_acting_primary_segments` (the number of primary compute nodes to be run on each other host after one host fails).

```
gp_vmem_protect_limit calculation
gp_vmem_protect_limit = gp_vmem/maximum_acting_primary_segments
                    = 176/11
                    = 16 GB
                    = 16384 MB
```

Configure the resource queue

You can use resource queues to limit the number of concurrent queries and the total memory usage. When a query is running, it is added to the corresponding queue and the resources used are recorded in the queue. The resource limit of the queue is applied to all sessions in the queue.

The resource queue in AnalyticDB for PostgreSQL is similar to `cgroup` in Linux.

The syntax to create a resource queue is as follows:

Command: CREATE RESOURCE QUEUE

Description: create a new resource queue for workload management

Syntax:

```
CREATE RESOURCE QUEUE name WITH (queue_attribute=value [, ... ])
```

where queue_attribute is:

ACTIVE_STATEMENTS=integer

[MAX_COST=float [COST_OVERCOMMIT={TRUE|FALSE}]]

[MIN_COST=float]

[PRIORITY={MIN|LOW|MEDIUM|HIGH|MAX}]

[MEMORY_LIMIT='memory_units']

```
| MAX_COST=float [ COST_OVERCOMMIT={TRUE|FALSE} ]
```

[ACTIVE_STATEMENTS=integer]

[MIN_COST=float]

[PRIORITY={MIN|LOW|MEDIUM|HIGH|MAX}]

[MEMORY_LIMIT='memory_units']

The [Resource queue creation parameters](#) table describes the parameters for creating the resource queue.

Resource queue creation parameters

Parameter	Description
ACTIVE_STATEMENTS	<p>The number of SQL statements that are allowed to run (in the active state) concurrently.</p> <p>The value -1 indicates an unlimited number of SQL statements can run concurrently.</p>

Parameter	Description
<p>MEMORY_LIMIT 'memory_units kB, MB or GB'</p>	<p>Specifies the maximum memory usage allowed by all SQL statements in the resource queue. The value -1 indicates unlimited memory usage, but it is easy to trigger OOM errors because it is limited by the database or system parameters mentioned in the preceding sections.</p> <p>The memory usage of SQL statements is limited by resource queues and parameters.</p> <ul style="list-style-type: none"> When the <code>gp_resqueue_memory_policy</code> parameter is set to <code>none</code>, the limit is the same as that in the Greenplum databases earlier than version 4.1. When the <code>gp_resqueue_memory_policy</code> parameter is set to <code>auto</code> and you have specified the <code>statement_mem</code> parameter for a session or at the database level, the allowed memory of a single query will exceed the <code>MEMORY_LIMIT</code> of the resource queue. <p>Example:</p> <pre>=> SET statement_mem='2GB'; => SELECT * FROM my_big_table WHERE column='value' ORDER BY id; => RESET statement_mem;</pre> <ul style="list-style-type: none"> The system parameter <code>max_statement_mem</code> can limit the maximum memory usage at the compute node level. The memory requested by a single query cannot exceed <code>max_statement_mem</code>. <p>You can modify the <code>statement_mem</code> parameter at the session level, but do not modify the <code>max_statement_mem</code> parameter. We recommend that you specify <code>max_statement_mem</code> as follows:</p> <pre>(segghost_physical_memory) / (average_number_concurrent_queries)</pre> <ul style="list-style-type: none"> When the <code>gp_resqueue_memory_policy</code> parameter is set to <code>eager_free</code>, it indicates that the query is divided into several stages and that the database allocates the memory requested in the current stage. For example, if a query requests 1 GB of memory in total but only needs 100 MB during each stage, the database will allocate 100 MB of memory to the query. You can use <code>eager_free</code> to reduce the possibility of insufficient memory for the query.
<p>MAX_COST float</p>	<p>The maximum cost of the queries that are allowed to execute concurrently by the resource group. The cost is the estimated total cost in the SQL execution plan.</p> <p>The value of the parameter can be specified as a floating-point number (such as 100.0) or an exponent (such as 1e+2). A value of -1 indicates the cost is unlimited.</p>
<p>COST_OVERCOMMIT boolean</p>	<p>Specifies whether the limit of <code>max_cost</code> can be exceeded when the system is idle. The value <code>TRUE</code> indicates the limit can be exceeded.</p>
<p>MIN_COST float</p>	<p>When the resources requested exceed the limit, the queries are queued. However, when the cost of a query is lower than the <code>min_cost</code>, the query can run without queuing.</p>
<p>PRIORITY= {MIN LOW MEDIUM HIGH MAX}</p>	<p>The priority of the current resource queue. When resources are insufficient, CPU resources are allocated to the resource queue with a higher priority. The SQL statements in the resource queue with a higher priority can obtain CPU resources first. We recommend that you allocate users that initiate queries with high real-time requirements to resource queues with higher priority.</p> <p>This parameter is similar to the CPU resource group in the Linux <code>cgroup</code> and the time slice policy of real-time and common tasks.</p>

Example of modifying resource queue limits:

```
ALTER RESOURCE QUEUE myqueue WITH (MAX_COST=-1.0, MIN_COST= -1.0);
```

Example of putting the user in the resource queue:

```
ALTER ROLE sammy RESOURCE QUEUE poweruser;
```

The following table describes the parameters of resource queues.

Resource queue parameters

Parameter	Description
<code>gp_resqueue_memory_policy</code>	Specifies the memory management policy of the resource queue.
<code>gp_resqueue_priority</code>	Specifies whether to enable query prioritization. Valid values: <ul style="list-style-type: none"> On. Off. If this parameter is disabled, existing priority settings are not evaluated.
<code>gp_resqueue_priority_cpucore_per_segment</code>	Specifies the number of CPU cores allocated to each compute node. For example, if an 8-core host is configured with two primary compute nodes, you can set the parameter to 4. If there are no other nodes on the primary node, set the parameter to 8. When the CPU is preempted, the SQL statements running in the resource group with higher priority are allocated with CPU resources first.
<code>gp_resqueue_priority_sweeper_interval</code>	Specifies the interval at which CPU utilization is recalculated for all active statements. The share value is calculated when the SQL statement is executed. You can calculate the share value based on the priority and <code>gp_resqueue_priority_cpucore_per_segment</code> . The smaller the value and the more frequent the calculation, the better the result brought by the priority settings and the larger the overhead.

Tips for configuring resource queues:

- We recommend that you create a resource queue for each user.

The default resource queue of AnalyticDB for PostgreSQL is `pg_default`. If no queue is created, all users are assigned to `pg_default`. This operation is not recommended. We recommend that you create a resource queue for each user. Typically, a database user corresponds to a business. Different database users may correspond to different businesses or users, such as business users, analysts, developers, and DBAs.

- We do not recommend that you use superusers to execute queries.

Queries initiated by superusers are only limited by the preceding parameters and not by the resource queue. We do not recommend that you use superusers to execute queries if you want to use resource queues to limit the use of resources.

- `ACTIVE_STATEMENTS` indicates the SQL statements that can be executed concurrently within the resource queue. When the cost of a query is lower than the `min_cost`, the query can run without queuing.
- You can specify the `MEMORY_LIMIT` parameter to set the allowed maximum memory usage of all the SQL statements in a resource queue. The `statement_mem` parameter has higher priority that can break through the limit of resource queues.

 **Note** The memory of all resource queues cannot exceed `gp_vmem_protect_limit`.

- You can distinguish businesses by configuring the priorities of resource queues.

For example, assume that report forms have top priority, while common businesses and analysts have lower priorities. In this case, you can create three resource queues with the max, high, and medium priorities, respectively.

- If the number of resources requested at different times vary, you can use the `crontab` command to adjust the limits of resource queues periodically based on usage patterns.

For example, the queue of analysts has top priority during the day, while the queue of forms has lower priority at night. AnalyticDB for PostgreSQL does not support resource limits by time period. Therefore, you can only deploy tasks externally by using the `ALTER RESOURCE QUEUE` statement.

- You can use the view provided by `gp_toolkit` to observe the resource usage of the resource queues.

```
gp_toolkit.gp_resq_activity
```

```
gp_toolkit.gp_resq_activity_by_queue
```

```
gp_toolkit.gp_resq_priority_backend
```

```
gp_toolkit.gp_resq_priority_statement
```

```
gp_toolkit.gp_resq_role
```

```
gp_toolkit.gp_resqueue_status
```

16.KVStore for Redis

16.1. What is KVStore for Redis?

KVStore for Redis is a key-value storage database service that is compatible with open source Redis protocols. KVStore for Redis supports various data types, such as strings, lists, sets, sorted sets, and hash tables. The service also provides advanced features, such as transactions, message subscription, and message publishing.

You can easily deploy and manage KVStore for Redis databases in the KVStore for Redis console.

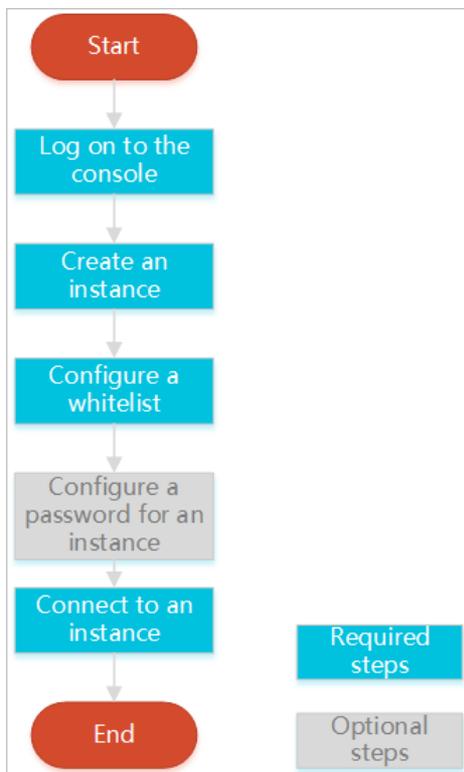
- You can create an instance to initialize a database.
- Before you use a KVStore for Redis instance, add IP addresses or CIDR blocks that are used to access the database to the instance whitelist.
- You can manage instances in the KVStore for Redis console.
- To secure data, you can periodically or immediately back up or restore databases in the KVStore for Redis console.
- You can log on to a database by using a client and then execute SQL statements to perform database operations.

16.2. Quick Start

16.2.1. Get started with KVStore for Redis

This topic describes all operations on an instance, from creating an instance to logging on to a database. This allows you to understand how to create and manage an instance.

The following figure shows the flowchart of managing a KVStore for Redis instance.



- **Log on to the KVStore for Redis console**
This topic describes how to log on to the KVStore for Redis console.
- **Create an instance**

KVStore for Redis supports two types of networks: classic network and Virtual Private Cloud (VPC). You can create KVStore for Redis instances of one of these network types.

- **Configure a whitelist**

Before you use a KVStore for Redis instance, add IP addresses or CIDR blocks that are used to access the database to the instance whitelist to enhance database security and stability.

- If you have not specified a password when you create the instance, set the password of the instance on the **Instance Information** page.

- **Connect to the instance**

You can use a client that supports Redis protocols or use the Redis command-line interface (redis-cli) tool to connect to the KVStore for Redis instance.

16.2.2. Log on to the KVStore for Redis console

This topic describes how to log on to the KVStore for Redis console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > KVStore for Redis**.

16.2.3. Create an instance

This topic describes how to create an instance in the KVStore for Redis console.

Prerequisites

To use the Virtual Private Cloud (VPC) service, you must create a VPC network in the same region where you create the KVStore for Redis instance.

 **Note** The network type cannot be changed after the instance is created.

Procedure

1. **Log on to the KVStore for Redis console.**
2. Click **Create Instance** in the upper-right corner.

3. Set the following parameters.

KVStore for Redis instance parameters

Section	Parameter	Description
Basic	Organization	The organization to which the KVStore for Redis instance to be created belongs.
	Resource Set	The resource set to which the KVStore for Redis instance belongs.  Notice After you select a resource set, the KVStore for Redis instance is accessible only to the members of the specified resource set.
Region	Region	Specifies the region where the KVStore for Redis instance is deployed.
	Zone	Specifies the zone where the KVStore for Redis instance is deployed.
Specifications	Engine Version	The following engine versions are supported: <ul style="list-style-type: none"> Redis2.8 Redis4.0
	Architecture Type	The architecture type of the KVStore for Redis instance. KVStore for Redis provides cluster and standard architectures. The cluster architecture meets large-capacity or high-performance requirements. Native Redis databases run in a single-threading model. If your database does not require high performance, we recommend that you use a standard instance. For higher performance, select a cluster architecture.
	Node Type	The node type for the KVStore for Redis instance. KVStore for Redis supports the primary-secondary dual-node structure.
	Instance Type	The specification of the instance. The maximum connections and maximum internal network bandwidth vary among different instance specifications.
Network	Network Type	The network type of the instance. On the Apsara Stack, a classic network and a VPC network have the following differences: <ul style="list-style-type: none"> Classic network: Cloud services in a classic network are not isolated. Unauthorized access to a cloud service is blocked by only security groups or the service whitelist policy. VPC: A VPC helps you build an isolated network environment on Apsara Stack. You can customize the routing table, Classless Inter-Domain Routing (CIDR) blocks, and gateway of a VPC network. You can also migrate applications to the cloud by using a leased line or virtual private network (VPN) without service interruption to integrate your on-premises data center and cloud resources in a VPC network into a virtual data center.  Note Before you select the VPC type, create a VPC network. For more information, see <i>Create a VPC network</i> and <i>Create a VSwitch</i> in <i>VPC User Guide</i> .

Section	Parameter	Description
Password	Instance Name	Enter the name of the KVStore for Redis instance. <ul style="list-style-type: none"> ◦ The name must be 2 to 128 characters in length ◦ and can contain uppercase and lowercase letters, digits, underscores (_), and hyphens (-). It must start with a lowercase letter or Chinese character.
	Password Setting	You can select Now or Later .
	Password	Set a password used to connect to the instance. The password must follow these rules: <ul style="list-style-type: none"> ◦ The password must be 8 to 30 characters in length. ◦ The password must contain uppercase letters, lowercase letters, and digits. Special characters are not supported.
	Confirm Password	Enter the specified password again.

4. After you set the parameters, click **Submit**.

16.2.4. Configure a whitelist

Before using a KVStore for Redis instance, add IP addresses or CIDR blocks used to access the database to the instance whitelist to improve database security and stability.

Context

 **Note** A properly configured whitelist can guarantee the highest-level security protection for your KVStore for Redis instance. We recommend that you maintain the whitelist on a regular basis.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Whitelist Settings** in the left-side navigation pane.
4. On the **Whitelist Settings** page, proceed in either of the following ways:
 - To customize the whitelist group name, create a new whitelist group:
 - a. Click **Add a Whitelist Group** in the upper-right corner.
 - b. In the **Add a Whitelist Group** dialog box that appears, set **Group Name**.

 **Note** A group name must be 2 to 32 characters in length and contain lowercase letters, digits, or underscores (_). The group name must start with a lowercase letter and end with a letter or digit. You cannot change this name after you create the whitelist group.

- If you do not require a custom whitelist group, click **Modify** next to the target whitelist group.
5. In the **Add a Whitelist Group** or **Modify Whitelist of Group** dialog box that appears, proceed in either of the following ways:
 - Manually modify the **Whitelist of Group** field:

- a. In the **Whitelist of Group** field, enter the IP addresses or CIDR blocks that you can use to connect to the KVStore for Redis instance.

Manually modify the whitelist group

Note

- Set the whitelist to `0.0.0.0/0` to allow connections from all IP addresses.
- Set the whitelist to `127.0.0.1` to block connections from all IP addresses.
- Set the whitelist to a CIDR block to allow connections from the IP addresses within the CIDR block, such as `10.10.10.0/24`.
- When you enter multiple IP addresses or CIDR blocks, separate them with commas (,) and leave no space before or after each comma.
- You can add 1,000 or fewer IP addresses or CIDR blocks to each whitelist group.

- b. Click **OK**.

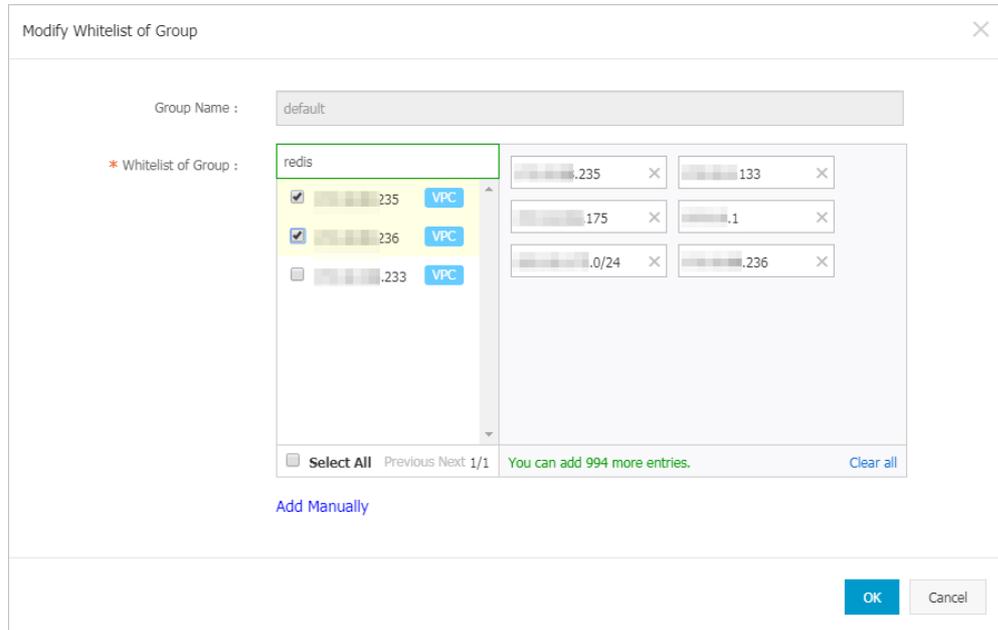
- o Load internal IP addresses of target ECS instances under the current Alibaba Cloud account:

- a. Click **Load ECS Internal IP Addresses**.

Load internal IP addresses of target ECS instances

b. Select internal IP addresses of target ECS instances.

Select internal IP addresses of target ECS instances



Note You can perform a fuzzy search by ECS instance name, ID, or IP address on the search bar above the list of ECS internal IP addresses.

c. Click OK.

16.2.5. Connect to an instance

16.2.5.1. Use a Redis client

You can connect to an KVStore for Redis instance by using clients for different programming languages.

The database service of KVStore for Redis is compatible with that of native Redis. Therefore, you can connect to both database services in similar ways. All clients that are compatible with the Redis protocol support connections to KVStore for Redis. You can use any of these clients that are suitable for your applications.

For more information about Redis clients, visit <https://redis.io/clients>.

Prerequisites

- The internal IP address of the Elastic Compute Service (ECS) instance or the public IP address of the local host has been added to a whitelist of the KVStore for Redis instance. For more information, see [Configure a whitelist](#).
- If you use a custom account to connect to the KVStore for Redis instance, the connection password must be in the format of `<user>:<password>`. For example, if the username of a custom account is `admin` and the password is `password`, the password used to connect to the KVStore for Redis instance must be in the format of `admin:password`.

Jedis clien

You can use a Jedis client to connect to KVStore for Redis in any of the following ways:

- Single Jedis connection. This method is not recommended because a client cannot automatically reconnect to KVStore for Redis after a connection times out.
- JedisPool-based connection. This method is recommended.

To use a Jedis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the Jedis client. For more information, see [Jedis](#).
2. Example of single Jedis connection
 - i. Open the Eclipse client, create a project, and then enter the following code:

```
import redis.clients.jedis.Jedis;
public class jedistest {
public static void main(String[] args) {
try {
    String host = "xx.kvstore.aliyuncs.com";//You can view the connection address of the target instance in the console.
    int port = 6379;
    Jedis jedis = new Jedis(host, port);
    //Authentication information.
    jedis.auth("password");//password
    String key = "redis";
    String value = "aliyun-redis";
    //Select a database. Default value: 0.
    jedis.select(1);
    //Set a key.
    jedis.set(key, value);
    System.out.println("Set Key " + key + " Value: " + value);
    //Obtain the configured key and value.
    String getvalue = jedis.get(key);
    System.out.println("Get Key " + key + " ReturnValue: " + getvalue);
    jedis.quit();
    jedis.close();
}
catch (Exception e) {
    e.printStackTrace();
}
}
}
```

- ii. Run the project. You have connected to KVStore for Redis if you see the following result in the Eclipse console.

```
Set Key redis Value aliyun-redis
Get Key redis ReturnValue aliyun-redis
```

Then, you can use a Jedis client to manage your KVStore for Redis instance. You can also connect to your KVStore for Redis instance by using JedisPool.

3. Example of JedisPool-based connection

- i. Open the Eclipse client, create a project, and then configure the following pom file:

```
<dependency>
<groupId>redis.clients</groupId>
<artifactId>jedis</artifactId>
<version>2.7.2</version>
<type>jar</type>
<scope>compile</scope>
</dependency>
```

- ii. Add the following application to the project:

```
import org.apache.commons.pool2.PooledObject;
import org.apache.commons.pool2.PooledObjectFactory;
import org.apache.commons.pool2.impl.DefaultPooledObject;
import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.HostAndPort;
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;
import redis.clients.jedis.JedisPoolConfig;
```

- iii. If your Jedis client version is Jedis-2.7.2, enter the following code in the project:

```
JedisPoolConfig config = new JedisPoolConfig();
//Maximum number of idle connections. You can customize this parameter. Make sure that the specified maximum number of idle connections does not exceed the maximum number of connections that the KVStore for Redis instance supports.
config.setMaxIdle(200);
//Maximum number of connections. You can customize this parameter. Make sure that the specified maximum number of connections does not exceed the maximum number of connections that the KVStore for Redis instance supports.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "Password";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
try {
jedis = pool.getResource();
/// ... do stuff here ... for example
jedis.set("foo", "bar");
String foobar = jedis.get("foo");
jedis.zadd("sose", 0, "car");
jedis.zadd("sose", 0, "bike");
Set<String> sose = jedis.zrange("sose", 0, -1);
} finally {
if (jedis != null) {
jedis.close();
}
}
/// ... when closing your application:
pool.destroy();
```

- iv. If your Jedis client version is Jedis-2.6 or Jedis-2.5, enter the following code in the project:

```
JedisPoolConfig config = new JedisPoolConfig();
//Maximum number of idle connections. You can customize this parameter. Make sure that the specified maximum number of idle connections does not exceed the maximum number of connections that the KVStore for Redis instance supports.
config.setMaxIdle(200);
//Maximum number of connections. You can customize this parameter. Make sure that the specified maximum number of connections does not exceed the maximum number of connections that the KVStore for Redis instance supports.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
String host = "*.aliyuncs.com";
String password = "Password";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
boolean broken = false;
try {
    jedis = pool.getResource();
    /// ... do stuff here ... for example
    jedis.set("foo", "bar");
    String foobar = jedis.get("foo");
    jedis.zadd("sose", 0, "car");
    jedis.zadd("sose", 0, "bike");
    Set<String> sose = jedis.zrange("sose", 0, -1);
}
catch(Exception e)
{
    broken = true;
} finally {
    if (broken) {
        pool.returnBrokenResource(jedis);
    } else if (jedis != null) {
        pool.returnResource(jedis);
    }
}
```

- v. Run the project. You have connected to KVStore for Redis if you see the following result in the Eclipse console.

```
Set Key redis Value aliyun-redis
Get Key redis ReturnValue aliyun-redis
```

Then, you can use a Jedis client to manage your KVStore for Redis instance.

PhpRedis client

To use a PhpRedis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the PhpRedis client. For more information, see [PhpRedis](#).

2. In an editor that supports PHP editing, enter the following code:

```
<? php
/* Replace the following parameter values with the host name and port number of the target instance. */
$host = "localhost";
$port = 6379;
/* Replace the following parameter values with the ID and password of the target instance. */
$user = "test_username";
$pwd = "test_password";
$redis = new Redis();
if ($redis->connect($host, $port) == false) {
    die($redis->getLastError());
}
if ($redis->auth($pwd) == false) {
    die($redis->getLastError());
}
/* You can perform database operations after authentication. For more information, visit https://github.com/php
Redis/phpredis. */.
if ($redis->set("foo", "bar") == false) {
    die($redis->getLastError());
}
$value = $redis->get("foo");
echo $value;
? >
```

3. Run the code. Then, you can use a PhpRedis client to connect to your KVStore for Redis instance. For more information, visit <https://github.com/phpredis/phpredis>.

Redis-py client

To use a redis-py client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install the redis-py client. For more information, see [redis-py](#).
2. In an editor that supports Python editing, enter the following code. You can use a redis-py client to connect to the KVStore for Redis instance and perform database operations.

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
import redis
#Replace the following parameter values with the host name and port number of the target instance.
host = 'localhost'
port = 6379
#Replace the following parameter value with the password of the target instance.
pwd = 'test_password'
r = redis.StrictRedis(host=host, port=port, password=pwd)
#You can perform database operations after you establish a connection. For more information, visit https://github.co
m/andymccurdy/redis-py.
r.set('foo', 'bar');
print r.get('foo')
```

C or C++ client

To use a C or C++ client to connect to an KVStore for Redis instance, perform the following steps:

1. Download, compile, and install the C client by using the following code:

```
git clone https://github.com/redis/hiredis.git
cd hiredis
make
sudo make install
```

2. Enter the following code in the C or C++ editor:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>
int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;
    if (argc < 4) {
        printf("Usage: example xxx.kvstore.aliyuncs.com 6379 instance_id password\n");
        exit(0);
    }
    const char *hostname = argv[1];
    const int port = atoi(argv[2]);
    const char *instance_id = argv[3];
    const char *password = argv[4];
    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    c = redisConnectWithTimeout(hostname, port, timeout);
    if (c == NULL || c->err) {
        if (c) {
            printf("Connection error: %s\n", c->errstr);
            redisFree(c);
        } else {
            printf("Connection error: can't allocate redis context\n");
        }
        exit(1);
    }
    /* AUTH */
    reply = redisCommand(c, "AUTH %s", password);
    printf("AUTH: %s\n", reply->str);
    freeReplyObject(reply);
    /* PING server */
    reply = redisCommand(c, "PING");
    printf("PING: %s\n", reply->str);
    freeReplyObject(reply);
    /* Set a key */
    reply = redisCommand(c, "SET %s %s", "foo", "hello world");
```

```

printf("SET: %s\n", reply->str);
freeReplyObject(reply);
/* Set a key using binary safe API */
reply = redisCommand(c,"SET %b %b", "bar", (size_t) 3, "hello", (size_t) 5);
printf("SET (binary API): %s\n", reply->str);
freeReplyObject(reply);
/* Try a GET and two INCR */
reply = redisCommand(c,"GET foo");
printf("GET foo: %s\n", reply->str);
freeReplyObject(reply);
reply = redisCommand(c,"INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* again ... */
reply = redisCommand(c,"INCR counter");
printf("INCR counter: %lld\n", reply->integer);
freeReplyObject(reply);
/* Create a list of numbers, from 0 to 9 */
reply = redisCommand(c,"DEL mylist");
freeReplyObject(reply);
for (j = 0; j < 10; j++) {
    char buf[64];
    snprintf(buf,64,"%d",j);
    reply = redisCommand(c,"LPUSH mylist element-%s", buf);
    freeReplyObject(reply);
}
/* Let's check what we have inside the list */
reply = redisCommand(c,"LRANGE mylist 0 -1");
if (reply->type == REDIS_REPLY_ARRAY) {
    for (j = 0; j < reply->elements; j++) {
        printf("%u) %s\n", j, reply->element[j]->str);
    }
}
freeReplyObject(reply);
/* Disconnects and frees the context */
redisFree(c);
return 0;
}

```

3. Compile the code.

```
gcc -o example -g example.c -I /usr/local/include/hiredis -lhiredis
```

4. Run a test.

```
example xxx.kvstore.aliyuncs.com 6379 instance_id password
```

Now, the C or C++ client is connected to the KVStore for Redis instance.

.NET client

To use a .NET client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and use the .NET client.

```
git clone https://github.com/ServiceStack/ServiceStack.Redis
```

2. Create a .NET project on the .NET client.
3. Add the reference file stored in the library file directory `ServiceStack.Redis/lib/tests` to the client.
4. Enter the following code in the .NET project to connect to the KVStore for Redis instance. For more information about API operations, visit <https://github.com/ServiceStack/ServiceStack.Redis>.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ServiceStack.Redis;
namespace ServiceStack.Redis.Tests
{
    class Program
    {
        public static void RedisClientTest()
        {
            string host = "127.0.0.1"; /*IP address of the host that you want to connect to*/
            string password = "password"; /*Password*/
            RedisClient redisClient = new RedisClient(host, 6379, password);
            string key = "test-aliyun";
            string value = "test-aliyun-value";
            redisClient.Set(key, value);
            string listKey = "test-aliyun-list";
            System.Console.WriteLine("set key " + key + " value " + value);
            string getValue = System.Text.Encoding.Default.GetString(redisClient.Get(key));
            System.Console.WriteLine("get key " + key + " value " + getValue);
            System.Console.Read();
        }
        public static void RedisPoolClientTest()
        {
            string[] testReadWriteHosts = new[] {
                "redis://password@127.0.0.1:6379" /*redis://Password@IP address that you want to connect to:Port*/
            };
            RedisConfig.VerifyMasterConnections = false; /*You must set the parameter.*/
            PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10 /*Number of connections in the pool*/, 10 /*Connection pool timeout value*/, testReadWriteHosts);
            for (int i = 0; i < 100; i++){
                IRedisClient redisClient = redisPoolManager.GetClient(); /*Obtain the connection.*/
                RedisNativeClient redisNativeClient = (RedisNativeClient)redisClient;
                redisNativeClient.Client = null; /*KVStore for Redis does not support the CLIENT SETNAME command. Set Client to null.*/
            }
        }
    }
}
```

```

try
{
    string key = "test-aliyun1111";
    string value = "test-aliyun-value1111";
    redisClient.Set(key, value);
    string listKey = "test-aliyun-list";
    redisClient.AddItemToList(listKey, value);
    System.Console.WriteLine("set key " + key + " value " + value);
    string getValue = redisClient.GetValue(key);
    System.Console.WriteLine("get key " + getValue);
    redisClient.Dispose();//
}
catch (Exception e)
{
    System.Console.WriteLine(e.Message);
}
}
}
System.Console.Read();
}
static void Main(string[] args)
{
    //Single-connection mode.
    RedisClientTest();
    //Connection-pool mode.
    RedisPoolClientTest();
}
}
}

```

node-redis client

To use a node-redis client to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install a node-redis client.

```
npm install hiredis redis
```

2. Enter and run the following code on the node-redis client to connect to the KVStore for Redis instance.

```

var redis = require("redis"),
client = redis.createClient(<port>, <"host">, {detect_buffers: true});
client.auth("password", redis.print)

```

Note In the code, the port field specifies the port of the KVStore for Redis instance. Default value: 6379. The host field specifies the endpoint of the KVStore for Redis instance. The following example shows the settings of the port and host fields:

```
client = redis.createClient(6379, "r-abcdefg.redis.rds.aliyuncs.com", {detect_buffers: true});
```

3. Use the KVStore for Redis instance.

```

// Write data to the instance.
client.set("key", "OK");
// Query data on the instance. The instance returns data of the String type.
client.get("key", function (err, reply) {
  console.log(reply.toString()); // print `OK`
});
// If a buffer is imported, a buffer is returned.
client.get(new Buffer("key"), function (err, reply) {
  console.log(reply.toString()); // print ``
});
client.quit();

```

C# client StackExchange.Redis

To use the C# client StackExchange.Redis to connect to an KVStore for Redis instance, perform the following steps:

1. Download and install [StackExchange.Redis](#).
2. Add a reference.

```
using StackExchange.Redis;
```

3. Initialize ConnectionMultiplexer.

ConnectionMultiplexer is the core of StackExchange.Redis, and shared and reused in the entire application. You must use ConnectionMultiplexer as a singleton. ConnectionMultiplexer is initialized in the following way:

```

// redis config
private static ConfigurationOptions configurationOptions = ConfigurationOptions.Parse("127.0.0.1:6379,password=xxx,connectTimeout=2000");
//the lock for singleton
private static readonly object Locker = new object();
//singleton
private static ConnectionMultiplexer redisConn;
//singleton
public static ConnectionMultiplexer getRedisConn()
{
  if (redisConn == null)
  {
    lock (Locker)
    {
      if (redisConn == null || ! redisConn.IsConnected)
      {
        redisConn = ConnectionMultiplexer.Connect(configurationOptions);
      }
    }
  }
  return redisConn;
}

```

Note

ConfigurationOptions contains multiple options, such as keepAlive, connectRetry, and name. For more information, see [StackExchange.Redis.ConfigurationOptions](#).

4. `GetDatabase()` returns a lightweight object. You can obtain this object from the object of `ConnectionMultiplexer`.

```
redisConn = getRedisConn();
var db = redisConn.GetDatabase();
```

5. The following examples show five types of data structures, which are strings, hashes, lists, sets, and sorted sets. The API operations used in these examples are different from their usage in the native Redis service.

- string

```
//set get
string strKey = "hello";
string strValue = "world";
bool setResult = db.StringSet(strKey, strValue);
Console.WriteLine("set " + strKey + " " + strValue + ", result is " + setResult);
//incr
string counterKey = "counter";
long counterValue = db.StringIncrement(counterKey);
Console.WriteLine("incr " + counterKey + ", result is " + counterValue);
//expire
db.KeyExpire(strKey, new TimeSpan(0, 0, 5));
Thread.Sleep(5 * 1000);
Console.WriteLine("expire " + strKey + ", after 5 seconds, value is " + db.StringGet(strKey));
//mset mget
KeyValuePair<RedisKey, RedisValue> kv1 = new KeyValuePair<RedisKey, RedisValue>("key1", "value1");
KeyValuePair<RedisKey, RedisValue> kv2 = new KeyValuePair<RedisKey, RedisValue>("key2", "value2");
db.StringSet(new KeyValuePair<RedisKey, RedisValue>[] {kv1, kv2});
RedisValue[] values = db.StringGet(new RedisKey[] {kv1.Key, kv2.Key});
Console.WriteLine("mget " + kv1.Key.ToString() + " " + kv2.Key.ToString() + ", result is " + values[0] + "&&" + values[1]);
```

- hash

```
string hashKey = "myhash";
//hset
db.HashSet(hashKey,"f1","v1");
db.HashSet(hashKey,"f2","v2");
HashEntry[] values = db.HashGetAll(hashKey);
//hgetall
Console.WriteLine("hgetall " + hashKey + ", result is");
for (int i = 0; i < values.Length; i++)
{
    HashEntry hashEntry = values[i];
    Console.WriteLine(" " + hashEntry.Name.ToString() + " " + hashEntry.Value.ToString());
}
Console.WriteLine();
```

- list

```
//list key
string listKey = "myList";
//rpush
db.ListRightPush(listKey, "a");
db.ListRightPush(listKey, "b");
db.ListRightPush(listKey, "c");
//lrange
RedisValue[] values = db.ListRange(listKey, 0, -1);
Console.WriteLine("lrange " + listKey + " 0 -1, result is ");
for (int i = 0; i < values.Length; i++)
{
    Console.WriteLine(values[i] + " ");
}
Console.WriteLine();
```

- set

```
//set key
string setKey = "mySet";
//sadd
db.SetAdd(setKey, "a");
db.SetAdd(setKey, "b");
db.SetAdd(setKey, "c");
//sismember
bool isContains = db.SetContains(setKey, "a");
Console.WriteLine("set " + setKey + " contains a is " + isContains );
```

- sorted set

```

string sortedSetKey = "myZset";
//sadd
db.SortedSetAdd(sortedSetKey, "xiaoming", 85);
db.SortedSetAdd(sortedSetKey, "xiaohong", 100);
db.SortedSetAdd(sortedSetKey, "xiaofei", 62);
db.SortedSetAdd(sortedSetKey, "xiaotang", 73);
//zrevrangebyscore
RedisValue[] names = db.SortedSetRangeByRank(sortedSetKey, 0, 2, Order.Ascending);
Console.WriteLine("zrevrangebyscore " + sortedSetKey + " 0 2, result is ");
for (int i = 0; i < names.Length; i++)
{
    Console.WriteLine(names[i] + " ");
}
Console.WriteLine();

```

16.2.5.2. Use redis-cli

You can use the Redis command-line interface (`redis-cli`) tool to connect to a KVStore for Redis instance.

 **Notice** KVStore for Redis only supports connections over an internal network. Therefore, you must install `redis-cli` on an Elastic Compute Service (ECS) instance in the same VPC network as an KVStore for Redis instance, and connect the ECS instance to the KVStore for Redis instance to manage data.

Install redis-cli

Install a Linux-based version of Redis to use `redis-cli`. For more information, visit the [Redis official website](#).

Prerequisites

Connections over an internal network

- If the KVStore for Redis instance and the ECS instance run in a classic network, both instances must be deployed in the same region.
- You have added the internal IP address of the ECS instance to an whitelist of the KVStore for Redis instance.
- The operating system of the local host must be Linux.
- You have installed the Linux-based version of Redis on the ECS instance.
- If you use a custom account to connect to the KVStore for Redis instance, the connection password must be in the format of `<user>:<password>`. For example, if the username of a custom account is `admin` and the password is `password`, the password used to connect to the KVStore for Redis instance must be in the format of `admin:password`.

Connect to a KVStore for Redis instance

On the command line, run the following command to connect to the KVStore for Redis instance.

```
redis-cli -h <host> -p <port> -a <password>
```

Parameters

Parameter	Description
<code>-h</code>	Specifies the endpoint of the KVStore for Redis instance.

Parameter	Description
-p	Specifies the service port of the KVStore for Redis instance. The default port number is 6379 and cannot be changed.
-a	Specifies the password used to connect to the KVStore for Redis instance. To enhance data security, you can skip this parameter to avoid revealing the password in plaintext. After you run the preceding command, you can enter <code>auth <password></code> to complete the authentication. The following figure shows an example.

Example

```
[root@ ~]# redis-cli -h r-bp1.redis.rds.aliyuncs.com -p 6379
r-bp1.redis.rds.aliyuncs.com:6379> auth a
OK
r-bp1.redis.rds.aliyuncs.com:6379>
```

16.3. Instance management

16.3.1. Change the password

If you forget your password, need to change your password, or have not set a password for an instance, you can set a new password for the instance.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the upper-right corner of the **Basic Information** page, click **Modify Password**.
4. In the **Change Password** dialog box that appears, set **Old Password**, **New Password**, **Confirm Password**.

Note

- If you forget your password, you can click **Forgot password?** in the **Change Password** dialog box. In the **Reset Password** dialog box that appears, you can set a new password.
- The password must be 8 to 32 characters in length.
- The password must contain characters from at least three of the following categories: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @ # \$ % ^ & * () _ + - =

16.3.2. Configure a whitelist

Before using a KVStore for Redis instance, add IP addresses or CIDR blocks used to access the database to the instance whitelist to improve database security and stability.

Context

Note A properly configured whitelist can guarantee the highest-level security protection for your KVStore for Redis instance. We recommend that you maintain the whitelist on a regular basis.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions**

column.

3. On the **Instance Information** page, click **Whitelist Settings** in the left-side navigation pane.

4. On the **Whitelist Settings** page, proceed in either of the following ways:

- To customize the whitelist group name, create a new whitelist group:
 - a. Click **Add a Whitelist Group** in the upper-right corner.
 - b. In the **Add a Whitelist Group** dialog box that appears, set **Group Name**.

Note A group name must be 2 to 32 characters in length and contain lowercase letters, digits, or underscores (_). The group name must start with a lowercase letter and end with a letter or digit. You cannot change this name after you create the whitelist group.

- If you do not require a custom whitelist group, click **Modify** next to the target whitelist group.

5. In the **Add a Whitelist Group** or **Modify Whitelist of Group** dialog box that appears, proceed in either of the following ways:

- **Manually modify the Whitelist of Group field:**
 - a. In the **Whitelist of Group** field, enter the IP addresses or CIDR blocks that you can use to connect to the KVStore for Redis instance.

Manually modify the whitelist group

Note

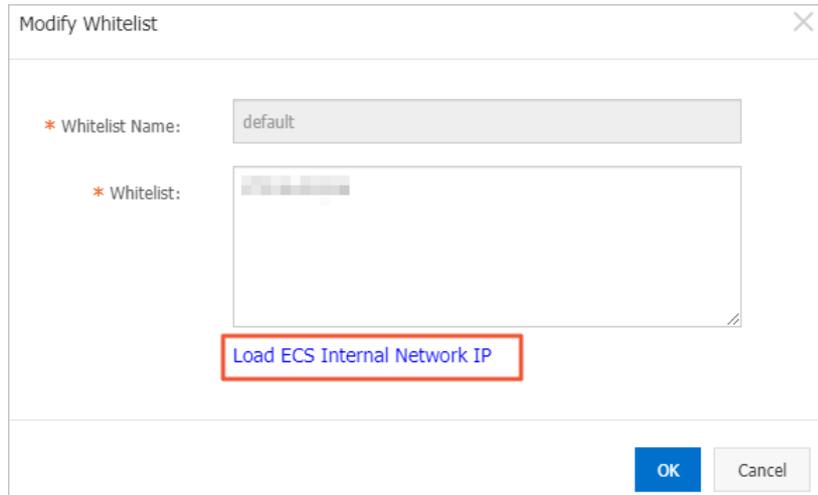
- Set the whitelist to `0.0.0.0/0` to allow connections from all IP addresses.
- Set the whitelist to `127.0.0.1` to block connections from all IP addresses.
- Set the whitelist to a CIDR block to allow connections from the IP addresses within the CIDR block, such as `10.10.10.0/24`.
- When you enter multiple IP addresses or CIDR blocks, separate them with commas (,) and leave no space before or after each comma.
- You can add 1,000 or fewer IP addresses or CIDR blocks to each whitelist group.

b. Click **OK**.

- Load internal IP addresses of target ECS instances under the current Alibaba Cloud account:

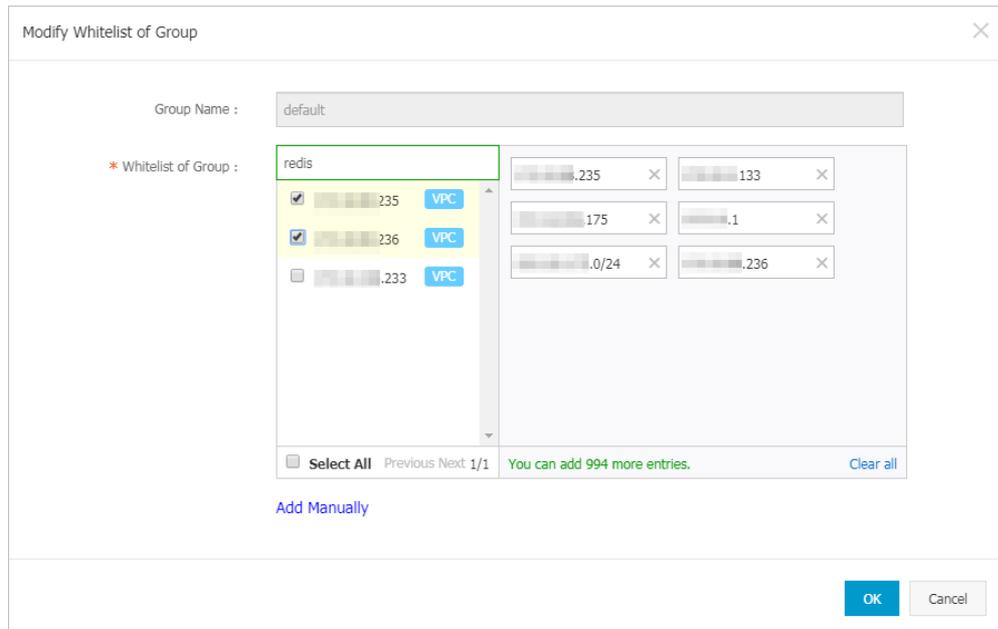
a. Click Load ECS Internal IP Addresses.

Load internal IP addresses of target ECS instances



b. Select internal IP addresses of target ECS instances.

Select internal IP addresses of target ECS instances



Note You can perform a fuzzy search by ECS instance name, ID, or IP address on the search bar above the list of ECS internal IP addresses.

c. Click OK.

16.3.3. Change configurations

This topic describes how to change the configuration of a KVStore for Redis instance.

Context

 **Note** After configuration changes have been completed, the system will migrate data and experience transient disconnection for a few seconds during this process. We recommend that you upgrade or downgrade the instance configuration during off-peak hours.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Then, click the instance ID or click **Change Configurations** in the **Actions** column.
3. On the **Change Configurations** page, change the configurations and click **Submit**. The following example provides common configurations:

Configuration	Description
Architecture Type	The architecture type of the KVStore for Redis instance. KVStore for Redis provides cluster and standard architectures. The cluster architecture meets large-capacity or high-performance requirements. Native Redis databases run in a single-threading model. If your database does not require high performance, we recommend that you use a standard instance. To achieve higher performance, select a cluster instance.
Instance Class	The specification of the instance. The maximum number of connections and maximum internal bandwidth vary, depending on the instance specification.

16.3.4. Set a maintenance window

You can modify the default maintenance window to perform maintenance on KVStore for Redis during off-peak hours.

Context

To ensure the stability of KVStore for Redis instances on the Alibaba Cloud platform, the backend system performs maintenance on instances and servers occasionally.

To guarantee the stability of the maintenance process, instances will enter the **Maintaining Instance** status before the preset maintenance window on the day of maintenance. While an instance is in this state, data in the database can still be accessed and query operations such as performance monitoring are still available. However, change operations such as configuration change are temporarily unavailable for this instance in the console.

 **Note** During the maintenance process, instances may be disconnected in the process of maintenance. We recommend that you set the maintenance window to a period during off-peak hours.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Settings** to the right of the **Maintenance Window** field in the **Basic Information** section.
4. Select time periods and click **Save**.

 **Note** The time periods are in UTC+8.

16.3.5. Upgrade the minor version

Alibaba Cloud has continuously optimized the kernel of KVStore for Redis to fix security vulnerabilities and provide more stable services. You can upgrade the kernel version (minor version) of a KVStore for Redis instance with one click in the console.

Context

Note

- We recommend that you upgrade instance versions during off-peak hours and ensure that your application supports automatic reconnection.
- The system automatically checks the kernel version of an instance. If the current version is the latest, the **Minor Version Upgrade** button in the upper-right corner of the **Basic Information** section for this instance will appear dimmed.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Minor Version Upgrade** in the upper-right corner of the **Basic Information** section.
4. In the **Minor Version Upgrade** dialog box that appears, click **Upgrade Now**. On the **Instance Information** page, the instance status will become **Upgrading a minor version**. When the instance status returns to **Available**, the upgrade has been completed.

16.3.6. Configure SSL encryption

The standard and cluster Instances of Redis 2.8 and the cluster instances of Redis 4.0 support secure sockets layer (SSL) encryption. You can enable SSL encryption to ensure more secure data transmission.

Context

 **Note** SSL encryption may increase the network response time of instances. We recommend that you enable this feature only when necessary.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **SSL Settings**.
4. In the upper-right corner of the **SSL Settings** page, click **Configure SSL**.
5. In the **Configure SSL** dialog box that appears, turn on the **Enable** switch. The switch will turn from green to gray when it is enabled. Click **OK**.
 - If an error message is displayed to indicate that the instance is in an abnormal state, click **OK** in the message that appears.
 - If an error message is displayed to indicate that the feature is not supported in this version, upgrade the minor version of the instance. For more information, see [Upgrade the minor version](#).
 - After the operation, you must wait for a short period of time before the system displays the operation result.
 - You can also click **Update Validity** and **Download CA Certificate** in the upper-right corner of the **SSL**

Settings page to perform relevant operations.

16.3.7. Clear data

You can clear the data of a KVStore for Redis instance in the console.

Context

 **Warning** This operation will delete all data contained on an instance. Deleted data cannot be restored. Proceed with caution.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the upper-right corner of the **Instance Information** page, click **Clear Data**.
4. In the **Clear Data** message that appears, click **OK**.

16.3.8. Release an instance

You can release a KVStore for Redis instance at any time based on your business needs. This topic describes how to release a KVStore for Redis instance.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance List** page, find the target instance. Then, click **Release** in the **Actions** column.

 **Warning** After an instance is released, it cannot be restored. Proceed with caution. We recommend that you back up your data before releasing the instance.

4. In the **Release Instance** message that appears, click **OK**.

16.3.9. Manage database accounts

KVStore for Redis allows you to create up to 20 database accounts for an instance. You can grant permissions to these accounts and manage your instance based on the actual needs to minimize misoperations.

Prerequisites

The engine version of the instance is Redis 4.0 or later.

 **Note** If the engine version of the instance is not Redis 4.0, only the default account is available. The default account is created when you create the instance. For more information about how to change the password of the default account, see [Change the password](#).

Context

You can create accounts, delete accounts, reset the password, and change the permissions. After an account is created, you can use this account to log on to the database and use the command-line tool to perform operations on the database with the account and granted permissions.

Create an account

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instances** page, find the instance that you want to manage and click the instance ID. By default, the **Instance Information** page is displayed. In the left-side navigation pane, click **Account Management**.

 **Note** If **Account Management** is not available for an instance of Redis 4.0 or later, you can try to upgrade the minor version. For more information, see [Upgrade the minor version](#).

4. In the upper-right corner of the **Account Management** page, click **Create**.
5. In the **Create Account** dialog box that appears, set the following parameters and click **OK**.

 **Note**

16.3.10. Use a Lua script

KVStore for Redis instances of all editions support Lua commands.

Support for Lua commands

Lua scripts improve the performance of KVStore for Redis. With support for the Lua environment, KVStore for Redis is able to perform check-and-set (CAS) operations, allowing you to combine and run multiple commands in an efficient manner.

 **Note** If the `Eval` command cannot be executed, such as when the "ERR command eval not support for normal user" message is displayed, you can try to [Upgrade the minor version](#). During the upgrade, the instance may be disconnected and become read-only for a few seconds. We recommend that you upgrade the version of an instance during off-peak hours.

Limits on Lua scripts

To ensure that all operations in a Lua script are performed within the same hash slot, the cluster edition of KVStore for Redis sets the following limits on a Lua script:

- The Lua script uses the `redis.call/redis.pcall` function to call Redis commands. For these commands, all the keys must be passed by using the `KEYS` array, which cannot be replaced by Lua variables. Otherwise, the following error message is returned:

```
-ERR bad lua script for redis cluster, all the keys that the script uses should be passed using the KEYS arrayrn
```

- All the keys that the script uses must be allocated in the same hash slot. Otherwise, the following error message is returned:

```
-ERR eval/evalsha command keys must be in same slotrn
```

- Keys must be passed for all the commands to be called. Otherwise, the following error message is returned:

```
-ERR for redis cluster, eval/evalsha number of keys can't be negative or zerorn
```

 **Note** If you want to break the Lua limits of Redis Cluster and can ensure that all operations are performed in the same hash slot in the code, you can set the `script_check_enable` parameter to 0 in the console to disable the backend script check.

16.3.11. Restart an instance

You can restart an instance from the Instance List page of the console.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the Instance List page, find the target instance. Then, click **Restart** in the Actions column.

 **Notice** During the restart, the instance may be disconnected for a few seconds. We recommend that you restart instances during off-peak hours and ensure that your application supports automatic reconnection.

3. In the dialog box that appears, select a restart time and click **OK**.
 - **Restart Immediately:** restarts the instance immediately.
 - **Restart Within Maintenance Window:** restarts the instance within the preset [maintenance window](#).

16.3.12. Export the list of instances

You can export the list of KVStore for Redis instances from the KVStore for Redis console for offline management.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. In the upper-right corner of the Instance List page, click the **Export Instances** icon.
3. In the **Export Instance List** dialog box that appears, select the columns to export and click **OK**.

 **Note** After you click **OK**, the browser begins to download the CSV file. You can use Excel or a text editor to view this file.

16.4. Connection management

16.4.1. View connection strings

You can view the internal and public endpoints of instances in the KVStore for Redis console.

Context

-  **Note**
- The virtual IP address of a KVStore for Redis instance may change when you maintain or modify the service. To ensure connection availability, we recommend that you use a connection string to access the KVStore for Redis instance.
 - For more information about how to apply for a public connection string, see [Applies for a public connection string](#).

Procedure

1. [Log on to the KVStore for Redis console.](#)

2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, view **Internal Connection Address (Host)** and **Public Endpoint (Host)** in the **Connection Information** section.

16.4.2. Apply for a public endpoint

This topic describes how to apply for a public endpoint.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Apply for External IP Address** in the **Connection Information** section.
4. In the **Apply for External IP Address** dialog box that appears, enter an endpoint and port number, and click **OK**.

Note

- The custom endpoint prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
- The custom port ranges from 1024 to 65535. The default value is 6379.
- After you apply for a public endpoint, you must add the public IP address to an IP address whitelist of the instance to connect to the instance over the Internet. For more information, see [Configure a whitelist](#).

5. On the **Instance Information** page, view the **Public Endpoint** in the **Connection Information** section.

 **Note** If a public endpoint is no longer needed, you can click **Release Public Endpoint** next to the **Public Endpoint** to release the endpoint.

16.4.3. Change the connection string of an instance

KVStore for Redis allows you to modify internal and public endpoints for instances. When changing the KVStore for Redis instance, you can change the endpoint of the new instance to the endpoint of the original instance without modifying the application.

Prerequisites

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

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the **Instance Information** page, click **Modify Public Endpoint** in the **Connection Information** section.
4. In the **Modify Public Endpoint** dialog box that appears, set **Connection Type**, **Endpoint**, and **Port**. Click **OK**.

Note

- The custom endpoint prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
- The custom port range is 1024 to 65535. The default value is 6379.
- If **Connection Type** is set to **Internal Address**, you cannot set **Port**.

16.5. Parameter configuration

KVStore for Redis allows you to customize certain instance parameters. This topic describes parameters and the common methods to modify them in the KVStore for Redis console.

Context

KVStore for Redis is completely compatible with the native database services of Redis. The method to set parameters for KVStore for Redis is similar to that of an on-premises Redis database. You can set the parameters described in this topic in the KVStore for Redis console.

Parameters

Parameters

Parameter	Description
#no_loose_check-whitelist-always	Specifies whether to check whether the client IP address is in the whitelist of the KVStore for Redis instance after password-free access is enabled in Virtual Private Cloud (VPC). Default value: no. If you set this parameter to yes, the whitelist will still take effect in password-free access mode for VPC. Valid values: <ul style="list-style-type: none"> • yes • no
#no_loose_disabled-commands	Specifies the disabled commands. Separate multiple commands with commas (,). You can disable the following commands: FLUSHALL, FLUSHDB, KEYS, HGETALL, EVAL, EVALSHA, and SCRIPT.
#no_loose_ssl-enabled	Specifies whether to enable SSL encryption. Default value: no. Valid values: <ul style="list-style-type: none"> • yes • no
#no_loose_sentinel-enabled	Specifies whether to enable Sentinel-compatible mode. Default value: no. Valid values: <ul style="list-style-type: none"> • yes • no
client-output-buffer-limit pubsub	Limits the size of output buffers for Pub/Sub clients. This parameter can contain options in the following format: <code><hard limit> <soft limit> <soft seconds></code> . <ul style="list-style-type: none"> • Hard limit: If the output buffer of a Pub/Sub client reaches or exceeds the number of bytes specified by hard limit, the client is immediately disconnected. • soft limit and soft seconds: If the output buffer of a Pub/Sub client reaches or exceeds the size in bytes specified by soft limit for a period of time in seconds specified by soft seconds, the client will be disconnected.

Parameter	Description
dynamic-hz	<p>Specifies whether to enable dynamic frequency control for background tasks. Default value: yes. Valid values:</p> <ul style="list-style-type: none"> • yes • no
hash-max-ziplist-entries	<p>Specifies the maximum size of each key-value pair stored within a hash in bytes. A hash is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the hash in bytes must be less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored within the hash must be less than the value of the hash-max-ziplist-entries parameter.
hash-max-ziplist-value	<p>Specifies the maximum size of each key-value pair stored within a hash in bytes. A hash is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the hash in bytes must be less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored within the hash must be less than the value of the hash-max-ziplist-entries parameter.
hz	<p>Specifies the execution frequency for background tasks, such as tasks to evict expired keys. Valid values: 1 to 500. Default value: 10. The larger the value of the hz parameter, the more frequently background tasks are performed and the more precisely timeout events are handled, but the more CPU KVStore for Redis consumes. We recommend that you do not set the hz parameter to a value greater than 100.</p>
lazyfree-lazy-eviction	<p>Specifies whether to enable lazyfree for the eviction feature. Default value: no. Valid values:</p> <ul style="list-style-type: none"> • yes • no
lazyfree-lazy-expire	<p>Specifies whether to enable lazyfree to delete expired keys. Default value: yes. Valid values:</p> <ul style="list-style-type: none"> • yes • no
lazyfree-lazy-server-del	<p>Specifies whether to enable lazyfree to asynchronously delete data with the DEL command. Default value: yes. Valid values:</p> <ul style="list-style-type: none"> • yes • no

Parameter	Description
list-compress-depth	<p>Specifies the number of nodes that are not compressed at each side in a list. Default value: 0. Valid values:</p> <ul style="list-style-type: none"> • 0: does not compress any nodes in the list. • 1: does not compress the first node from each side of the list, but compresses all nodes in between. • 2: does not compress the first two nodes from each side of the list, but compresses all nodes in between. • 3: does not compress the first three nodes from each side of the list, but compresses all nodes in between. • And so on up to 65535.
list-max-ziplist-size	<ul style="list-style-type: none"> • Specifies the maximum size of each ziplist in a quicklist. A positive number indicates the maximum number of elements in each ziplist of a quicklist. For example, if you set this parameter to 5, each ziplist of a quicklist can contain a maximum of five elements. • A negative number indicates the maximum number of bytes in each ziplist of a quicklist. Default value: -2. Valid values: <ul style="list-style-type: none"> ◦ -5: indicates that each ziplist of a quicklist cannot exceed 64 KB (1 KB = 1,024 bytes). ◦ -4: indicates that each ziplist of a quicklist cannot exceed 32 KB. ◦ -3: indicates that each ziplist of a quicklist cannot exceed 16 KB. ◦ -2: indicates that each ziplist of a quicklist cannot exceed 8 KB. ◦ -1: indicates that each ziplist of a quicklist cannot exceed 4 KB.
maxmemory-policy	<p>Specifies the policy used to evict keys if the memory is fully occupied. Valid values: LRU means least recently used. LFU means least frequently used. LRU, LFU, and TTL are implemented by using approximated randomized algorithms.</p> <ul style="list-style-type: none"> • volatile-lru: evicts the approximated least recently used (LRU) keys among keys with a preset expiration time. • allkeys-lru: evicts the approximated LRU keys. • volatile-lfu: evicts the approximated least frequently used (LFU) keys among keys with a preset expiration time. • allkeys-lfu: evicts the approximated LFU keys. • volatile-random: evicts random keys among keys with a preset expiration time. • allkeys-random: evicts random keys. • volatile-ttl: evicts keys with the nearest time to live (TTL) among keys with a preset expiration time. • noeviction: does not evict any keys, but returns an error on write operations.

Parameter	Description
notify-keyspace-events	<p>Specifies the events that the Redis server can notify clients of. The value of this parameter is any combination of the following characters, each of which specifies a type of event to be notified:</p> <ul style="list-style-type: none"> • • K: keyspace events, published with the <code>__keyspace@<db>__</code> prefix. • E: keyevent events, published with the <code>__keyevent@<db>__</code> prefix. • g: generic commands that are non-type specific, such as DEL, EXPIRE, and RENAME. • l: list commands. • s: set commands. • h: hash commands. • z: sorted set commands. • x: expired key events. An expired key event is generated when a key expires. • e: evicted key events. An evicted key event is generated when a key is evicted due to the policy specified by the <code>maxmemory-policy</code> parameter. • A: the alias for <code>g\$lshzxe</code>.
set-max-intset-entries	<p>Specifies the maximum number of data entries in a set. A set is encoded by using intset when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The set is composed of just strings. The number of strings is less than the value of this parameter. 2. All strings are integers in radix 10 in the range of 64-bit signed integers.
slowlog-log-slower-than	<p>Specifies whether to log slow queries.</p> <ul style="list-style-type: none"> • Negative number: does not log slow queries. • 0: logs all queries. • Positive number: logs queries that exceed an execution time specified by this positive number, in microseconds. <p>Valid values: 0 to 10,000,000. Default value: 10,000.</p>
slowlog-max-len	<p>Specifies the maximum number of slow query log entries that can be stored.</p> <p>Valid values: 100 to 10,000. Default value: 1,024.</p>
stream-node-max-bytes	<p>Specifies the maximum memory that can be used by each macro node in streams. Valid values: 0 to 999,999,999,999,999. If you set the parameter to 0, each macro node can use an unlimited amount of memory.</p>
stream-node-max-entries	<p>Specifies the maximum number of stream entries that can be stored within each macro node. Valid values: 0 to 999,999,999,999,999. If you set the parameter to 0, each macro node can store unlimited stream entries.</p>
timeout	<p>Specifies a timeout period for client connections. Unit: seconds. Valid values: 0 to 100,000. 0 indicates that client connections never time out.</p>

Parameter	Description
zset-max-ziplist-entries	<p>Specifies the maximum size of each key-value pair stored within a sorted set in bytes. A sorted set is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the sorted set in bytes must be less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored within the sorted set must be less than the value of the zset-max-ziplist-entries parameter.
zset-max-ziplist-value	<p>Specifies the maximum size of each key-value pair stored within a sorted set in bytes. A sorted set is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the sorted set in bytes must be less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored within the sorted set must be less than the value of the zset-max-ziplist-entries parameter.
list-max-ziplist-entries	<p>Specifies the maximum size of each key-value pair stored within a list in bytes. A list is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the list in bytes must be less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored within the list is less than the value of the list-max-ziplist-entries parameter.
list-max-ziplist-value	<p>Specifies the maximum size of each key-value pair stored within a list in bytes. A list is encoded using ziplist when it meets the following conditions:</p> <ol style="list-style-type: none"> 1. The maximum size of each key-value pair stored within the list in bytes must be less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored within the list is less than the value of the list-max-ziplist-entries parameter.
cluster_compat_enable	<p>Specifies whether to enable compatibility with the syntax of Redis Cluster. Default value: 1. Valid values:</p> <ul style="list-style-type: none"> • 0: no • 1: yes
script_check_enable	<p>Specifies whether to confirm that all the keys used in a Lua script are in the same hash slot. Default value: 1. Valid values:</p> <ul style="list-style-type: none"> • 0: no • 1: yes

 **Note** The maxclients parameter, which is used to specify the maximum number of connections to Redis data nodes, is fixed to 10,000. You cannot modify the value of this parameter.

Configure parameters in the KVStore for Redis console

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.

3. In the left-side navigation pane of the Instance Information page, click **System Parameters**.
4. Find the target parameter and click **Modify** in the Action column.
5. In the dialog box that appears, modify the parameter value and click **OK**.

16.6. Backup and recovery

16.6.1. Back up data automatically

An increasing number of applications use KVStore for Redis for persistent storage. Because of this, KVStore for Redis supports routine backup mechanisms to restore data after misoperations occur. Alibaba Cloud provides secondary nodes to back up .rdb files as snapshots. Backup operations do not affect the performance of your instance. You can customize the backup operation in the console.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the Instance List page, find the target instance. Click the instance ID or click **Manage** in the Actions column.
3. In the left-side navigation pane of the Instance Information page, click **Backup and Recovery**.
4. On the Backup and Recovery page, click the Backup Settings tab.
5. Click **Edit** to customize the automatic backup cycle and backup time.

 **Note** Backup data is retained for seven days. You cannot modify this configuration.

6. Click **OK**.

16.6.2. Back up data manually

You can initiate a manual backup task in the console at any time.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the Instance List page, find the target instance. Click the instance ID or click **Manage** in the Actions column.
3. In the left-side navigation pane of the Instance Information page, click **Backup and Recovery**.
4. In the upper-right corner of the Data Backup tab, click **Create Backup**.
5. In the message that appears, click **Confirm**.

 **Note** On the Data Backup tab, you can select a time range to query historical backup data. Backup data is retained for seven days, so you can query historical backup data in the past seven days.

16.6.3. Download backup files

To archive these backup files for a longer period, you can copy their URLs in the console and download the database backup files to a local directory.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the Instance List page, find the target instance. Click the instance ID or click **Manage** in the Actions column.
3. In the left-side navigation pane of the Instance Information page, click **Backup and Recovery**.

4. On the **Data Backup** tab, select the backup set to be archived and click **Download**.
5. In the **Download Backup File** dialog box that appears, click one of the following buttons to continue with the procedure:
 - **Download**: downloads the backup file to a local directory.
 - **Get URL for Internet**: copies the public URL for downloading the backup file, and downloads the backup file over the Internet.
 - **Get URL for Intranet**: copies the internal URL for downloading the backup file, and downloads the backup file over the internal network.
 - **Cancel**: cancels downloading the backup file.

16.6.4. Restore data

You can use backup files to restore data in the console.

Context



Notice

- Data restoration is highly risky. Check the data to be restored before performing this operation. Proceed with caution.
- This feature is not applicable to non-cluster KVStore for Redis instances.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **Backup and Recovery**.
4. On the **Data Backup** tab, select the target backup file and click **Restore Data**.
5. In the **Restore Data** dialog box that appears, click **Continue**. You can apply backup files to a new instance by [cloning an instance](#).

16.6.5. Clone an instance

You can apply backup files to a new instance by cloning an instance.

Context



Note This feature is applicable to non-cluster KVStore for Redis instances.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **Backup and Recovery**.
4. On the **Data Backup** tab, select the target backup set and click **Clone Instance**.
5. In the **Clone Instance** message that appears, click **OK**.

16.7. Performance monitoring

16.7.1. View monitoring data

You can query the monitoring data of a KVStore for Redis instance for a specified period within the last month.

Procedure

1. [Log on to the KVStore for Redis console](#).
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **Performance Monitor**.
4. On the **Historical Monitoring Data** page, click the calendar icon next to **Query Time**.
5. Set a time range in which you want to query monitoring data and click **OK**.

 **Note** For more information about the monitoring metrics, see [Understand metrics](#).

16.7.2. Customize metrics

You can select the metrics to be displayed on the **Historical Monitoring Data** page of the KVStore for Redis console as needed.

Context

KVStore for Redis supports more than 10 groups of monitoring metrics. By default, the **Performance Monitor** page displays the monitoring metrics of the basic monitoring group. You can click the **Custom Metrics** button to switch to the metrics of other monitoring groups. The following table describes the monitoring groups.

Monitoring group	Description
Basic monitoring group	The basic instance monitoring information, such as the QPS, bandwidth, and memory usage.
Key monitoring group	The monitoring information on the use of key-value related commands, such as the number of times DEL and EXITS are called.
String monitoring group	The monitoring information on the use of string-related commands, such as the number of times APPEND and MGET are called.
Hash monitoring group	The monitoring information on the use of hash-related commands, such as the number of times HGET and HDEL are called.
List monitoring group	The monitoring information on the use of list-related commands, such as the number of times BLPOP and BRPOP are called.
Set monitoring group	The monitoring information on the use of set-related commands, such as the number of times SADD and SCARD are called.
Zset monitoring group	The monitoring information on the use of zset-related commands, such as the number of times ZADD and ZCARD are called.
HyperLog monitoring group	The monitoring information on the use of HyperLogLog-related commands, such as the number of times PFADD and PFCOUNT are called.
Pub/Sub monitoring group	The monitoring information on the use of publication and subscription-related commands, such as the number of times PUBLISH and SUBSCRIBE are called.
Transaction monitoring group	The monitoring information on the use of transaction-related commands, such as the number of times WATCH, MULTI, and EXEC are called.
Lua script monitoring group	The monitoring information on the use of Lua script-related commands, such as the number of times EVAL and SCRIPT are called.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **Performance Monitor**.
4. On the **Historical Monitoring Data** page, click **Customize Metrics** in the **Data Index** section.
5. In the **Customize Metrics** dialog box that appears, select the new monitoring group and click **OK**.

16.7.3. Modify monitoring frequency

KVStore for Redis console allows you to set the frequency at which monitoring data is collected.

Context

You can set the monitoring frequency to either 5 or 60 seconds to specify how often monitoring data to be collected by KVStore for Redis. The default monitoring time of 60 seconds is sufficient to meet common monitoring requirements. If you need to observe certain metrics at a higher frequency and lower latency, you can change the monitoring frequency to 5 seconds as described in the following section. Monitoring data does not occupy instance storage space, and collection of monitoring data does not affect normal running of the instance.

Procedure

1. [Log on to the KVStore for Redis console.](#)
2. On the **Instance List** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. In the left-side navigation pane of the **Instance Information** page, click **Performance Monitor**.
4. In the upper-right corner of the **Historical Monitoring Data** page, click **Monitoring Frequency**.
5. In the **Monitoring Frequency** dialog box that appears, select the new monitoring frequency and click **OK**.

16.7.4. Monitoring metrics

KVStore for Redis monitors more than 10 groups of metrics in real time. This allows you to monitor the running status of KVStore for Redis instances. This topic describes these metrics.

Basic monitoring metrics

Metric	Unit	Description	Statistical method
CpuUsage	%	The CPU usage.	Monitor the CPU usage when collecting monitoring data.
UsedMemory	Bytes	The amount of the used memory.	Monitor the amount of the used memory when collecting monitoring data.
TotalQps	Counts/s	The number of requests received by an instance per second.	Divide the number of requests in a monitoring cycle by the number of seconds in the monitoring cycle.
ConnCount	Counts	The number of connections.	Monitor the number of connections when collecting monitoring data.

Metric	Unit	Description	Statistical method
InFlow	KBps	The amount of data received by an instance per second.	Divide the amount of data received in a monitoring cycle by the number of seconds in the monitoring cycle.
OutFlow	KBps	The amount of data sent by an instance per second.	Divide the amount of data sent in a monitoring cycle by the number of seconds in the monitoring cycle.
FailedCount	Counts/s	The average number of abnormal requests per second.	Divide the total number of abnormal requests in a monitoring cycle by the number of seconds in the monitoring cycle.
AvgRt	us	The average response time of all requests.  Note For more information, see Response time (RT) metrics .	Divide the processing time of all requests in a monitoring cycle by the number of requests in the monitoring cycle.
MaxRt	us	The maximum response time of requests.  Note For more information, see Response time (RT) metrics .	Monitor the maximum amount of time consumed for processing a single request in a monitoring cycle.
Keys	Counts	The total number of keys.	Monitor the number of keys when collecting monitoring data.
Expires	Counts	The total number of keys for which an expiration time is set.	Monitor the total number of keys for which an expiration time is set when collecting monitoring data.
ExpiredKeys	Counts	The total number of expired keys.	Monitor the cumulative sum when collecting monitoring data. After the instance is restarted, the cumulative sum is calculated again.
EvictedKeys	Counts	The total number of keys that are evicted because the memory is fully occupied.	Monitor the cumulative sum when collecting monitoring data. After the instance is restarted, the cumulative sum is calculated again.

Metric	Unit	Description	Statistical method
request	Bytes	The total amount of request data received by KVStore for Redis nodes in a monitoring cycle.	See the description of this metric.
response	Bytes	The total amount of response data sent by KVStore for Redis nodes in a monitoring cycle.	See the description of this metric.
request_max	Bytes	The maximum amount of data that a single request has in a monitoring cycle.	See the description of this metric.
response_max	Bytes	The maximum amount of data that a single response has in a monitoring cycle.	See the description of this metric.
traffic_control_input	Counts	The number of times that downstream throttling is triggered.	Monitor the cumulative sum in a monitoring cycle.
traffic_control_output	Counts	The number of times that upstream throttling is triggered.	Monitor the cumulative sum in a monitoring cycle.
traffic_control_input_status	Counts	Indicates whether downstream throttling has been triggered in a monitoring cycle. A value of 0 indicates that throttling has not been triggered. A value of 1 indicates that throttling has been triggered.	See the description of this metric.
traffic_control_output_status	Counts	Indicates whether upstream throttling has been triggered in a monitoring cycle. A value of 0 indicates that throttling has not been triggered. A value of 1 indicates that throttling has been triggered.	See the description of this metric.
hit_rate	%	The request hit ratio. This metric indicates the probability that data exists in a KVStore for Redis instance when the data is requested.	Calculate the percentage of the hit requests to the total number of requests in a monitoring cycle.
hit	Counts	The number of hit requests.	Monitor the number of hit requests in a monitoring cycle.
miss	Counts	The number of missed requests.	Monitor the number of missed requests in a monitoring cycle.

Metric	Unit	Description	Statistical method
evicted_keys_per_sec	Counts/s	The number of keys that are evicted per second.	Divide the total number of keys evicted in a monitoring cycle by the number of seconds in the monitoring cycle.

Other monitoring metrics

The system also uses other monitoring metrics to monitor specific types of data or specific features. These monitoring metrics are classified into:

- Metrics that reflect the number of times that commands are used. For example, the del, dump, and exists metrics for keys monitoring indicate the number of times the DEL, DUMP, and EXISTS commands are used.
- **Response time (RT) metrics** of commands. For example, the metrics that end with avg_rt such as del_avg_rt, dump_avg_rt, and exists_avg_rt in the key monitoring group are used to monitor the average response time of the DEL, DUMP, and EXISTS commands in a monitoring cycle.

Response time (RT) metrics

All groups of monitoring metrics include response time metrics. Such metrics end with Rt or rt. For example, the AvgRt and MaxRt metrics in the basic monitoring metrics or the del_avg_rt and exists_avg_rt metrics in the keys monitoring.

The AvgRt and MaxRt metrics in the basic monitoring group are the most frequently used response time metrics. These metrics have different meanings for proxy nodes and data nodes.

- For a cluster instance or a read/write splitting instance, the AvgRt metric of a proxy node reflects the average time consumed by the proxy node to process all commands. A proxy node processes a command by following these steps:
 - i. The proxy node receives a command and forwards the command to a data node.
 - ii. The data node processes the command and responds to the proxy node.
 - iii. The proxy node returns the command processing result.

The AvgRt metric of the proxy node includes the amount of time consumed by the data node to process a command and the waiting time. It also includes the amount of time consumed for network communication between the proxy node and the data node.

- For data nodes of a cluster instance or a read/write splitting instance or for a standard instance, the AvgRt metric reflects the average time consumed by a data node to process all commands. This metric records the period from the time when the data node receives the command to the time when the data node returns the result. This metric does not include the time consumed by the proxy node to process a command and the time consumed for network communication.
- The MaxRt metric specifies the maximum response time of requests. The statistical method of this metric is similar to that of the AvgRt metric for all KVStore for Redis instances.

17. ApsaraDB for MongoDB

17.1. Before you start

You must get familiar with the precautions and limits of ApsaraDB for MongoDB before you start.

To ensure the stability and security of ApsaraDB for MongoDB instances, pay attention to the limits described in [ApsaraDB for MongoDB limits](#).

ApsaraDB for MongoDB limits

Operation	Limit
Create a database replica	<ul style="list-style-type: none"> The system automatically creates a three-node replica set. The replica set consists of a primary node, a secondary node, and a hidden secondary node (invisible to you). You cannot manually create a secondary node.
Restart a database	You must restart an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console or by using OpenAPI Explorer.

17.2. Log on to the ApsaraDB for MongoDB console

This topic describes how to log on to the ApsaraDB for MongoDB console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

- In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
- Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

- Click **Login** to go to the ASCM console homepage.
- In the top navigation bar, choose **Products > Database Services > ApsaraDB for MongoDB**.

17.3. Quick start

17.3.1. Use ApsaraDB for MongoDB

This topic is a quick start guide to basic usage operations for ApsaraDB for MongoDB, such as creating an instance, configuring a whitelist, and connecting to an instance. Flowcharts are used to describe the basic procedures in ApsaraDB for MongoDB, and guide you to create an ApsaraDB for MongoDB instance.



- **Create an ApsaraDB for MongoDB instance**

An instance is a virtual database server on which you can create and manage multiple databases.

- **Configure a whitelist for an ApsaraDB for MongoDB instance**

After you create an ApsaraDB for MongoDB instance, you need to configure a whitelist for the instance to allow external devices to access the instance.

A whitelist can enhance access security for ApsaraDB for MongoDB instances. We recommend that you update the whitelist on a regular basis. The normal services of the instance are not affected if you configure a whitelist.

- **Connect to a replica set instance by using the mongo shell**

After you create an instance and configure a whitelist, you can use the mongo shell to connect to the instance.

17.3.2. Create an ApsaraDB for MongoDB instance

This topic describes how to create an instance in the ApsaraDB for MongoDB console.

Prerequisites

An account is obtained to log on to the ApsaraDB for MongoDB console.

Procedure

1. **Log on to the ApsaraDB for MongoDB console.**
2. Click **Create Instance** to go to the **Create MongoDB Instance** page and configure the parameters.

The following table describes the required parameters.

Parameters for creating an instance

Section	Parameter	Description
Basic Settings	Organization	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.

Section	Parameter	Description
Region	Region	Select a region for the new instance.
	Zone	Select a zone for the new instance.
Specifications	Database Engine	Select a database engine for the new instance. In this case, you can select only MongoDB.
	Engine Version	Select a database version for the new instance. Valid values: <ul style="list-style-type: none"> ○ 3.0 ○ 3.4 ○ 4.0
	Node Type	ApsaraDB for MongoDB supports the following options: <ul style="list-style-type: none"> ○ Three-node Replica Set: uses dedicated memory and I/O resources but shares CPU and storage resources with other general-purpose instances on the same server. ○ Dedicated Instance: uses dedicated CPU, memory, storage, and I/O resources to ensure long-term performance. In this case, an instance is not affected by other instances on the same server. ○ Dedicated Host: exclusively uses all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	Select a node specification for the new instance. Available specifications are displayed on the Create MongoDB Instance page.
	Storage Capacity	Specify the storage capacity for the new instance to store data, system files, log files, and transaction files.
Network	Network Type	ApsaraDB for MongoDB supports the following options: <ul style="list-style-type: none"> ○ Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ○ VPC: A VPC helps you to build an isolated network in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note If you select VPC, you must configure the VPC and VSwitch parameters. </div>
Password	Instance Name	Set the name for the new instance. The name must be 2 to 256 characters in length and can contain digits, letters, underscores (_), and hyphens (-). It must start with a letter.
	Set Password	Determine when to set the password for logging on to databases in the new instance. You can select Now to set the logon password immediately, or select Later to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance .
	Password	Set a password, which must meet the following requirements: <ul style="list-style-type: none"> ○ It must be 8 to 30 characters in length. ○ It must contain uppercase letters, lowercase letters, and digits. Special characters are not allowed.

Section	Parameter	Description
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Password.

3. Click **Submit** to create the instance.

17.3.3. Configure a whitelist for an ApsaraDB for MongoDB instance

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or Classless Inter-Domain Routing (CIDR) blocks that you use for database access to a whitelist of this instance. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

The system creates a default whitelist for each instance. This whitelist can be modified or cleared, but it cannot be deleted. After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 0.0.0.0/0 to the default whitelist. The IP address 0.0.0.0/0 indicates that all IP addresses are allowed to access this instance. For database security, we recommend that you remove the IP address 0.0.0.0/0 and add only IP addresses or CIDR blocks that you allow to whitelists.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **Data Security > Whitelist Setting**.
5. You can manually configure a whitelist or import ECS Internal IP addresses to the whitelist. **Manually modify a whitelist**
 - i. Find the whitelist you want to modify and choose  > **Manually Modify** in the **Operation** column.
 - ii. Enter IP addresses or CIDR blocks.

Note

- Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix. The IP address prefix can consist of 1 to 32 bits.
- 0.0.0.0/0 or a blank field indicates that there is no IP access restrictions. In this case, the database may be at high security risk. We recommend that you set the access permission only for the IP address or CIDR block of your Web server.

Load IP addresses of ECS instances

- i. Find the target whitelist and choose  > **Import ECS Intranet IP** in the **Operation** column.
- ii. From the displayed internal IP addresses of ECS instances under the current account, find the target IP addresses and click  to add them to the whitelist.

iii. Click OK.

17.3.4. Overview of replica set instance connections

This topic describes how to obtain connection strings and connection string URIs that are supported by ApsaraDB for MongoDB, as well as how to use them to connect to replica set instances. You can use a connection string to connect to either the primary or secondary node, and use a connection string URI to connect to both of them. For high availability, we recommend that you connect your application to both primary and secondary nodes by using connection string URIs.

View connection information

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Database Connection** to view connection information.



Intranet Connection - Classic Network	
Role	Address
Primary	ods-  .mongodb.rds.intra.env17e.shuguang.com:3717
Secondary	ods-  .mongodb.rds.intra.env17e.shuguang.com:3717
ConnectionStringURI	mongodb://root:****@dds-  .mongodb.rds.intra.env17e.shuguang.com:3717

Description of connection information

Item	Description
Address	<ul style="list-style-type: none"> • Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. • Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than the classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Role	<ul style="list-style-type: none"> • Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. • Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance.
Primary/Secondary	<p>The connection string of a primary or secondary node is in the following format:</p> <pre><host>:<port></pre> <ul style="list-style-type: none"> • <host>: the endpoint you use to connect to the replica set instance. • <port>: the port you use to connect to the replica set instance.

Item	Description
ConnectionStringURI	<p>A connection string URI is in the following format:</p> <pre data-bbox="405 315 1386 412">mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][? options]]</pre> <ul style="list-style-type: none"> • <code>mongodb://</code>: the prefix of the connection string URI. It indicates a connection string URI. • <code>username:password@</code>: the username and password you use to log on to a database of the replica set instance. You must separate them with a colon (:). • <code>hostX:portX</code>: the endpoint and port of a node in the replica set instance. • <code>/database</code>: the name of the authentication database. It is the database where the database user is created. • <code>? options</code>: additional connection options. <p> Note If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, when a node fails, the read and write operations of your application are not affected as a result of the failover.</p>

Related information

- [Connect to a replica set instance by using the mongo shell](#)

17.3.5. Use the mongo shell to connect to a replica set instance

This topic describes how to use the mongo shell to connect to a replica set instance. The mongo shell is a database management tool provided by ApsaraDB for MongoDB. You can install it on your client or in an ECS instance.

Prerequisites

- The version of the mongo shell is the same as your replica set instance. This ensures successful authentication. For information about the installation procedure, visit [MongoDB official documentation](#). Choose the version in the upper-left corner of the page based on your client version.
- The IP address of your client is added to a whitelist of the replica set instance. For more information, see [Configure a whitelist for an ApsaraDB for MongoDB instance](#).

Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Database Connection** to obtain the connection string of a node and the connection string URI.

 **Note** For more information about the connection strings, see [Overview of replica set instance connections](#).

5. Connect to the replica set instance from your client or ECS instance where the mongo shell is installed.
 - Single-node connection.

During regular tests, you can directly connect to a primary or secondary node. Take note that after the primary node fails, the system automatically switches to the secondary node, and the roles of connected nodes change. This affects the read and write operations of your application.

Command format:

```
mongo --host <host> -u <username> -p --authenticationDatabase <database>
```

- **<host>**: the connection string of the primary or secondary node.
 - **Primary**: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance.
 - **Secondary**: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance.
- **<username>**: the username you use to log on to a database of the replica set instance. The initial username is root.
- **<database>**: the name of the authentication database. It is the database where the database user is created. If the username is root, enter admin.

Example:

```
mongo --host dds-bp*****.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

When `Enter password:` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see [Reset the password for an ApsaraDB for MongoDB instance](#).

 **Note** The password you enter is not displayed.

- **HA connection (recommended)**: You can use a connection string URI to connect to both the primary and secondary nodes of a replica set instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

Command format:

```
mongo "<ConnectionStringURI>"
```

- The connection string URI must be enclosed in a pair of double quotation marks ("").
- **<ConnectionStringURI>**: the connection string URI of the replica set instance.

You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see [Reset the password for an ApsaraDB for MongoDB instance](#).

17.4. Instances

17.4.1. Create an ApsaraDB for MongoDB instance

This topic describes how to create an instance in the ApsaraDB for MongoDB console.

Prerequisites

An account is obtained to log on to the ApsaraDB for MongoDB console.

Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).
2. Click **Create Instance** to go to the **Create MongoDB Instance** page and configure the parameters.

The following table describes the required parameters.

Parameters for creating an instance

Section	Parameter	Description
Basic Settings	Organization	Select an organization for the new instance.
	Resource Set	Select a resource set for the new instance.
Region	Region	Select a region for the new instance.
	Zone	Select a zone for the new instance
Specifications	Database Engine	Select a database engine for the new instance. In this case, you can select only MongoDB.
	Engine Version	Select a database engine version for the new instance. Valid values : <ul style="list-style-type: none"> ◦ 3.0 ◦ 3.4 ◦ 4.0
	Node Type	ApsaraDB for MongoDB supports the following options: <ul style="list-style-type: none"> ◦ Three-node Replica Set: uses dedicated memory and I/O resources but shares CPU and storage resources with other general-purpose instances on the same server. ◦ Dedicated Instance: uses dedicated CPU, memory, storage, and I/O resources to ensure long-term performance. In this case, an instance is not affected by other instances on the same server. ◦ Dedicated Host: exclusively uses all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	Select a node specification for the new instance. Available specifications are displayed on the Create MongoDB Instance page.
	Storage Capacity	Specify the storage capacity for the new instance to store data, system files, log files, and transaction files.
Network	Network Type	ApsaraDB for MongoDB supports the following options: <ul style="list-style-type: none"> ◦ Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. ◦ VPC: A VPC helps you build an isolated network in Apsara Stack. You can customize route tables, CIDR blocks, and gateways in a VPC. We recommend that you select VPC for enhanced security. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note If you select VPC, you must configure the VPC and VSwitch parameters </div>
	Instance Name	Set the name for the new instance. The name must be 2 to 256 characters in length, and can contain digits, letters, underscores (_), and hyphens (-). It must start with a letter.
	Set Password	Determine when to set the password for logging on to databases in the new instance. You can select Now to set the logon password immediately, or select Later to set the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance .

Password Section	Parameter	Description
	Password	Set a password. The password must meet the following requirements: <ul style="list-style-type: none"> It must be 8 to 32 characters in length. It must contain uppercase letters, lowercase letters, and digits. Special characters are not allowed.
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Password.

- Click **Submit** to create the instance.

17.4.2. View the details of an ApsaraDB for MongoDB instance

This topic describes how to view the details of an ApsaraDB for MongoDB instance, such as the basic information, internal network connection information, status, and configurations.

Procedure

- Log on to the [ApsaraDB for MongoDB console](#).
- Go to the instance details page. Either of the following methods can be used:
 - Find the target instance and click its ID to go to the **Basic Information** page, where you can view the details of the instance.
 - In the **Operations** column corresponding to the target instance, choose  > **Manage** to go to the **Basic Information** page, where you can view the details of the instance.

17.4.3. Restart an ApsaraDB for MongoDB instance

This topic describes how to restart an ApsaraDB for MongoDB instance if the number of connections exceeds the upper limit or the performance of the instance deteriorates.

Context

 **Note** When an ApsaraDB for MongoDB instance is restarted, all its connections are terminated. Make appropriate service arrangements before you restart an ApsaraDB for MongoDB instance. Proceed with caution.

Procedure

- Log on to the [ApsaraDB for MongoDB console](#).
- On the **Replica Set Instances** page, find the target instance.
- Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
- In the upper-right corner of the page, click **Restart Instance**.

 **Note** You can also choose  > **Restart** in the **Operations** column corresponding to the target instance.

- In the **Restart Instance** message, click **OK**.

17.4.4. Change the specifications of an ApsaraDB for MongoDB instance

This topic describes how to change the specifications of an ApsaraDB for MongoDB instance. You can upgrade or downgrade an ApsaraDB for MongoDB instance to meet your business needs.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. Click **Change Configuration** in the upper right corner of the **Specification Information** section to go to the **Modify Instance** page.

 **Note** To go to the **Modify Instance** page, you can also choose  > **Change Configuration** in the **Operations** column corresponding to the target instance on the **Replica Set Instances** page.

5. On the **Modify Instance** page, change the instance specifications. You can change values of the following parameters:
 - **Node Type**
 - **Node Specifications**
 - **Storage Capacity**
6. Click **Submit**.

17.4.5. Change the name of an ApsaraDB for MongoDB instance

This topic describes how to change the name of an ApsaraDB for MongoDB instance.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. Click **Edit** to the right of **Instance Name**.

 **Note**

- The instance name must be 2 to 128 characters in length.
- It can contain letters, underscores (_), hyphens (-), and digits.
- It must start with a letter and cannot start with `http://` or `https://`.

5. Click **OK**.

17.4.6. Reset the password for an ApsaraDB for MongoDB instance

This topic describes how to reset your password in the ApsaraDB for MongoDB console.

Context

 **Notice** For data security, we recommend that you change your password on a regular basis.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Reset Password** in the **Operation** column and configure the parameters in the **Reset Password** pane. [Parameters for resetting a password](#) describes the parameter configurations.

Parameters for resetting a password

Parameter	Description
New Password	<ul style="list-style-type: none"> ◦ The password must be 8 to 32 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ◦ Special characters include ! # \$ % ^ & * () _ + - =
Confirm New Password	Enter the password again. The password you enter here must be the same as that in New Password.

6. Click **OK**.

17.4.7. Release an ApsaraDB for MongoDB instance

This topic describes how to manually release an ApsaraDB for MongoDB instance to meet your business needs.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the lower-right corner of the **Basic Information** section, click **Release**.

 **Note** You can also go to the **Replica Set Instances** page, find the target instance, click the  icon in the **Operations** column, and then choose **Release**.

5. In the **Release Instance** message, click **OK**.

 **Warning** After you release an ApsaraDB for MongoDB instance, data in the instance can no longer be recovered. Proceed with caution.

17.4.8. Back up an ApsaraDB for MongoDB instance

17.4.8.1. Configure automatic backup for an ApsaraDB for MongoDB instance

This topic describes how to configure automatic backup for an ApsaraDB for MongoDB instance. ApsaraDB for MongoDB automatically backs up data based on the backup policy you specify.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **Backup and Recovery** > **Backup Settings**. Configure the parameters in the Backup Settings pane.

[Backup policy parameters](#) describes the parameter configurations.

Backup policy parameters

Parameter	Description
Retention Days	The number of days for which you want to retain backup files. This parameter can only be set to seven days.
Backup Time	The hour at which you want to perform the backup task.
Day of Week	The backup cycle. You can select one or more days in a week.

5. Click **OK**.

17.4.8.2. Manually back up an ApsaraDB for MongoDB instance

This topic describes how to manually back up an ApsaraDB for MongoDB instance.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the upper-right corner of the page, click **Backup Instance**.
□
5. In the **Backup Instance** pane, click **OK**.

17.4.9. Monitoring

This topic describes the performance metrics provided by ApsaraDB for MongoDB to check the status of ApsaraDB for MongoDB instances. You can view these performance metrics in the ApsaraDB for MongoDB console.

Procedure

1. [Log on to the ApsaraDB for MongoDB console.](#)
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Monitoring Info**.

You can select a time range to query historical performance metrics. [Performance metrics](#) describes the metric details.

Performance metrics

Performance metric	Description	Monitoring interval
CPU Usage	cpu_usage: the CPU utilization of the instance.	300 seconds
Memory Usage	mem_usage: the memory usage of the instance.	300 seconds
IOPS Usage	The IOPS used by the instance. The following items are included: <ul style="list-style-type: none"> ◦ data_iops: the IOPS of the data disk ◦ log_iops: the IOPS of the log disk 	300 seconds
IOPS Usage	iops_usage: the ratio of the IOPS used by the instance to the maximum IOPS allowed.	300 seconds
Disk Space Usage	The total disk space used by the instance. The following items are included: <ul style="list-style-type: none"> ◦ ins_size: the total space used ◦ data_size: the space used by the data disk ◦ log_size: the space used by the log disk 	300 seconds
Disk Space Usage	disk_usage: the ratio of the total disk space used by the instance to the maximum disk space that can be used.	300 seconds

Performance metric	Description	Monitoring interval
Opcounters	<p>The queries per second (QPS) of the instance. The following items are included:</p> <ul style="list-style-type: none"> ◦ The number of insert operations ◦ The number of query operations ◦ The number of update operations ◦ The number of delete operations ◦ The number of getmore operations ◦ The number of command operations 	300 seconds
Connections	<p>current_conn: the number of current connections to the instance.</p>	300 seconds
Cursors	<p>The number of cursors used by the instance. The following items are included:</p> <ul style="list-style-type: none"> ◦ total_open: the number of cursors that are opened ◦ timed_out: the number of cursors that timed out 	300 seconds
Network	<p>The network traffic of the instance. The following items are included:</p> <ul style="list-style-type: none"> ◦ bytes_in: the inbound network traffic ◦ bytes_out: the outbound network traffic ◦ num_requests: the number of requests that are processed 	300 seconds
Global Lock	<p>The length of the queue waiting for global locks in the instance. The following items are included:</p> <ul style="list-style-type: none"> ◦ gl_cq_total: the length of the queue waiting for both global read and write locks ◦ gl_cq_readers: the length of the queue waiting for global read locks ◦ gl_cq_writers: the length of the queue waiting for global write locks 	300 seconds

Performance metric	Description	Monitoring interval
WiredTiger	<p>The cache metrics of the WiredTiger engine used by the instance. The following items are included:</p> <ul style="list-style-type: none"> ◦ <code>bytes_read_into_cache</code>: the amount of data that is read into the cache ◦ <code>bytes_written_from_cache</code>: the amount of data that is written from the cache to the disk ◦ <code>maximum_bytes_configured</code>: the size of the maximum available disk space that is configured 	300 seconds
Master-slave Delay	<p><code>repl_lag</code>: the latency in data synchronization between the primary and secondary nodes of the instance.</p>	300 seconds

17.5. Database connection

17.5.1. Use the mongo shell to connect to a replica set instance

This topic describes how to use the mongo shell to connect to a replica set instance. The mongo shell is a database management tool provided by ApsaraDB for MongoDB. You can install it on your client or in an ECS instance.

Prerequisites

- The version of the mongo shell is the same as your replica set instance. This ensures successful authentication. For information about the installation procedure, visit [MongoDB official documentation](#). Choose the version in the upper-left corner of the page based on your client version.
- The IP address of your client is added to a whitelist of the replica set instance. For more information, see [Configure a whitelist for an ApsaraDB for MongoDB instance](#).

Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Database Connection** to obtain the connection string of a node and the connection string URI.

 **Note** For more information about the connection strings, see [Overview of replica set instance connections](#).

5. Connect to the replica set instance from your client or ECS instance where the mongo shell is installed.
 - Single-node connection.

During regular tests, you can directly connect to a primary or secondary node. Take note that after the primary node fails, the system automatically switches to the secondary node, and the roles of connected nodes change. This affects the read and write operations of your application.

Command format:

```
mongo --host <host> -u <username> -p --authenticationDatabase <database>
```

- **<host>**: the connection string of the primary or secondary node.
 - **Primary**: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance.
 - **Secondary**: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance.
- **<username>**: the username you use to log on to a database of the replica set instance. The initial username is `root`.
- **<database>**: the name of the authentication database. It is the database where the database user is created. If the username is `root`, enter `admin`.

Example:

```
mongo --host dds-bp*****.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

When `Enter password:` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see [Reset the password for an ApsaraDB for MongoDB instance](#).

 **Note** The password you enter is not displayed.

- **HA connection (recommended)**: You can use a connection string URI to connect to both the primary and secondary nodes of a replica set instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

Command format:

```
mongo "<ConnectionStringURI>"
```

- The connection string URI must be enclosed in a pair of double quotation marks ("").
- **<ConnectionStringURI>**: the connection string URI of the replica set instance.

You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see [Reset the password for an ApsaraDB for MongoDB instance](#).

17.5.2. Use DMS to log on to a replica set instance of ApsaraDB for MongoDB

You can use DMS to connect to an ApsaraDB for RDS instance.

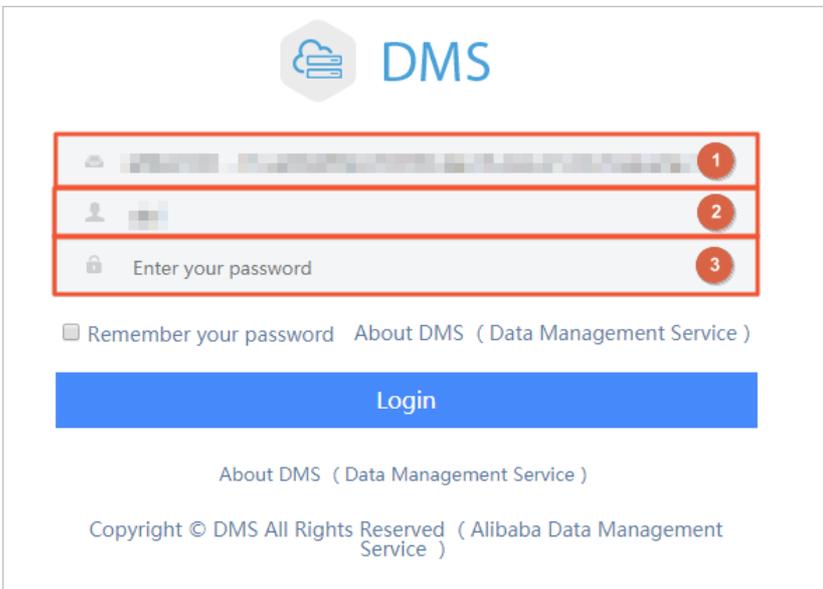
Prerequisites

The IP address whitelist is configured. For more information about how to configure the IP address whitelist, see [Configure a whitelist for an ApsaraDB for MongoDB instance](#).

Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).

2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the upper-right corner of the page, click **Logon** to go to the DMS logon page.
5. On the DMS logon page, set the following parameters:



- ①: the combination of the internal endpoint and port number of an instance, which is in the `<internal endpoint>:<port number>` format. For more information about how to view the internal endpoint and port number of an instance, see [Overview of replica set instance connections](#).
 - ②: the account used to access the instance.
 - ③: the password of the account used to access the database.
6. Click **Login**.

 **Note** If you want your web browser to remember the password, select **Remember your password** before you click **Login**.

17.5.3. Overview of replica set instance connections

This topic describes how to obtain connection strings and connection string URIs that are supported by ApsaraDB for MongoDB, as well as how to use them to connect to replica set instances. You can use a connection string to connect to either the primary or secondary node, and use a connection string URI to connect to both of them. For high availability, we recommend that you connect your application to both primary and secondary nodes by using connection string URIs.

View connection information

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, click **Database Connection** to view connection information.

Basic Information	Intranet Connection - Classic Network	
Accounts	Role	Address
Database Connection	Primary	dds-*****.mongodb.rds.intra.env17e.shuguang.com:3717
Backup and Recovery	Secondary	dds-*****.mongodb.rds.intra.env17e.shuguang.com:3717
Monitoring Info	ConnectionStringURI	mongodb://root:****@dds-*****.mongodb.rds.intra.env17e.shuguang.com:3717
Service Availability		

Description of connection information

Item	Description
Address	<ul style="list-style-type: none"> Intranet Connection - Classic Network: Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by security groups or whitelists. Intranet Connection - VPC: A VPC is an isolated network with higher security and performance than the classic network. By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Role	<ul style="list-style-type: none"> Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance.
Primary/Secondary	<p>The connection string of a primary or secondary node is in the following format:</p> <pre><host>:<port></pre> <ul style="list-style-type: none"> <host>: the endpoint you use to connect to the replica set instance. <port>: the port you use to connect to the replica set instance.
ConnectionStringURI	<p>A connection string URI is in the following format:</p> <pre>mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database] [? options]]</pre> <ul style="list-style-type: none"> mongodb://: the prefix of the connection string URI. It indicates a connection string URI. username:password@: the username and password you use to log on to a database of the replica set instance. You must separate them with a colon (:). hostX:portX: the endpoint and port of a node in the replica set instance. /database: the name of the authentication database. It is the database where the database user is created. ? options: additional connection options. <p>Note If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, when a node fails, the read and write operations of your application are not affected as a result of the failover.</p>

Related information

- [Connect to a replica set instance by using the mongo shell](#)

17.6. Security and audit

17.6.1. Configure a whitelist for an ApsaraDB for MongoDB instance

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or Classless Inter-Domain Routing (CIDR) blocks that you use for database access to a whitelist of this instance. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

The system creates a default whitelist for each instance. This whitelist can be modified or cleared, but it cannot be deleted. After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 0.0.0.0/0 to the default whitelist. The IP address 0.0.0.0/0 indicates that all IP addresses are allowed to access this instance. For database security, we recommend that you remove the IP address 0.0.0.0/0 and add only IP addresses or CIDR blocks that you allow to whitelists.

Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **Data Security** > **Whitelist Setting**.
5. You can manually configure a whitelist or import ECS Internal IP addresses to the whitelist. **Manually modify a whitelist**
 - i. Find the whitelist you want to modify and choose  > **Manually Modify** in the **Operation** column.
 - ii. Enter IP addresses or CIDR blocks.

Note

- Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix. The IP address prefix can consist of 1 to 32 bits.
- 0.0.0.0/0 or a blank field indicates that there is no IP access restrictions. In this case, the database may be at high security risk. We recommend that you set the access permission only for the IP address or CIDR block of your Web server.

Load IP addresses of ECS instances

- i. Find the target whitelist and choose  > **Import ECS Intranet IP** in the **Operation** column.
- ii. From the displayed internal IP addresses of ECS instances under the current account, find the target IP addresses and click  to add them to the whitelist.
- iii. Click **OK**.

17.6.2. Add or delete a whitelist

This topic describes how to add or delete whitelists that consist of the IP addresses allowed to access the databases.

Context

If your business involves multiple applications and you need to add an IP whitelist for each of them, you can sort the IP addresses into different whitelists.

Create a whitelist

1. Log on to the ApsaraDB for MongoDB console.
2. On the Replica Set Instances page, find the target instance.
3. Click the instance ID or choose  > Manage in the Operations column. Then, the Basic Information page appears.
4. In the left-side navigation pane, choose Data Security > Whitelist Setting.
5. Click Add a Whitelist Group in the upper-left corner of the page.
6. In the pane that appears on the right side of the page, configure Group Name and IP White List and click OK.

Note

- **Group Name:** The group name must be 2 to 32 characters in length, and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.
- **IP White List**
 - Separate multiple IP addresses with commas (,). You can add a maximum of 1,000 different IP addresses to a whitelist. Supported formats are IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix. The IP address prefix can consist of 1 to 32 bits.
 - 0.0.0.0/0 or a blank field indicates that there is no IP access restrictions. In this case, the database may be at high security risk. We recommend that you set the access permission only for the IP address or CIDR block of your Web server.

Delete an IP address whitelist

1. Log on to the ApsaraDB for MongoDB console.
2. On the Replica Set Instances page, find the target instance.
3. Click the instance ID or choose  > Manage in the Operations column. Then, the Basic Information page appears.
4. In the left-side navigation pane, choose Data Security > Whitelist Setting.
5. Find the target whitelist, and choose  > Delete Whitelist Group in the Operation column.

 **Note** You cannot delete the default whitelist.

6. In the Delete Whitelist Group message, click OK.

17.6.3. Audit logs

This topic describes audit logs provided in the ApsaraDB for MongoDB console. You can query the statement execution logs, operations logs, and error logs of an ApsaraDB for MongoDB instance to locate and analyze faults.

Context

The audit log feature records all operations that a client performs on a connected database. This feature provides references for you to perform fault analysis, behavior analysis, and security auditing because you can obtain the operation execution details from the audit logs. Audit logs are essential in the regulatory operations of Finance Cloud and other core business scenarios.

 **Note** Audit logs are stored for seven days, after which they are deleted.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. On the Replica Set Instances page, find the target instance.
3. Click the instance ID or choose  > Manage in the Operations column. Then, the Basic Information page appears.
4. In the left-side navigation pane, choose Data Security > Audit Log.
5. Click Enable Audit Log in the upper-left corner. In the Enable Audit message, click OK.

Result

On the Audit Log page, specify the time range, database name, database user, and keyword to query audit logs. You can also perform the following operations:

- **Export File:** exports an audit log file.
- **File List:** displays a list of audit logs.
- **Disable Audit Log:** stops the collection of information on database operations and deletes the saved audit logs.

17.6.4. Configure SSL encryption

This topic describes how to enhance link security by enabling Secure Sockets Layer (SSL) encryption and installing SSL CA certificates on your application services. The SSL encryption feature encrypts network connections at the transport layer to improve data security and ensure data integrity during communication.

Prerequisites

- The instance is a replica set instance.
- The database version of the instance is 3.4 or 4.0.

Impacts

When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Make appropriate service arrangements and make sure that your application can reconnect to the instance after it is disconnected.

 **Note** When an instance is restarted, all its nodes are restarted in turn, and each node goes through a transient connection of about 30 seconds. If the instance houses more than 10,000 collections, the transient connections last longer.

Precautions

- You can download SSL CA certificate files only from the ApsaraDB for MongoDB console.
- After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when necessary. For example, you can enable SSL encryption when you connect to an ApsaraDB for MongoDB instance over the Internet.

 **Note** In most cases, connections that use an internal endpoint do not require SSL encryption.

- After you enable SSL encryption for an instance, both SSL and non-SSL connections are supported.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **Data Security** > **SSL**.
5. Perform one of the following operations.

 **Note** When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Make appropriate service arrangements and make sure that your application can reconnect to the instance after it is disconnected.

Operation	Prerequisite	Procedure
Enable SSL encryption	The SSL encryption status is Disabled .	Turn on SSL Status . In the message that appears, click OK .
Update an SSL CA certificate	The SSL encryption status is Enabled .	Click Update Certificate . In the message that appears, click OK .
Download an SSL CA certificate file	The SSL encryption status is Enabled .	Click Download Certificate to download an SSL CA certificate file to your computer.
Disable SSL encryption	The SSL encryption status is Enabled .	Turn off SSL Status . In the message that appears, click OK .

17.6.5. Configure TDE

Transparent data encryption (TDE) encrypts and decrypts data files in real time. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the sizes of data files. When you use TDE, you do not need to modify your application that uses the ApsaraDB for MongoDB instance. To enhance data security, you can enable the TDE feature for an instance in the ApsaraDB for MongoDB console.

Prerequisites

The database version of the instance is 4.0.

 **Note** Before you enable TDE, you can create a MongoDB 4.0 instance to test the compatibility between your application and the database version. You can release the instance after the test is complete.

Impacts

- When you enable TDE, your instance is restarted, and your application is disconnected from the instance. We recommend that you enable TDE during off-peak hours and make sure that your application can reconnect to the instance after it is disconnected.
- TDE increases the CPU utilization of your instance.

Precautions

- You cannot disable TDE after it is enabled.
- You can enable TDE for an instance and disable encryption for a collection.

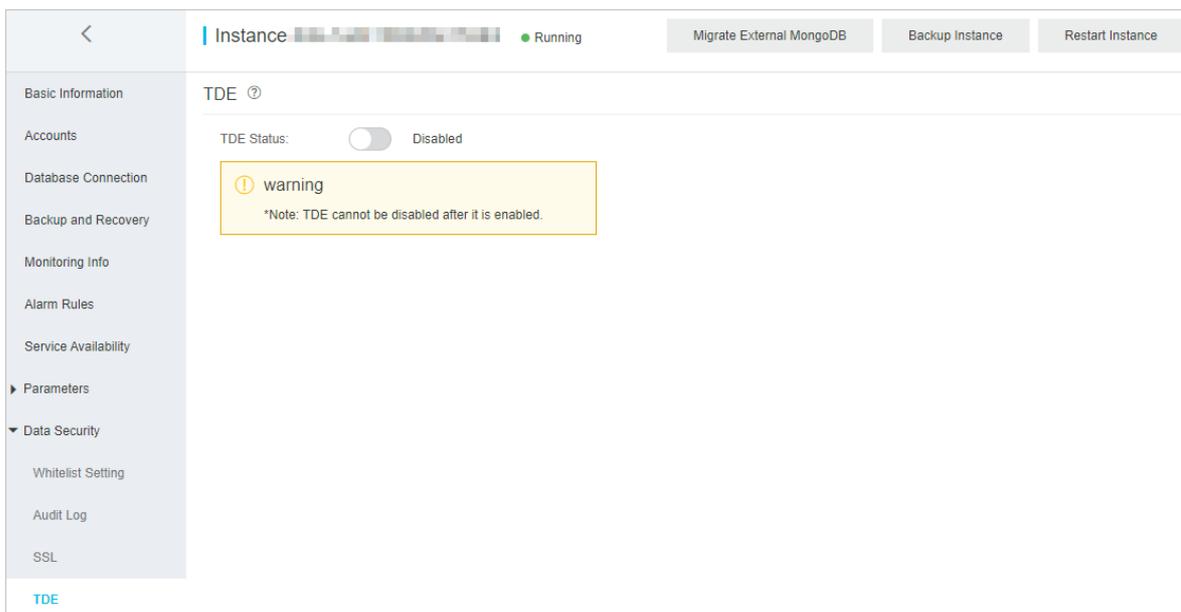
 **Note** In special business scenarios, you can choose not to encrypt a collection when you create it. For more information, see [Disable encryption for a specified collection](#).

- After you enable TDE, only new collections are encrypted. Existing collections are not encrypted.

- Key Management Service (KMS) generates and manages the keys used by TDE. ApsaraDB for MongoDB does not provide keys or certificates required for encryption.

Procedure

1. Log on to the [ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **Data Security** > **TDE**.
5. Turn on **TDE Status** to enable TDE.



6. In the **Restart Instance** dialog box, click **OK**.
The instance status changes to **Modifying TDE**. After the status changes to **Running**, TDE is enabled.

Disable encryption for a specified collection

After you enable TDE, all new collections are encrypted. In special business scenarios, you can choose not to encrypt a collection when you create it. To create a collection with encryption disabled, follow these steps:

1. Connect to a replica set instance by using the mongo shell. For more information, see [Connect to a replica set instance by using the mongo shell](#).
2. Run the following command to create a collection with encryption disabled:

```
db.createCollection("<collection_name>",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)"
}}})
```

 **Note** <collection_name>: the name of the collection.

Example

```
db.createCollection("customer",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)" }}})
```

17.6.6. Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB database in Secure Sockets Layer (SSL) encryption mode. SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity.

Prerequisites

- The instance is a replica set instance, and the database version of the instance is 3.4 or 4.0.
- SSL encryption is enabled for the instance. For more information, see [Configure SSL encryption](#).
- Mongo shell 3.0 or later is installed on the local server or ECS instance from which you want to connect to the database. For more information about the installation procedure, visit [MongoDB official documentation](#).
- The IP address of the local server or the ECS instance is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see [Configure a whitelist for an ApsaraDB for MongoDB instance](#).

Precautions

After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when necessary.

Procedure

A local server with a Linux operating system is used in the following example:

1. Download an SSL CA certificate package. For more information, see [Configure SSL encryption](#).
2. Decompress the package and upload the certificate files to the local server or the ECS instance where the mongo shell is installed.

 **Note** In this example, the `.pem` file is uploaded to the `/root/sslcafile/` directory of the local server.

3. On the local server or in the ECS instance, run the following command to connect to a database of the ApsaraDB for MongoDB instance:

```
mongo --host <host> -u <username> -p --authenticationDatabase <database> --ssl --sslCAFile <sslCAFile_path> -
--sslAllowInvalidHostnames
```

Note

- `<host>`: the connection string, including the port number, of the primary or secondary node in the ApsaraDB for MongoDB instance. For more information, see [Overview of replica set instance connections](#). If you want to connect to a database of the ApsaraDB for MongoDB instance over an internal network, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instance. If the network type is VPC, make sure that the two instances are in the same VPC.
- `<username>`: the username you use to log on to a database of the replica set instance. The initial username is root.
- `<database>`: the name of the authentication database. It is the database where the database user is created. If the username is root, enter admin.
- `<sslCAFile_path>`: the path of the SSL CA certificate files.

Example:

```
mongo --host dds-bpxxxxxxx-pub.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin --s
sl --sslCAFile /root/sslcafile/ApsaraDB-CA-Chain.pem --sslAllowInvalidHostnames
```

- When `Enter password:` is displayed, enter the password of the database user and press Enter.

 Note

- The password characters are not displayed when you enter the password.
- If you forget the password of the root user, you can reset it. For more information, see [Reset the password for an ApsaraDB for MongoDB instance](#).

17.7. CloudDBA

17.7.1. Authorize DAS to manage ApsaraDB for MongoDB instances

Database Autonomy Service (DAS) supports fast scaling, switchover, and centralized management of multiple environments. DAS is integrated into ApsaraDB for MongoDB to facilitate operations and maintenance (O&M). This topic describes how to authorize DAS to manage your ApsaraDB for MongoDB instances when you use the real-time performance, session, and capacity analysis features of CloudDBA for the first time.

Procedure

- [Log on to the ApsaraDB for MongoDB console](#).
- On the **Replica Set Instances** page, find the target instance.
- Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
- In the left-side navigation pane, choose **CloudDBA > Realtime performance**.
- Specify **Database Account and Password**, and click **Authorize**.

Result

The page is refreshed. Then, you can use the real-time performance, session, and capacity analysis features. You will not be prompted for authorization again.

17.7.2. Performance trends

This topic describes how to view performance trends in specific ranges, compare performance trends, and customize charts to view performance trends on your ApsaraDB for MongoDB instances.

Go to the Performance page

- [Log on to the ApsaraDB for MongoDB console](#).
- On the **Replica Set Instances** page, find the target instance.
- Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
- In the left-side navigation pane, choose **CloudDBA > Performance**.

 Note For more information about performance trends, see relevant topics in Database Autonomy Service (DAS) User Guide.

17.7.3. Real-time performance

This topic describes how to view real-time monitoring statistics of your ApsaraDB for MongoDB instances, such as read/write latency, queries per second (QPS), operations, connections, and network traffic.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage ApsaraDB for MongoDB instances. For more information, see [Authorize DAS to manage ApsaraDB for MongoDB instances](#).

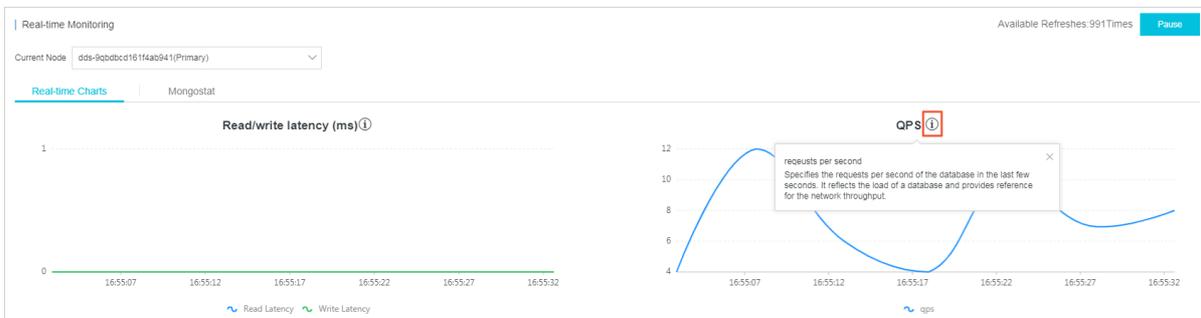
Procedure

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **CloudDBA** > **Realtime performance**.

Real-time Monitoring

On the Real-time Monitoring page, you can select the Real-time Charts or Mongostat tab to view monitoring statistics. When you refresh or go to the Real-time Monitoring page, the information on the Real-time Charts and Mongostat tabs is refreshed, and the number of available refreshes is reset in the upper-right corner.

Real-time Charts



The Real-time Charts tab is displayed on the Real-time Monitoring page. Line charts on the tab are refreshed every five seconds.

 **Note** For more information about individual metrics, click the  icon above each chart.

mongostat

time	query	insert	update	delete	getmore	cmd	dirty	used	qr/qw	ar/aw	vsize	mapped	in(Byte/s)	out(Byte/s)
16:55:33	0	0	0	0	4	35	0%	2%	0/0	0/0	1.6G	0	2.29 k	24.82 k
16:55:28	0	0	0	0	2	66	0.3%	2%	0/0	0/0	1.6G	0	2.04 k	14.32 k
16:55:22	0	0	0	0	4	43	0.3%	2%	0/0	0/0	1.6G	0	2.13 k	10.73 k
16:55:17	0	0	0	0	2	19	0.3%	2%	0/0	0/0	1.6G	0	1.30 k	9.76 k
16:55:13	0	0	0	0	2	59	0.3%	2%	0/0	0/0	1.6G	0	1.78 k	13.34 k
16:55:07	0	0	2	5	8	88	0.3%	2%	0/0	0/0	1.6G	0	3.98 k	13.09 k
16:55:02	0	0	0	0	4	19	0.29%	2%	0/0	0/0	1.6G	0	1.75 k	9.01 k
16:55:57	0	0	0	0	2	66	0.29%	2%	0/0	0/0	1.6G	0	1.91 k	13.92 k
16:55:52	0	0	0	0	4	22	0.29%	2%	0/0	0/0	1.6G	0	1.96 k	9.57 k
16:55:47	0	0	0	0	2	19	0.29%	2%	0/0	0/0	1.6G	0	1.19 k	9.01 k
16:55:42	0	0	0	0	4	36	0.29%	2%	0/0	0/0	1.6G	0	2.12 k	11.42 k
16:55:37	0	0	0	0	2	43	0%	2%	0/0	0/0	1.6G	0	1.69 k	11.79 k
16:55:32	0	0	0	0	4	36	0%	2%	0/0	0/0	1.6G	0	2.43 k	24.82 k
16:55:27	0	0	0	0	2	36	0.24%	2%	0/0	0/0	1.6G	0	1.48 k	11.96 k

Click the **Mongostat** tab. On the tab, you can view Mongostat command outputs. A new line of monitoring data is added every five seconds. The tab can contain up to 999 lines of information.

 **Note** For more information about Mongostat command outputs, visit [MongoDB official documentation](#).

17.7.4. Instance sessions

This topic describes how to view real-time monitoring statistics of your ApsaraDB for MongoDB instances, such as read/write latency, queries per second (QPS), operations, connections, and network traffic.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage your ApsaraDB for MongoDB instances. For more information, see [Authorize DAS to manage ApsaraDB for MongoDB instances](#).

View instance sessions

1. [Log on to the ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **CloudDBA > Session**.

-  **Note**
- If you turn on **Auto Refresh**, the system updates session data on the page every 30 seconds.
 - By default, the system displays only active sessions. You can turn on **Display All** to view both active and inactive sessions.
 - In the **Session Statistics** section, you can view information about sessions in the **Overview**, **Statistics by Client**, and **Statistics by Namespace** charts.

Terminate instance sessions

 **Warning** To avoid unexpected results, we recommend that you do not terminate system-level sessions.

1. In the **Instance Sessions** section, select the sessions you want to terminate and click **Kill Selected**.

Instance Sessions (Data Update Time:)

Refresh Auto Refresh Display All Enter a value Current Node dds-bp-... (Primary)

You can select multiple sessions by holding Shift. Inactive sessions cannot be killed. Kill Selected 2

Opld	A...	Operation	Operation Type	Time Spent (s) ↓↑	Execution Plan	Hostname	IP Address	Connection Description	Namespace
34631	●	{"getMore":5.894006457...	getmore	4	COLLSCAN	...	11.2...	conn68	local.oplog.rs
34630	●	{"getMore":5.926545304...	getmore	4	COLLSCAN	...	11.2...	conn65	local.oplog.rs
34703	●	{"currentOp":1.0,"\$all":1....	command	0		...	100. ...	conn1388	admin.\$cmd.aggregate
34701	●	{"find":"customer","filter":...	query	0	COLLSCAN	...	172. ...	conn686	db10.customer

2. In the message that appears, click OK.

17.7.5. Capacity analysis

This topic describes how to view information about the capacity analysis feature, including storage, exceptions, storage trend, tablespaces, and data space. The information helps you detect and resolve exceptions in the database storage to ensure database stability.

Prerequisites

Database Autonomy Service (DAS) is authorized to manage ApsaraDB for MongoDB instances. For more information, see [Authorize DAS to manage ApsaraDB for MongoDB instances](#).

Procedure

1. Log on to the [ApsaraDB for MongoDB console](#).
2. On the **Replica Set Instances** page, find the target instance.
3. Click the instance ID or choose  > **Manage** in the **Operations** column. Then, the **Basic Information** page appears.
4. In the left-side navigation pane, choose **CloudDBA** > **Capacity analysis**.
5. In the upper-right corner, click **Re-analyze**. Then, wait until the analysis is complete.
6. On the **Storage Overview** or **Data Space** tab, view the analysis results.



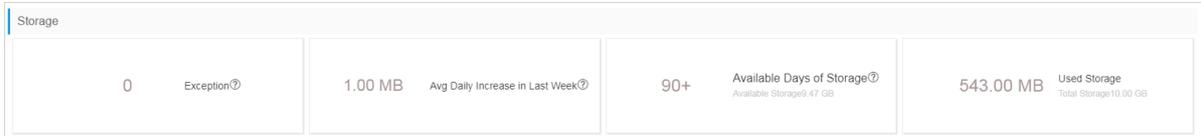
For more information about Storage Overview, see [Storage Overview](#).

For more information about Data Space, see [Data Space](#).

Storage Overview

On the Storage Overview tab, you can view information in the Storage, Exceptions, Storage Trend, and Tablespaces sections.

- Storage Overview



Item	Description
Exceptions	<p>The number of detected storage exceptions. ApsaraDB for MongoDB can detect the following types of exceptions:</p> <ul style="list-style-type: none"> Over 90% of the storage capacity is used. The total physical storage will be unavailable in seven days. The number of indexes in a collection exceeds 10.
Avg Daily Increase in Last Week	<p>The average daily increase of storage usage over the last seven days. Formula: (Storage usage at the time of collection - Storage usage seven days ago)/7.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p>Note</p> <ul style="list-style-type: none"> The increase speed is the average value during the seven days before the collection time. This parameter is only used as a reference for scenarios in which the traffic remains stable. Abrupt storage changes caused by batch imports, deletion of historical data, instance migration, or instance re-creation affect the accuracy of the data. </div>
Available Days of Storage	<p>The estimated number of days during which storage space is available. Formula: Size of available storage space/Average daily increase over the last week.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p>Note</p> <ul style="list-style-type: none"> 90+ indicates that the disk storage is sufficient for more than 90 days of usage. This parameter is used only as a reference for scenarios in which the traffic remains stable. Abrupt storage changes caused by batch imports, deletion of historical data, instance migration, or instance re-creation affect the accuracy of the data. </div>
Used Storage	The size of used storage space in contrast to the total size of storage space.

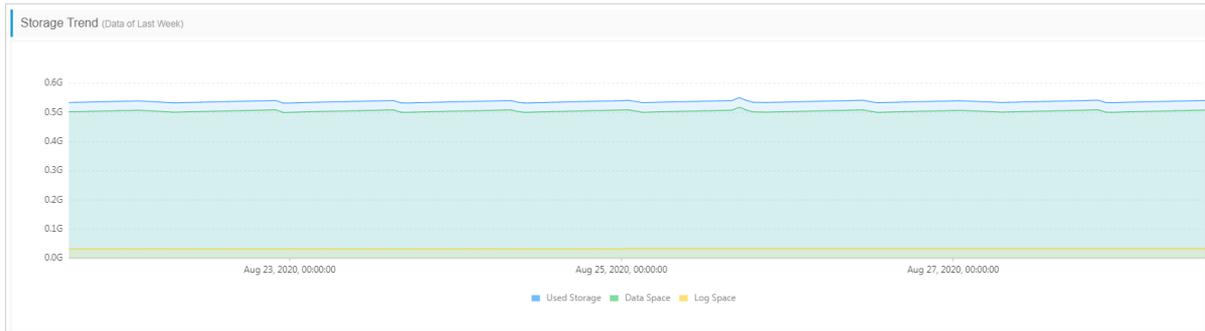
- Exceptions

Information about detected storage exceptions. You can resolve the exceptions based on information in this section.

Table/Collection Name (Click to View)	DB	Exception	Start Time
No storage exceptions found			

- Storage Trend

Changes of storage usage over the last week, such as changes of used storage, data space, and log space.



• **Tablespaces**

Information of all tables, such as the database name, storage engine, and collection storage.

Collection Name (Click to View) ↑	DB ↑	Storage Engine ↑	Collection Storage ↑	Collection Storage Percentage ↑	Index Storage ↑	Data Space ↑	Data Size ↑	Compression Percentage (↑) ↓	Collection Rows ↑	Avg Row Size ↑
No table information										

Note You can click a collection name to view its indexes.

Data Space

The Data Space tab shows the total storage capacity and tablespace information of each database.

Note

- You can click a data space to view its tablespace information.
- You can click a collection name to view its indexes.

Collection Name (Click to View) ↑	DB ↑	Storage Engine ↑	Collection Storage ↑	Collection Storage Percentage ↑	Index Storage ↑	Data Space ↑	Data Size ↑	Compression Percentage (↑) ↓	Collection Rows ↑	Avg Row Size ↑
No table information										

17.7.6. Slow query logs

This topic describes how to view slow query logs of ApsaraDB for MongoDB instances. You can locate, analyze, diagnose, and track slow query logs to create indexes, which improves the utilization of resources in the instances.

Procedure

- Log on to the ApsaraDB for MongoDB console.
- On the Replica Set Instances page, find the target instance.
- Click the instance ID or choose  > **Manage** in the Operations column. Then, the Basic Information page appears.
- In the left-side navigation pane, choose **CloudDBA > Slow query log**.

Note By default, slow query logs generated in the past 15 minutes are displayed in the trend chart. You can specify the time range and click Search to view slow query logs in specific periods of time. The maximum time range is one day.

5. View details of slow query logs by using one of the following methods: Method 1:
 - i. Click the **Slow Log Details** tab in the lower part of the page.
 - ii. On the **Slow Log Details** tab, select the database that you want to query.

Note If the request content of the target database is hidden, you can move the pointer over the corresponding request content and view the complete content.

Method 2:

- i. In the slow log trend chart, click the time of a specific slow query log and view its details on the **Slow Log Statistical** tab.



- ii. On the **Slow Log Statistical** tab, click **Sample** in the **Actions** column. In the **Slow Log Sample** dialog box, you can view details of the slow query log.

Slow Log Sample Note: Binary data in the sample is replaced with the \$binData string.

Execution Finish Time	Actions	Namespace	Request Content	User	Client	Avg Execution Duration (ms)	docsExamined	keysExamined	Avg Returned Rows
Aug 28, 2020, 14:04:25	ismaster	admin.\$cmd	["op":"command","ns":"admin.\$cmd","command":{"ismaster":1,"client":{"driver":...		11.200.150.7	295.00	-	-	-

Operation Type	Namespace	Request Template	Total Executions ↓↑	Avg Execution Duration (ms) ↓↑	Max Execution Duration (ms) ↓↑	Avg DocsExamined ↓↑	Max DocsExamined ↓↑	Avg KeysExamined ↓↑	Max KeysExamined ↓↑	Avg Returned Rows ↓↑	Max Returned Rows ↓↑	Actions
ismaster	admin.\$cmd	0	2	247.000	295	-	-	-	-	-	-	Sample Optimize
isMaster	admin.\$cmd	0	1	117.000	117	-	-	-	-	-	-	Sample Optimize

Note If the request content of the target database is hidden, you can move the pointer over the corresponding request content and view the complete content.

Export slow query logs

You can click **Export Slow Log** on the **Slow Log Statistical** tab to save the slow query log information to your computer.

18. ApsaraDB for OceanBase

18.1. What is ApsaraDB for OceanBase?

ApsaraDB for OceanBase is a financial-grade, distributed relational database service that features high performance, high availability, and high scalability. It supports active geo-redundancy and geo-disaster recovery to ensure high availability. It also supports high scalability to meet the increasing business requirements.

These features of ApsaraDB for OceanBase help you handle the challenges that are brought by rapid business growth. ApsaraDB for OceanBase also provides scalable and low latency database services in high throughput scenarios. This ensures improved user experience. For example, during the Double 11 in 2017, ApsaraDB for OceanBase handled all the transactions and payment requests. The maximum number of transactions that were made on Alipay reached 256,000 per second. The maximum number of processed requests per second reached 42 million. ApsaraDB for OceanBase accelerates the development of Internet finance.

The distributed engine of ApsaraDB for OceanBase uses the Paxos protocol and maintains multiple replicas. For the Paxos protocol, transactions can be committed only after they are approved by a majority of the acceptors. The Paxos protocol and multiple-replica design allow ApsaraDB for OceanBase to offer high availability and disaster recovery capabilities. ApsaraDB for OceanBase can help you achieve zero downtime. ApsaraDB for OceanBase supports high-availability architectures, such as active geo-redundancy and geo-disaster recovery. You can deploy the ApsaraDB for OceanBase service across data centers, regions, or continents. ApsaraDB for OceanBase provides financial-grade availability features and ensures strong consistency of transactions.

ApsaraDB for OceanBase is similar to an in-memory database and adopts a read/write splitting architecture. To ensure high efficiency for the storage engine, ApsaraDB for OceanBase stores baseline data in solid-state drives (SSDs) and stores incremental data in memory. This ensures that ApsaraDB for OceanBase offers high performance services. ApsaraDB for OceanBase is a cloud-based database service that supports multi-tenant data isolation. Each cluster of ApsaraDB for OceanBase can provide services for multiple tenants. The tenants are isolated so that they are not affected by each other.

ApsaraDB for OceanBase is compatible with most of the MySQL 5.6 features. This allows you to migrate MySQL-based services to ApsaraDB for OceanBase based on zero or small code modifications. This improves the efficiency of developing applications and migrating services. In ApsaraDB for OceanBase, you can create partitioned tables and use subpartitions. This serves as an alternative to MySQL sharding solutions. The ApsaraDB for OceanBase console provides an easy way for you to manage complex databases. For example, you can use the console to upgrade or downgrade instances, view performance data, and view optimization suggestions.

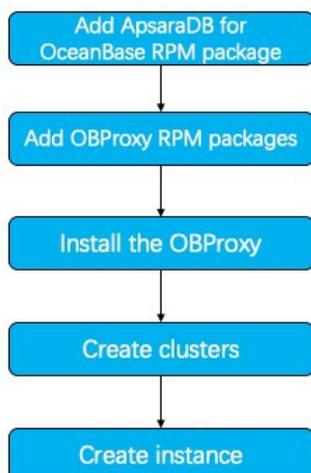
18.2. Quick start

18.2.1. Overview

This topic provides a quick start for ApsaraDB for OceanBase. This topic describes how to log on to the ApsaraDB for OceanBase console, add RPM packages of ApsaraDB for OceanBase and the OBProxy, install the OBProxy, create clusters, and create instances. After you perform these steps, you can use the created instances.

For more information about the steps, see [Quick start flowchart](#).

Quick start flowchart



The following section describes each step in the flowchart:

1. **Add ApsaraDB for OceanBase RPM packages**

You can use RPM packages or the source code to install ApsaraDB for OceanBase. In the ApsaraDB for OceanBase console, you can manage the RPM packages.

2. **Add OBProxy RPM packages**

In this step, add the RPM package of the OBProxy for ApsaraDB for OceanBase.

3. **Install the OBProxy**

The OBProxy is a reverse proxy server that is specific to ApsaraDB for OceanBase. ApsaraDB for OceanBase provides distributed relational database services. You can use the OBProxy to prevent transient connections and ensure that backend exceptions and operations such as breakdowns, upgrades, and network jitters are transparent to users. The OBProxy is also compatible with the MySQL protocol. The OBProxy supports strong checks, hot upgrades, and multiple clusters. In terms of frontend user requests, the OBProxy provides routing and forwarding services that feature high performance and high accuracy. In terms of backend server services, the OBProxy provides disaster recovery solutions that feature high availability and high scalability.

4. **Create clusters**

In this step, create an ApsaraDB for OceanBase cluster and configure the relevant parameters.

5. **Create instances**

In this step, create an ApsaraDB for OceanBase instance and configure the relevant parameters.

18.2.2. Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase

This topic describes how to use the Apsara Stack O&M system to log on to the Apsara Stack Operations console for ApsaraDB for OceanBase. The Google Chrome browser is used as an example in this topic.

Prerequisites

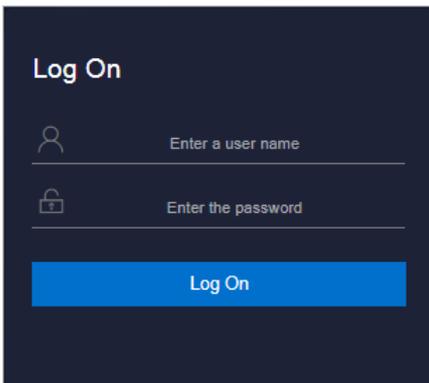
- The URL of the ASO console, and the username and password used for logging on to the console are obtained from the deployment personnel or an administrator.

The URL of the ASO console is in the following format: *region-id.aso.intranet-domain-id.com*.

- A browser is available. We recommend that you use Google Chrome.

Procedure

1. In the address bar, enter the URL *region-id.aso.intranet-domain-id.com* and press the Enter key.



Note You can select a language from the drop-down list in the upper-right corner of the page.

2. Enter your username and password.

Note Obtain the username and password for logging on to the ASO console from the deployment personnel or an administrator.

When you log on to the ASO console for the first time, you must change the password of your username as prompted.

To enhance security, a password must meet the following requirements:

- It must contain uppercase and lowercase letters.
 - It must contain digits.
 - It must contain special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).
 - It must be 10 to 20 characters in length.
3. Click **Log On** to go to the ASO console.
 4. In the left-side navigation pane, click **Products**. Then, choose **Product List > Database Services** and click **OceanBase Cloud Platform**. You are directed to the Apsara Stack Operations console for ApsaraDB for OceanBase.

18.2.3. Log on to the ApsaraDB for OceanBase console

This topic describes how to use the Apsara Stack Cloud Management (ASCM) console to log on to the ApsaraDB for OceanBase console. The Google Chrome browser is used as an example in this topic.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use Google Chrome.

Context

If you use the ASCM console to log on to the ApsaraDB for OceanBase console, you can only view monitoring information and manage instances. If you need to perform operations and maintenance (O&M) tasks on clusters, you must log on to the Apsara Stack Operations console. For more information, see [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase](#).

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.

2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, click **Products** and choose **Database Services > ApsaraDB for OceanBase**.
5. Select an organization from the **Organization** drop-down list and select a region from the **Region** drop-down list. Then, click **OceanBase Cloud Platform** to navigate to the ApsaraDB for OceanBase console.

18.2.4. Add ApsaraDB for OceanBase RPM packages

In the Apsara Stack Operations console for ApsaraDB for OceanBase, you can add an RPM Package Manager (RPM) package of the current ApsaraDB for OceanBase version.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **O&M**. In the list that appears, select **RPM Package Management**.
3. On the RPM Package Management page, click the **Database** tab. On the tab, click **Upload RPM Package**.
4. In the Upload RPM Package dialog box, select observer from the File Type drop-down list and click **Upload**. Then, select the RPM package that you want to upload from the local directory.
5. Click **OK**.

18.2.5. Add OBProxy RPM packages

In the Apsara Stack Operations console for ApsaraDB for OceanBase, you can add an RPM package of the OBProxy that matches the current ApsaraDB for OceanBase version.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **O&M**. In the list that appears, select **RPM Package Management**.
3. On the RPM Package Management page, click the **Database** tab. On the tab, click **Upload RPM Package**.
4. In the dialog box that appears, select obproxy from the File Type drop-down list, and click **Upload**. Then, select the RPM package that you want to upload.
5. Click **OK**.

18.2.6. Install the OBProxy

After you add an OBProxy RPM package, you can install the OBProxy.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **O&M**. In the list that appears, select **OBProxy**.

3. On the **Servers** tab of the **OBProxy** page, find the IP address of the **OBProxy** server to be installed. In the **Actions** column for the IP address, click **Install**.
4. In the dialog box that appears, set **OBProxy Name** to the name of the **OBProxy** based on your project name. From the **OBProxy Version** drop-down list, select an **OBProxy** version, such as **obproxy-1.5.0-1410335.el7.x86_64.rpm**. You can select the current time for **Start Time**.
5. Click **OK**.

18.2.7. Create clusters

Before you use **ApsaraDB for OceanBase**, create clusters in the **Apsara Stack Operations** console for **ApsaraDB for OceanBase**.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Clusters**.
3. On the page that appears, click **Create Cluster**. In the **Create Cluster** dialog box, configure the parameters as prompted. For more information about the parameters, see [Parameters for creating a cluster](#).

Parameters for creating a cluster

Parameter	Description
Cluster Name	The name of the ApsaraDB for OceanBase cluster.
OceanBase Version	The version of the ApsaraDB for OceanBase cluster. We recommend that you select the latest version.
Cluster Group	The group to which the cluster belongs. You can enter the name of the cluster group.
Cluster Model	The cluster model. If the minimal specifications are used for ApsaraDB for OceanBase , the cluster model is 1-1-1. You can set this parameter to 2-2-2 or 3-3-3 based on your cluster model. If five replicas are deployed, set this parameter to 2-2-2-2-2. To obtain the cluster model, contact the owner of the ApsaraDB for OceanBase project.
Data Center & Model Distribution	The data center and the machine model. Select the data center and the machine model based on the specified cluster model.

The screenshot shows the 'Create Cluster' interface. On the left is a navigation menu with 'Clusters' selected. The main area contains the following fields:

- Cluster Name:** A text input field with a placeholder 'Enter a cluster name'. Below it, a note states: 'The cluster name only supports English, numbers, and underscores, and cannot exceed 48 in length.'
- OceanBase Version:** A dropdown menu with the placeholder 'Select an OceanBase version'.
- Cluster Group:** A text input field with a placeholder 'Enter the name of the new cluster group'. Below it, a note states: 'The cluster group name only supports English, numbers, and underscores, and cannot exceed 48 in length.'
- Cluster Model:** A series of five input boxes for specifying server counts in each zone. To the right is a button labeled 'Data Center & Model Distribution'. Below this, a note states: 'The number of servers in each zone. After completing the cluster model configuration, add the information about the corresponding data center and model.'

At the bottom of the dialog are two buttons: 'Create' (in blue) and 'Cancel'.

4. After you specify the preceding parameters, click **Create**.

18.2.8. Create instances

After you create ApsaraDB for OceanBase clusters, you can create OceanBase instances in the ApsaraDB for OceanBase console.

Procedure

1. Log on to the [Apsara Stack Operations console for ApsaraDB for OceanBase](#).
2. In the left-side navigation pane, click **Instances**.
3. On the page that appears, click **Create Instance**. In the **Create Instance** dialog box, configure the parameters as prompted. [Parameters for creating an instance](#) describes the relevant parameters.

Parameters for creating an instance

Parameter	Description
OceanBase Version	The ApsaraDB for OceanBase version. Specify this parameter based on your business requirements.
Cluster Group	The cluster group where the instance is created.
Instance Name	The name of the instance. Specify this parameter based on your business requirements.
Zone	The zone where the instance is deployed. From the Zone drop-down list, select Default Zone.
Instance Specifications	The type of the instance. Specify this parameter based on your business requirements.
Tenant Whitelist	The tenant whitelist of the instance. Separate IP addresses or CIDR blocks with commas (,). The parameter value % indicates that all the tenants of the instance are added to the whitelist. We recommend that you use the default value %.
Tenants	The total number of tenants that are bound to the instance.

The screenshot shows the 'Create Instance' dialog box in the ApsaraDB for OceanBase console. The left sidebar contains navigation options: Dashboard, Clusters, Instances (selected), Alarm, Monitoring, O&M, Resources, and Systems. The main content area shows the 'Create Instance' form with the following fields:

- OceanBase Version:** A radio button group with options 1.0, 2.0, 2.1, and 2.2. 1.0 is selected.
- Cluster Group:** A dropdown menu with the text 'Select a cluster group'.
- Instance Name:** A text input field with the placeholder 'Enter an instance name'. Below it, a note states: 'The value must be 5 to 24 characters in length and can contain only digits, letters, and underscores (.).'
- Instance Specifications:** Two dropdown menus. The first is labeled 'Select an instance type' and the second is labeled 'Select a specification'.
- Zone:** A dropdown menu with the text 'Default Zone'.
- Tenant Whitelist:** A text input field with the placeholder '%'. Below it, a note states: 'The parameter value % indicates that all the tenants of the instance are added to the whitelist. We recommend that you use the default value %.'
- Tenants:** A text input field with the placeholder 'Enter the total number of tenants'.

At the bottom of the dialog box, there are two buttons: 'OK' and 'Cancel'.

4. Click **OK**.

18.3. Clusters

18.3.1. Overview

Before you use ApsaraDB for OceanBase, create ApsaraDB for OceanBase clusters in the ApsaraDB for OceanBase console. After you create clusters, you can view the basic information and the operation logs of the clusters. You can also restart, upgrade, scale out, and delete the clusters, and perform other operations on the clusters.

18.3.2. Scale out clusters online

In the ApsaraDB for OceanBase console, you can scale out ApsaraDB for OceanBase clusters based on your business requirements.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Clusters**. The Clusters page appears.
3. In the Actions column for the cluster that you want to scale out, choose **More > Scale Out Cluster**.
4. In the Scale out Cluster dialog box, specify **New Servers**. In the **Model** column, select the model of the servers that you want to add. Then, select the current date and time from the **Start Time** date and time picker and click **OK**.

18.3.3. Restart clusters

You can restart clusters in the ApsaraDB for OceanBase console.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Clusters**. The Clusters page appears.
3. In the Actions column for the cluster that you want to restart, choose **Routine Actions > Restart Cluster**.
4. In the Restart Cluster dialog box, select the zones of the cluster that you want to restart and click the **Start Time** field. In the date and time picker that appears, click **Now** and then click **OK**.

18.3.4. Delete clusters

You can delete clusters in the ApsaraDB for OceanBase console.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Clusters**. The Clusters page appears.
3. In the Actions column for the cluster that you want to remove, choose **More > Delete Cluster**.
4. In the Delete Cluster dialog box, click the **Start Time** field. In the date and time picker that appears, click **Now** and then click **OK**.

18.3.5. Upgrade clusters

In the ApsaraDB for OceanBase console, you can perform online upgrades to upgrade ApsaraDB for OceanBase clusters to the specified versions.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Clusters**. The Clusters page appears.
3. In the Actions column for the cluster that you want to upgrade, choose **Routine Operations > Cluster Upgrade**.

4. In the Cluster Upgrade dialog box, select all the rows. Each row corresponds to one zone of the cluster. From the Upgrade Version drop-down list, select the cluster version that you want to upgrade. Then, click the Start Time field. In the date and time picker that appears, click Now and then click OK.

18.3.6. View the monitoring information about clusters

In the ApsaraDB for OceanBase console, you can view the monitoring information about the major resources of each cluster, including CPU usage, memory usage, and disk usage.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Monitoring** and then click **OceanBase Data Trends**.
3. On the **OceanBase Data Trends** page, view the performance information. By default, the page shows the performance information about all the clusters. To view the performance information based on different dimensions, specify **Cluster Group**, **Clusters**, **Time**, **Zone**, and **Server** parameters based on your business requirements. Then, click **Search**.

18.3.7. View the real-time monitoring information about clusters

In the ApsaraDB for OceanBase console, you can view the real-time monitoring information about clusters, including queries per second (QPS) and transactions per second (TPS).

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Monitoring** and then click **Real-time Monitoring**. The Real-time Monitoring page appears.
3. On the **Real-time Monitoring** page, view the real-time monitoring information. By default, the page shows the performance information of all the clusters. To view the real-time performance information about a specific cluster group, click the name of the cluster group and then click **Search**. The **Real-time Performance Data** section appears. In this section, you can view the performance information about the cluster group.

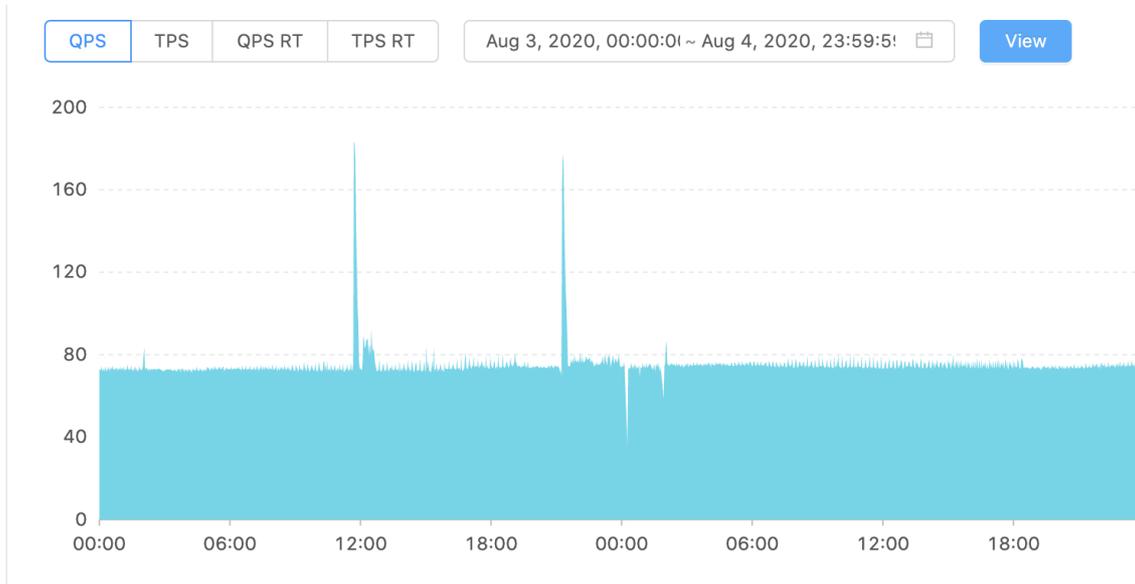
18.3.8. View performance metrics

You can view a wide range of performance metrics of clusters. For example, you can view the information about QPS, TPS, QPS response time (RT), TPS RT, top five clusters by major freeze time, O&M tasks, inspection data, and server usage.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#) After you log on to the ApsaraDB for OceanBase console, you can view the information about performance metrics on the **Dashboard** page. For example, you can view the information about QPS, TPS, QPS RT, TPS RT, top five clusters by major freeze time, O&M tasks, inspection data, and server usage. You can also view the performance trend of each metric by performing the following steps:
2. In the left-side navigation pane, click **Monitoring** and then click **OceanBase Data Trends**. The **OceanBase Data Trends** page appears.
3. On the **OceanBase Data Trends** page, click **QPS**, **TPS**, **QPS RT** or **TPS RT**. On the corresponding tab for the performance metric, specify the start time and end time for the search and click **View**.

Note In the lower part of the date and time picker for each performance metric, you can also select a time duration to specify the start time and the end time for the search. By default, the performance data in the last *X* period is displayed and the end time of the period is the current time. *X* represents the time duration that you select.



18.4. Instances

18.4.1. Overview

After you create instances, you can manage the instances. For example, you can view the basic information about the instances, reset the instance passwords, change the instance names, and start or stop the instances. You can also view the performance metrics, the change history, and the performance overview of each instance.

18.4.2. Change instance passwords

After you create instances, the default passwords for the instances are unknown. Therefore, you must log on to the console as an administrator and change the passwords.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Instances**. The **Instances** page appears.
3. In the **Instance Name** column, click the name of the instance for which you want to change the password. The **Basic Information** tab appears.
4. In the lower-left corner of the tab, click **Reset Password**.
5. In the **Reset Password** dialog box, enter and confirm the new password and click **OK**.

18.4.3. View instance details

You can view the basic information about instances, such as connection strings, instance information, and maintenance windows.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)

2. In the left-side navigation pane, click **Instances**. The **Instances** page appears.
3. In the **Instance Name** column, click the name of the instance whose basic information you want to view. On the **Basic Information** tab, view the basic information about the instance.

18.4.4. Change instance specifications

You can change instance specifications based on your business requirements.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Instances**. The **Instances** page appears.
3. In the **Instance Name** column, click the name of the instance whose specifications that you want to change. The **Basic Information** page appears.
4. In the lower-left corner of the page that appears, click **Modify Instance Specifications**.
5. In the **Modify Instance Specifications** dialog box, change the instance specifications.
6. Click **OK**.

18.4.5. Delete instances

To ensure efficient usage of resources, you can delete the instances that are no longer required.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Instances**. The **Instances** page appears.
3. In the **Instance Name** column, click the name of the instance that you want to delete. The **Basic Information** tab appears.
4. In the lower-left corner of the page that appears, click **Delete**.
5. In the message that appears, click **OK**.

 **Note** The instance cannot be recovered after the instance is deleted. Proceed with caution.

18.4.6. View the performance metrics of instances

You can view the major performance metrics of running instances, such as QPS and TPS.

Procedure

1. [Log on to the Apsara Stack Operations console for ApsaraDB for OceanBase.](#)
2. In the left-side navigation pane, click **Instances**. The **Instances** page appears.
3. In the **Instance Name** column, click the name of the instance whose performance metrics you want to view. The **Basic Information** tab appears.
4. Click the **Performance Metrics** tab. On the tab, view the performance information about the instance.

18.5. SQL syntax reference

18.5.1. Introduction to ApsaraDB for OceanBase SQL

ApsaraDB for OceanBase supports Structured Query Language (SQL). SQL is a computer language that you can use to organize, manage, and retrieve data in databases.

ApsaraDB for OceanBase is fully compatible with the MySQL protocol. Therefore, users of ApsaraDB for OceanBase SQL can connect to ApsaraDB for OceanBase by using MySQL clients, such as MySQL command-line, Java, and C clients.

18.5.2. Overview of ApsaraDB for OceanBase SQL

18.5.2.1. Identifiers

Use identifiers

In ApsaraDB for OceanBase, identifiers are the names of schema objects in SQL statements. The schema objects include tenants, databases, tables, views, indexes, table groups, columns, and aliases.

Before you use ApsaraDB for OceanBase identifiers, pay attention to the following considerations:

ApsaraDB for OceanBase identifiers describes the maximum length and the supported characters for each identifier.

ApsaraDB for OceanBase identifiers

Identifier	Maximum length (bytes)	Supported character
Username	16	A username can contain letters, digits, and underscores (_). It must start with a letter or an underscore (_). You cannot use ApsaraDB for OceanBase keywords when you specify the name.
Tenant name	64	A tenant name can contain letters, digits and underscores (_). It must start with a letter or an underscore (_). You cannot use ApsaraDB for OceanBase keywords when you specify the name.
Database name	64	A database name can contain letters, digits, underscores (_), and dollar signs (\$).
Table group name	64	A table group name can contain letters, digits and underscores (_). It must start with a letter or an underscore (_). You cannot use ApsaraDB for OceanBase keywords when you specify the name.
Table name	64	A table name can contain letters, digits, underscores (_), and dollar signs (\$).
Column name	64	A column name can contain letters, digits, underscores (_), and dollar signs (\$).
Index name	64	An index name can contain letters, digits, underscores (_), and dollar signs (\$).
Alias	255	An alias can contain letters, digits, underscores (_), and dollar signs (\$).

Identifier	Maximum length (bytes)	Supported character
Variable name	64	A variable name can contain alphanumericals, periods (.), underscores (_), and dollar signs (\$).

 Note

- In addition to the limits in the preceding table, you must meet some other identifier limits. For example, each identifier cannot contain the ASCII control character 0 or the eight-bit byte that represents 255.
- The names of databases, tables, or columns cannot end with spaces.
- We recommend that you do not use quotation marks in identifiers.

You can determine whether to use backticks (`) to enclose each identifier based on the actual scenarios. If identifiers are reserved keywords or contain special characters, you must use backticks (`) to enclose the identifiers. In ApsaraDB for OceanBase, you can use only backticks (`) to enclose identifiers.

```
mysql> SELECT * FROM `select` WHERE `select`.id > 100;
```

Reference identifiers

ApsaraDB for OceanBase allows you to use a name that consists of one or more identifiers. If a name consists of multiple identifiers, separate the identifiers with periods (.).

[Referable columns and descriptions](#) lists the columns that you can reference in ApsaraDB for OceanBase.

Referable columns and descriptions

Column reference	Description
col_name	The col_name column in the table. The query retrieves data from the table.
tbl_name.col_name	The col_name column in the tbl_name table of the default database.
db_name.tbl_name.col_name	The col_name column in the tbl_name table of the db_name database.

If the identifiers that compose a name must be enclosed in backticks (`), enclose each identifier instead of the entire name. For example, `my-tables`.`my-column` is valid and `my-tables.my-column` is invalid.

If columns can be uniquely identified by column names, you do not need to specify the tbl_name prefix or the db_name.tbl_name prefix for the columns in statements. If columns cannot be uniquely identified by column names, you must specify one of the two prefixes. For example, the t1 table contains one c column and the t2 table also contains one c column. You want to execute a single SELECT statement to retrieve the c column in the t1 and t2 tables. In this scenario, the c column cannot be uniquely identified by the column name because this column exists in both t1 and t2 tables. You must specify prefixes for the two c columns to distinguish the c column in the t1 table from the c column in the t2 table: t1.c and t2.c. Similarly, you may need to execute a single statement to retrieve columns from the t table in the db1 database and the t table in the db2 database. In this scenario, you must use db1.t.col_name and db2.t.col_name to specify the databases that store the t tables.

In qualified names, each word that follows a period must be an identifier. Therefore, the word does not need to be enclosed in backticks (`). This rule applies even if the word is a reserved keyword.

The `.tbl_name` syntax specifies the `tbl_name` table in the current database. This syntax is compatible with the Open Database Connectivity (ODBC) API because table names are also prefixed with periods (`.`) in some ODBC programs.

Identifier case sensitivity

In ApsaraDB for OceanBase, identifiers are not case-sensitive. To ensure compatibility with MySQL, ApsaraDB for OceanBase introduces the `lower_case_tables_name` system parameter that is used in MySQL. This system parameter specifies whether table names and database names that function as identifiers are case-sensitive.

[Description of lower_case_tables_name values](#) describes the values of the `lower_case_tables_name` parameters.

Description of lower_case_tables_name values

Value	Description
0	Table names and database names are case-sensitive.
1	Table names and database names are not case-sensitive.

18.5.2.2. Supported SQL statements

ApsaraDB for OceanBase supports the following SQL statements:

- Data definition language (DDL) statements
CREATE DATABASE, ALTER DATABASE, DROP DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, DROP VIEW, ALTER VIEW, TRUNCATE TABLE, and RENAME TABLE
- Data manipulation language (DML) statements
INSERT, REPLACE, DELETE, UPDATE, and SELECT statements and clauses
- Transaction statements
START TRANSACTION, COMMIT, and ROLLBACK
- Prepared statements
PREPARE, SET, EXECUTE, DEALLOCATE, DROP PREPARE, and DROP PREPARE
- Database administration statements
CREATE RESOURCE UNIT, DROP RESOURCE UNIT, CREATE RESOURCE POOL, ALTER RESOURCE POOL, DROP RESOURCE POOL, CREATE TENANT, ALTER TENANT, DROP TENANT, CREATE TABLEGROUP, DROP TABLEGROUP, ALTER TABLEGROUP, CREATE USER, DROP USER, RENAME USER, ALTER USER, GRANTS, REVOKE, SET, SET PASSWORD, SET GLOBAL, ALTER SYSTEM, GRANT, and REVOKE
- Useful SQL statements
SHOW, KILL, USE, DESCRIBE, EXPLAIN, HINT, HELP, and other SQL statements

18.5.2.3. SQL limits

This topic describes the following limits of SQL statements in ApsaraDB for OceanBase:

- ApsaraDB for OceanBase does not support custom data types or custom functions.
- ApsaraDB for OceanBase does not support updatable views, stored procedures, triggers, or cursors.
- ApsaraDB for OceanBase does not support temporary tables.
- ApsaraDB for OceanBase does not support compound statements, such as BEGIN...END, LOOP...END LOOP, REPEAT ...UNTIL...END REPEAT, and WHILE...DO...END WHILE.
- ApsaraDB for OceanBase does not support flow control statements, such as IF and WHILE.
- The INSERT or REPLACE statements that contain SELECT clauses cannot be executed to modify data. The DELETE statements cannot be executed to delete data from multiple tables. The UPDATE statements cannot

be executed to update data that is stored in multiple tables.

- The following list describes the quantity and maximum length limits:
 - When a table or an index is created, the maximum length of the primary key is 16 KB and the maximum length of a single row is 1.5 MB.
 - A single column of the VARCHAR data type can store a maximum of 262,143 bytes. The maximum length for the column of this type is 256 KB.
 - You can create a maximum of 64 primary keys.
 - You can store a maximum of 512 columns in a single table.
 - You can create a maximum of 128 indexes for a single table.
- The `SELECT... FOR UPDATE` statement supports only single-table queries.
- In ApsaraDB for OceanBase, you can execute the TRUNCATE TABLE statement on only a table that has one partition.

18.5.3. Partitions

18.5.3.1. Overview

ApsaraDB for OceanBase is a distributed database. If you need to distribute data in a table to multiple servers, you must create the table as a partitioned table. ApsaraDB for OceanBase automatically distributes data across ApsaraDB for OceanBase servers based on your partitions. You can also create non-partitioned tables. The data in each non-partitioned table is stored on only one ApsaraDB for OceanBase server.

The following list describes the characteristics of ApsaraDB for OceanBase partitioned tables:

- ApsaraDB for OceanBase supports `hash partitioning` , `key partitioning` and `range partitioning` .
- Tables are partitioned based on the partition key fields that you specify in the `CREATE TABLE` statements.
- The number of partitions in each partitioned table is specified in the `CREATE TABLE` statement.
- You can create non-partitioned tables. For example, you can create metadata tables that store only a few rows of service data as non-partitioned tables.
- In ApsaraDB for OceanBase, table data is distributed based on partitions. Therefore, to ensure high performance, the WHERE clauses in INSERT, REPLACE, SELECT, UPDATE, and DELETE statements must contain `PARTITION(partition_list)` . If this field is not contained, the system reports an error.
- In ApsaraDB for OceanBase, you can create a maximum of 8,192 partitions in a single partitioned table.

Introduction to partitions

In ApsaraDB for OceanBase, you can create partitions for each table based on the specified rules. The partitions can be distributed across ApsaraDB for OceanBase servers.

Partitioning functions specify the rules that you use to distribute data across partitions. Partitioning functions must be system functions.

Partitioning functions can be modulus functions, internal hash functions, or linear hash functions. To simplify the partitioning process, you can also use partitioning functions to match data against a range of continuous numeric values or a list of numeric values.

Partitioning functions are selected based on the partitioning types that you use. The partitioning functions use the values of the expressions that you provide as arguments. The expressions can represent the column values of the INT type. The expressions can also be the functions that are applied on one or more column values and return integers. The values of the expressions are passed to the partitioning functions. Then, the partitioning functions return sequence numbers that specify the partitions where the specific rows are stored. The specific rows are distinguished by the expression values.

You cannot use the partitioning functions to represent constants or random numbers.

The partitioning functions can use valid SQL expressions that return positive values. The positive values must be smaller than the maximum allowed positive integer that is specified by MAXVALUE.

Partitioning must be implemented on both data and indexes of each partitioned table. This means that you must create partitions for both data and indexes for each entire partitioned table. You cannot create partitions for only partial data of each partitioned table.

Partitioning benefits

- Partitioning offers an easy way for you to delete and add data. In most cases, you can delete the data that does not need to be stored by deleting the partitions that store the data. You can add data by creating partitions for the data.
- Partitioning allows you to optimize some queries. You can store the data that meets the conditions in the specified WHERE clauses in one or more partitions. In this scenario, when you run the corresponding queries, the system does not need to scan the other partitions. After you create partitioned tables, you can modify the partitions of the partitioned tables. Therefore, if this query optimization method is not used when you configure the partitioning scheme for the first time, you can modify the partitions to reorganize your data. This improves the efficiency of frequently run queries.
- Partitioning offers an easy method for you to process the queries that involve aggregate functions such as SUM() and COUNT() in parallel. The `SELECT salesperson_id, COUNT(orders) as order_total FROM sales GROUP BY salesperson_id;` statement is used as an example. In this example, the query can be performed on each table partition in parallel. The statement returns the final result by summarizing the result of each partition.
- Partitioning allows you to run data queries across disks to maximize the I/O throughput.

Partitioning types

The following list describes the supported partitioning types:

- Range partitioning

Rows are assigned to partitions based on a range of continuous column values that you specify for each partition.

- Hash partitioning

Data is assigned to partitions based on the results that are returned by user-defined expressions. The expressions use the column values in the rows that are to be inserted into the specified tables to calculate results.

Hash functions can contain valid expressions that return non-negative integers in ApsaraDB for OceanBase.

Partitions are automatically numbered when you create the partitions, regardless of partitioning types. The sequence numbers of partitions start from 0. When you insert a row into a partitioned table, the system uses the partition sequence number to identify the partition into which the row is inserted.

For example, if your partitioned table has four partitions, the sequence numbers of the four tables are 0, 1, 2, and 3. If you use range partitioning, make sure that you define a range for each partition. If you use hash partitioning, each user-defined function must return an integer that is greater than zero. Partition names are not case-sensitive.

- Key partitioning

Key partitioning is similar to hash partitioning. Hash partitioning uses user-defined expressions, and key partitioning uses hash functions that are provided by ApsaraDB for OceanBase servers. The column values on which key partitioning is implemented can be integers or values of other data types.

 **Note** Key partitioning in ApsaraDB for OceanBase uses MurmurHash functions.

18.5.3.2. Range partitioning

Syntax

```

...
PARTITION BY RANGE {(expr) | COLUMNS(column_list)}
  (partition_definition [, partition_definition] ...)
partition_definition:
  PARTITION partitionname
  VALUES {LESS THAN {(expr | value_list) | MAXVALUE}

```

For range partitioning, the specified ranges must be continuous and cannot overlap with each other. You must define the ranges by using the `VALUES LESS THAN` operator.

Examples

If you use range partitioning, each partition contains the rows in which the values fall in the specified range of continuous numeric values. The row values are returned by the expressions that you specify to implement range partitioning.

In the following examples, the `employees` table is used. The table stores employee records for 20 video stores that are numbered from 1 to 20.

The `employees` table is created by executing the following statement:

```

CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT NOT NULL,
  store_id INT NOT NULL
);

```

You can use multiple methods to implement range partitioning on the `employees` table based on your business requirements.

In one of these methods, you can use the `store_id` column. For example, you can split the table into four partitions by adding a `PARTITION BY RANGE` clause to the statement.

```

CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT NOT NULL,
  store_id INT NOT NULL
)
PARTITION BY RANGE (store_id) (
  PARTITION p0 VALUES LESS THAN (6),
  PARTITION p1 VALUES LESS THAN (11),
  PARTITION p2 VALUES LESS THAN (16),
  PARTITION p3 VALUES LESS THAN (21)
);

```

In this partitioning scheme, p0 stores the rows for the employees of stores 1 to 5, p1 stores the rows for the employees of stores 6 to 10, and so on.

 **Notice** The sequence numbers of the stores are sorted in ascending order. Each partition is defined based on the specified range of store sequence numbers. This meets the requirements of the `PARTITION BY RANGE` syntax. The requirements are similar to the syntax requirements of C or Java `switch ... case` statements.

A new row that contains the data such as `72, 'Michael', 'Widenius', '1998-06-25', NULL, 13` can be inserted into the p2 partition as expected. However, if you need to insert rows for store 21, errors are reported. This is because no rules cover the rows whose `store_id` values are greater than 20 and ApsaraDB for OceanBase servers cannot determine the partitions that store the rows.

To prevent errors of this type, you can execute a `CREATE TABLE` statement that includes the `catchall VALUES LESS THAN` clause. You can use the clause to specify the partition in which you store the values that are greater than the explicitly specified maximum value.

```

CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT NOT NULL,
  store_id INT NOT NULL
)
PARTITION BY RANGE (store_id) (
  PARTITION p0 VALUES LESS THAN (6),
  PARTITION p1 VALUES LESS THAN (11),
  PARTITION p2 VALUES LESS THAN (16),
  PARTITION p3 VALUES LESS THAN MAXVALUE
);

```

MAXVALUE specifies the maximum allowed integer. All the rows whose store_id column values are greater than or equal 16 are stored in the p3 partition. In this example, the number 16 is the maximum value that is specified in an explicit way. When the number of stores increases to 25, 30, or another larger value, you can execute the ALTER TABLE statement to add partitions for the added stores. For example, you can add a partition for stores 21 to 25 and add another partition for stores 26 to 30.

The table in the following example has the same schema as that in the preceding example. You can partition the table based on the continuous ranges of job_code column values. The job_code column values specify the job codes of employees. In this example, two-digit job codes represent regular in-store workers, three-digit job codes represent office and support personnel, and four-digit codes represent management personnel. You can execute the following statement to create the partitioned table:

```
CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT NOT NULL,
  store_id INT NOT NULL
)
PARTITION BY RANGE (job_code) (
  PARTITION p0 VALUES LESS THAN (100),
  PARTITION p1 VALUES LESS THAN (1000),
  PARTITION p2 VALUES LESS THAN (10000)
);
```

In this example, the rows for the in-store workers are stored in the p0 partition. The rows for the office and support personnel are stored in the p1 partition, and the rows for the management personnel are stored in the p2 partition.

You can also use an expression in the VALUES LESS THAN clause. Note that the returned values of the expression must be able to be used for LESS THAN (<) comparison to determine the ranges into which the returned values fall. Therefore, the returned values cannot be NULL. Due to the same reason, the values in the hired, separated, job_code, and store_id columns of the employees table are defined as non-NULL values.

In the first example, the employees table is partitioned based on the sequence numbers of stores. You can also partition the table by using an expression based on onboarding dates or offboarding dates. For example, you want to partition the table based on the YEAR (separated) values. The values specify the years when employees left the company. In this example, the following CREATE TABLE statement is executed to implement the partitioning scheme:

```
CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT,
  store_id INT
)
PARTITION BY RANGE (YEAR(separated)) (
  PARTITION p0 VALUES LESS THAN (1991),
  PARTITION p1 VALUES LESS THAN (1996),
  PARTITION p2 VALUES LESS THAN (2001),
  PARTITION p3 VALUES LESS THAN MAXVALUE
);
```

In this partitioning scheme, the p0 partition stores the rows for the employees who left the company before 1991. The p1 partition stores the rows for the employees who left the company between 1991 and 1995. The p2 partition stores the rows for the employees who left the company between 1996 and 2000. The p3 partition stores the rows for the employees who left the company after 2000.

Scenarios

- You need to delete the previous data.
- You need to use a column that stores date or time values that are sorted in ascending order.
- You need to frequently run the queries that are based on partition key columns.

18.5.3.3. Hash partitioning

Syntax

If you use hash partitioning, you must specify column values or expressions for the columns on which hashing is to be implemented. You must also specify the number of partitions for each partitioned table. ApsaraDB for OceanBase automatically implements hash partitioning based on the specified settings.

```
...
PARTITION BY HASH (expr)
PARTITIONS num
```

If you use hash partitioning to partition a table, you must add the following clause to the `CREATE TABLE` statement: `PARTITION BY HASH (expr)`. In the clause, `expr` specifies an expression that returns integers. You can also specify `expr` as the name of a column. The column must store integers in ApsaraDB for OceanBase. You may need to append a `PARTITIONS num` clause to the `PARTITION BY HASH (expr)` clause. In the `PARTITIONS num` clause, `num` specifies the number of partitions for the table. You must specify this parameter as a positive integer.

Examples

You can partition the table based on the years when employees were hired by executing the following statement:

```
CREATE TABLE employees (
  id INT NOT NULL,
  fname VARCHAR(30),
  lname VARCHAR(30),
  hired DATE NOT NULL DEFAULT '1970-01-01',
  separated DATE NOT NULL DEFAULT '9999-12-31',
  job_code INT,
  store_id INT
)
PARTITION BY HASH(YEAR(hired))
PARTITIONS 4
```

Scenarios

Hash partitioning ensures that data in tables is evenly distributed across table partitions. The number of partitions for each table is specified when you create the table.

18.5.3.4. Key partitioning

Syntax

If you use key partitioning to partition tables, the tables are partitioned based on the hash functions that are provided by ApsaraDB for OceanBase. Key partitioning uses one column or a list of columns as the partition key.

```
...
PARTITION BY KEY(column_list)
PARTITIONS num
```

Examples

Execute the following statement to create the k1 table and partition the table based on the id field:

```
mysql> create table k2(id int primary key, name varchar(20))
-> partition by key() partitions 2;
Query OK, 0 rows affected (0.29 sec)
```

In this example, id is the primary key field. `partition by key()` and `partition by key(id)` are equivalent.

18.5.3.5. Subpartitioning

Syntax

Subpartitioning is also known as composite partitioning. Subpartitioning divides each partition of a partitioned table into subpartitions.

```

...
PARTITION BY RANGE(expr)
SUBPARTITION BY [HASH|KEY](expr) SUBPARTITIONS N
(PARTITION P0 VALUES LESS THAN (V0), ...)
PARTITION BY [HASH|KEY](expr)
SUBPARTITION BY RANGE(expr) SUBPARTITION TEMPLATE (SUBPARTITION P0
VALUES LESS THAN (V0), ...)
PARTITIONS N
...

```

Examples

```

CREATE TABLE ts (id INT, purchased DATE, PRIMARY KEY (id, purchased))
PARTITION BY RANGE(YEAR(purchased))
SUBPARTITION BY HASH(TO_DAYS(purchased))
SUBPARTITIONS 2
(
  PARTITION p0 VALUES LESS THAN (1990),
  PARTITION p1 VALUES LESS THAN (2000),
  PARTITION p2 VALUES LESS THAN MAXVALUE
)

```

In this example, the `ts` table has three range partitions: `p0`, `p1`, and `p2`. Each partition is divided into two subpartitions. As a result, the entire table is divided into six partitions. The number of partitions is calculated based on the formula: $3 \times 2 = 6$. Based on the settings in the `PARTITION BY RANGE` clause, the first two partitions store only the rows whose values in the `purchased` column are smaller than 1990. The preceding statement is equivalent to the following statement:

```

CREATE TABLE ts (id INT, purchased DATE, PRIMARY KEY (id, purchased))
PARTITION BY RANGE(YEAR(purchased))
SUBPARTITION BY HASH(TO_DAYS(purchased))
(
  PARTITION p0 VALUES LESS THAN (1990)
  (
    SUBPARTITION s0,
    SUBPARTITION s1
  ),
  PARTITION p1 VALUES LESS THAN (2000)
  (
    SUBPARTITION s2,
    SUBPARTITION s3
  ),
  PARTITION p2 VALUES LESS THAN MAXVALUE
  (
    SUBPARTITION s4,
    SUBPARTITION s5
  )
);

```

Notes

Pay attention to the following syntax considerations when you implement subpartitioning:

- Each partition must have the same number of subpartitions.
- The primary key must cover all the partition key columns.
- If subpartitions are created based on ranges, we recommend that you use subpartitioning templates to create subpartitions.

```

CREATE TABLE ts (id INT, purchased DATE, PRIMARY KEY (id, purchased))
PARTITION BY HASH(id)
SUBPARTITION BY RANGE(YEAR(purchased))
SUBPARTITION TEMPLATE
(
  SUBPARTITION p0 VALUES LESS THAN (1990),
  SUBPARTITION p1 VALUES LESS THAN (2000),
  SUBPARTITION p2 VALUES LESS THAN MAXVALUE
)
PARTITIONS 2

```

- If you need to use the `SUBPARTITION` clause to define subpartitions for a partition of a partitioned table in an explicit way, you must define all the subpartitions of the partitioned table. Otherwise, the following statement fails to be executed:

```
CREATE TABLE ts (id INT, purchased DATE, PRIMARY KEY (id, purchased))
PARTITION BY RANGE(YEAR(purchased))
SUBPARTITION BY HASH(TO_DAYS(purchased))
(
  PARTITION p0 VALUES LESS THAN (1990)
  (
    SUBPARTITION s0,
    SUBPARTITION s1
  ),
  PARTITION p1 VALUES LESS THAN (2000),
  PARTITION p2 VALUES LESS THAN MAXVALUE
  (
    SUBPARTITION s2,
    SUBPARTITION s3
  )
)
```

An execution failure occurs even if the statement includes the `SUBPARTITIONS 2` clause.

- Each `SUBPARTITION` clause must include at least the subpartition name. Otherwise, you may need to specify a desired option for the subpartition or allow the subpartition to use the default settings for the option.
- Subpartition names must be unique across each partition. Subpartition names do not need to be unique across the entire table.

For example, the following `CREATE TABLE` statement is valid:

```
CREATE TABLE ts (id INT, purchased DATE, PRIMARY KEY (id,
purchased) )
PARTITION BY RANGE(YEAR(purchased))
SUBPARTITION BY HASH(TO_DAYS(purchased))
(
  PARTITION p0 VALUES LESS THAN (1990)
  (
    SUBPARTITION s0,
    SUBPARTITION s1
  ),
  PARTITION p1 VALUES LESS THAN (2000)
  (
    SUBPARTITION s0,
    SUBPARTITION s1
  ),
  PARTITION p2 VALUES LESS THAN MAXVALUE
  (
    SUBPARTITION s0,
    SUBPARTITION s1
  )
);
```

18.5.3.6. Handle NULL values

Handle NULL as zero

If a column value or the value of a user-defined expression is NULL, NULL must be handled to implement partitioning. In most cases, NULL is handled as zero.

If you do not want to use this method of handling NULL, you can implement the NOT NULL constraint when you create tables. To implement the constraint, we recommend that you declare `NOT NULL` for the columns when you create the tables.

 **Note** The NULL value indicates that the calculation result of an expression is NULL. NULL is handled as zero.

Examples

Assume that you insert a row that contains NULL values into a table that is partitioned by range. In the row, NULL is the column value that determines the partition. In this scenario, the system handles NULL as zero and assigns the row to the partition whose value range contains zero.

The tnrage and tnhash tables are used in the examples in this topic. In the following example, the tnrage table is created and a row is inserted into the table:

```
mysql> CREATE TABLE tnrage (
  -> id INT,
  -> name VARCHAR(5)
  -> )
  -> PARTITION BY RANGE(id) (
  -> PARTITION p1 VALUES LESS THAN (1),
  -> PARTITION p2 VALUES LESS THAN MAXVALUE
  -> );
mysql> INSERT INTO tnrage VALUES (NULL, 'jim');
mysql> SELECT * FROM tnrage;
+-----+-----+
| id | name |
+-----+-----+
| NULL | jim |
+-----+-----+
1 row in set (0.00 sec)
```

In the tnrage table, the `NOT NULL` constraint is not declared for the id column. Therefore, the id column can contain NULL values. In the following example, the ALTER TABLE statement is executed to delete the partition that contains NULL values. The SELECT statement is executed to check whether the rows that contain NULL values are stored in the p1 partition of the table.

```
mysql> ALTER TABLE tnrage DROP PARTITION p1;
Query OK, 0 rows affected (0.16 sec)

mysql> SELECT * FROM tnrage;
Empty set (0.00 sec)
```

If you use hash partitioning, an expression that returns NULL is handled as an expression that returns zero. To verify this rule, you can create a table that is partitioned based on hashing and insert a row that contains NULL into the table. Then, you can check whether the row is inserted into the table as expected. For example, execute the following statement to create the tnhash table in the test database:

```
CREATE TABLE tnhash (
  id INT,
  name VARCHAR(5)
)
PARTITION BY HASH(id)
PARTITIONS 2;
```

Execute the INSERT INTO statement to insert a row into the tnhash table. The value in the id column for this row is NULL. Then, execute the SELECT statement to check whether the row is inserted as expected.

```
mysql> INSERT INTO tnhash VALUES (NULL, 'sam');
Query OK, 1 row affected (0.00 sec)
mysql> SELECT * FROM tnhash;
+-----+-----+
| id | name |
+-----+-----+
| NULL | sam |
+-----+-----+
1 row in set (0.01 sec)
```

For an integer that is represented by N, the value of `NULL MOD N` is always NULL. The system handles NULL as zero when the data is assigned to partitions. This example assumes that the Bourne Again Shell (Bash) is used to check whether the row that contains NULL is inserted to the partition as expected. Therefore, you can go back to the system shell to perform the check. In the Bash shell, execute the SELECT statement to query the data that is stored in the p0 partition of the tnhash table. This allows you to check whether the row is inserted into the first partition whose default name is p0.

```
mysql> SELECT * FROM tnhash partition(p0);
+-----+-----+
| id | name |
+-----+-----+
| NULL | sam |
+-----+-----+
1 row in set (0.00 sec)
```

18.5.4. Data types

18.5.4.1. Overview

ApsaraDB for OceanBase supports three categories of data types: numeric types, string types, and date and time types. [Supported data types](#) describes the data types that ApsaraDB for OceanBase supports.

Supported data types

Category	Data type
Numeric types	<ul style="list-style-type: none"> TINYINT BOOL BOOLEAN SMALLINT MEDIUMINT INT INTEGER BIGINT FLOAT DOUBLE DECIMAL <div style="border: 1px solid #add8e6; padding: 5px; margin: 10px 0;"> <p> Note The precision for this data type is inconsistent with that for the DECIMAL data type in MySQL.</p> </div> <ul style="list-style-type: none"> DEC NUMERIC BIT
String types	<ul style="list-style-type: none"> CHAR VARCHAR BINARY VARBINARY ENUM SET
Date and time types	<ul style="list-style-type: none"> DATE DATETIME TIMESTAMP TIME YEAR

18.5.4.2. Numeric types

The numeric types are divided into the following groups:

- Integers

[Integer data types](#) describes the integer data types.

Integer data types

Data type	Storage size (bytes)	Range (signed or unsigned)	Minimum value (signed or unsigned)	Maximum value (signed or unsigned)
TINYINT	1	2^7-1	-128	127
		2^8-1	0	255
		$2^{15}-1$	-32768	32767

SMALLINT Data type	Storage size (bytes)	Range (signed or unsigned)	Minimum value (signed or unsigned)	Maximum value (signed or unsigned)
		$2^{16}-1$	0	65535
MEDIUMINT	3	$2^{23}-1$	-8388608	8388607
		$2^{24}-1$	0	16777215
INT	4	$2^{31}-1$	-2147483648	2147483647
		$2^{32}-1$	0	4294967295
BIGINT	8	$2^{63}-1$	-9223372036854775808	9223372036854775807
		$2^{64}-1$	0	18446744073709551615

- Floating-point numbers or decimal numbers

Floating-point data types describes the floating-point data types.

Floating-point data types

Data type	Storage size (bytes)	Description
FLOAT	4	This data type stores single-precision floating-point numbers.
FLOAT(<i>p</i>)	If $0 \leq p \leq 24$, the storage size is 4 bytes. If $25 \leq p \leq 53$, the storage size is 8 bytes.	If the precision ranges from 0 to 24 digits, the column type is FLOAT and each value in the FLOAT column is a single-precision floating-point number that occupies 4 bytes. If the precision ranges from 25 to 53 digits, the column type is DOUBLE and each value in the DOUBLE column is a double-precision floating-point number that occupies 8 bytes.
DOUBLE [PRECISION]	8	This data type stores double-precision floating-point numbers.
DECIMAL (M,D) NUMERIC (M,D)	Variable length	This data type stores fixed-point numbers.

Other types

Data type	Storage size (bytes)	Description
BIT(M)	The occupied bytes are calculated based on the formula: Number of occupied bytes = $(M + 7)/8$. The default value of M is 1 and the valid value range for M is 1 to 64.	This data type stores binary values.

Note

- You can specify the display width for the values of TINYINT, SMALLINT, MEDIUMINT, INT, and BIGINT data types. The format is data type(M), such as INT(20). In the format, M represents the maximum display width. The maximum valid display width is 255 bytes.

The display width is irrelevant to the storage size or the value range of each data type.

- The UNSIGNED modifier indicates that only positive values can be stored in the specified field.
- The ZEROFILL modifier indicates that output values are padded with 0 instead of spaces. If you specify the ZEROFILL attribute for a numeric column, the system automatically adds the UNSIGNED attribute to the column.
- SERIAL is an alias for `BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE`.

In the definition of integer columns, `SERIAL DEFAULT VALUE` is an alias for `NOT NULL AUTO_INCREMENT UNIQUE`.

TINYINT

```
TINYINT[(M)] [UNSIGNED] [ZEROFILL]
```

The TINYINT data type stores small one-byte integers.

The range of signed numbers for the data type is -128 to 127. The range of unsigned numbers for the data type is 0 to 255.

BOOL or BOOLEAN

BOOL or BOOLEAN is a synonym of TINYINT(1).

Zero values are handled as false, and non-zero values are handled as true.

SMALLINT

```
SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
```

The SMALLINT data type stores small two-byte integers.

The range of signed numbers for the data type is -32768 to 32767. The range of unsigned numbers for the data type is 0 to 65535.

MEDIUMINT

```
MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]
```

The MEDIUMINT data type stores medium-sized integers.

The range of signed numbers for the data type is -8388608 to 8388607. The range of unsigned numbers for the data type is 0 to 16777215.

INT

```
INT[(M)] [UNSIGNED] [ZEROFILL]
```

The INT data type stores normal-size integers.

The range of signed numbers for the data type is -2147483648 to 2147483647. The range of unsigned numbers for the data type is 0 to 4294967295.

Before you insert an invalid integer value into a table, the system automatically converts the value to zero.

INTEGER

```
INTEGER[(M)] [UNSIGNED] [ZEROFILL]
```

INTEGER is a synonym for INT.

BIGINT

```
BIGINT[(M)] [UNSIGNED] [ZEROFILL]
```

The BIGINT data type stores large integers.

The range of signed numbers for the data type is -9223372036854775808 to 9223372036854775807. The range of unsigned numbers for the data type is 0 to 18446744073709551615.

Signed BIGINT or DOUBLE values are used for all the arithmetic operations. Therefore, you cannot use the unsigned large integers that are larger than 9223372036854775807 (63 bits) in arithmetic operations except bit functions. If you use the unsigned large integers that are larger than 9223372036854775807, the last few digits in the result may be inaccurate. This is because rounding is implemented when you convert the values of the BIGINT data type into the values of the DOUBLE data type.

The system handles the values of the BIGINT data type in the following scenarios:

- A column of the BIGINT data type stores large unsigned integers.
- In MIN(col_name) or MAX(col_name), col_name specifies a BIGINT column.
- Operators such as plus signs (+), minus signs (-), and asterisks (*) are used and both operands are integers.
- Strings can be used to store exact integer values in columns of the BIGINT type. In this scenario, strings are converted into numbers and no intermediate double-precision numbers are generated during the conversion.
- If both operands of each plus sign (+), minus sign (-), or asterisk (*) operator are integers, the corresponding operator performs BIGINT operations. In this scenario, if you multiply two large integers that are returned by functions and the product is greater than 9223372036854775807, unexpected results are returned.

BIT

```
BIT[(M)]
```

The BIT data type stores binary values. *M* specifies the number of bits that can be stored. The valid value ranges from 1 to 64.

For example, if you specify *M* as 4, the binary values that can be stored range from 0000 to 1111. You can insert data to a column of the BIT data type by using `b'value'` or `0bvalue`, such as `b'1001'` or `0b1001`.

Note In the contexts of numeric values, the BIT data type stores `numeric values`. In the contexts of strings, the BIT data type stores `string values`. The `numeric values` are the numbers that represent the binary values. The `string values` are the strings that represent the binary values. When you assign values of a BIT column to variables, the assigned values are `numeric values`.

FLOAT

```
FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]
```

The FLOAT data type stores small single-precision floating-point numbers.

The valid value range is -2^{128} to $+2^{128}$. The valid value range can also be expressed as $-3.402823466E+38$ to $-1.175494351E-38$, 0 , and $1.175494351E-38$ to $3.402823466E+38$. This valid range specifies the theoretical limits based on the IEEE standard. The actual range may be smaller than the valid range because different hardware devices or operating systems are used.

M is the total number of digits and D is the number of digits that follow the decimal point. If M and D are not specified, values are stored based on the hardware limits. Each single-precision floating-point number is accurate to about seven decimal places.

Negative values are disallowed if the **UNSIGNED** attribute is specified.

 **Note** If you use the values of the **FLOAT** data type for calculations, unexpected errors may occur. To prevent these errors, the values of the **FLOAT** data type are converted into the values of the **DOUBLE** data type before calculations are performed.

DOUBLE

```
DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]
```

The **DOUBLE** data type stores normal-sized and double-precision floating-point numbers.

The valid value range is -2^{1024} to $+2^{1024}$. The valid value range can also be expressed as $-1.7976931348623157E+308$ to $-2.2250738585072014E-308$, 0 , and $2.2250738585072014E-308$ to $1.7976931348623157E+308$. This valid range specifies the theoretical limits based on the IEEE standard. The actual range may be smaller than the valid range because different hardware devices or operating systems are used.

M is the total number of digits and D is the number of digits that follow the decimal point. If M and D are not specified, values are stored based on the hardware limits. Each double-precision floating-point number is accurate to about 15 decimal places.

Negative values are disallowed if the **UNSIGNED** attribute is specified.

DOUBLE PRECISION

```
DOUBLE PRECISION [(M,D)] [UNSIGNED] [ZEROFILL], REAL[(M,D)] [UNSIGNED] [ZEROFILL]
```

DOUBLE PRECISION is a synonym of **DOUBLE**.

FLOAT(p)

```
FLOAT(p) [UNSIGNED] [ZEROFILL]
```

The **FLOAT(p)** data type stores floating-point numbers. p specifies the precision of a numeric value. The precision is represented by the number of digits in the numeric value. The value of the p parameter is used to only determine whether the data type of an output column is **FLOAT** or **DOUBLE**.

If the value of p ranges from 0 to 24, the data type is **FLOAT** and M or D values do not need to be specified. If the value of p ranges from 25 to 53, the data type is **DOUBLE** and M or D values do not need to be specified. The value range of the output column is the same as the value range for the **FLOAT** or **DOUBLE** data type that is described in this topic. The **FLOAT** data type stores single-precision floating-point numbers. The **DOUBLE** data type stores double-precision floating-point numbers.

DECIMAL (different from the DECIMAL type in MySQL)

```
DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
```

The DECIMAL data type stores packed exact fixed-point numbers. *M* is the total number of digits and specifies the precision of a numeric value. *D* is the number of digits that follow the decimal point and specifies the scale of the numeric value. The number of decimal points and negative number signs (-) is excluded from the *M* value. If the value of *D* is 0, values does not have decimal points or fractional parts. For the DECIMAL data type in ApsaraDB for OceanBase, the maximum number of digits that is specified by *M* is 38. In MySQL, the maximum number of digits is 65. *D* specifies the number of digits that follow the decimal point. The maximum number of digits that follow the decimal point for decimal numeric values is 30. If *D* is not specified, the default value 0 is used. If *M* is not specified, the default value 10 is used.

Negative values are disallowed if the UNSIGNED attribute is specified.

For DECIMAL columns in ApsaraDB for OceanBase, basic operations are performed based on a precision of 38 digits. The basic operations are addition (+), subtraction (-), multiplication (*), and division (/). In MySQL, the basic operations are performed based on a precision of 65 digits.

`DEC[(M,D)] [UNSIGNED] [ZEROFILL]`, `NUMERIC[(M,D)] [UNSIGNED] [ZEROFILL]` are synonyms of DECIMAL.

NUMERIC

`NUMERIC[(M,D)] [UNSIGNED] [ZEROFILL]` is a synonym of DECIMAL.

18.5.4.3. String types

String types describes the string types that ApsaraDB for OceanBase supports.

String types

String type	Number of bytes	Description
CHAR	0-255	<p>The length of a CHAR column is the fixed length that you specify when you create your table. The value length ranges from 0 to 255 bytes.</p> <p>When CHAR values are stored, spaces are padded to the right of each value until the specified length is reached.</p> <p>When CHAR values are retrieved, the trailing spaces of the values are removed.</p> <p>When CHAR values are stored or retrieved, uppercase and lowercase conversion is not performed.</p>
VARCHAR	0-262143	<p>The maximum length of a VARCHAR value is determined by the maximum length for the row values of the VARCHAR data type and the character set that you use.</p> <p>In ApsaraDB for OceanBase, the maximum valid length is 256 KB. In MySQL, the maximum value length is 64 KB.</p>
BINARY	0-255	<p>This data type is similar to the CHAR data type. The difference is that the columns of the BINARY data type store binary strings instead of non-binary strings.</p>

String type	Number of bytes	Description
VARBINARY	0-262143	This data type is similar to the VARCHAR data type. The difference is that the columns of the VARBINARY data type store binary strings instead of non-binary strings.

CHAR

```
CHAR[(M)] [CHARACTER SET charsetname] [COLLATE collationname]
```

CHAR is short for CHARACTER.

The length of a CHAR column is the fixed length that you specify when you create your table. The value length ranges from 0 to 255 bytes. When CHAR values are stored, spaces are padded to the right of each value until the specified length is reached.

VARCHAR

```
VARCHAR(M) [CHARACTER SET charsetname] [COLLATE collationname]
```

This data type stores variable-length strings.

M specifies the maximum value length of a column. The valid value range of *M* is 0 to 256 KB. The actual length that is required to store each value of a VARCHAR field is determined by the length of the real-time field value and the specific character set. For example, you can specify the maximum valid length as 256 KB characters.

If the system uses the UTF8MB4 character set, each character occupies 4 bytes and the maximum valid length is calculated based on the formula: 256 KB × 4 = 1 MB.

VARCHAR is short for CHAR VARYING.

When VARCHAR values are stored, the values are not padded with characters.

If the length of a value that is assigned to a CHAR or VARCHAR column exceeds the maximum length for the column, the value is truncated to meet the length requirement. If spaces are truncated, a warning message is returned. If non-space characters are truncated, an error message instead of a warning message is reported. You can enable the strict SQL mode to prevent the corresponding values from being inserted into the CHAR or the VARCHAR column.

BINARY, VARBINARY

BINARY and VARBINARY data types are similar to CHAR and VARCHAR data types. The difference is that the columns of BINARY and VARBINARY types store binary strings instead of non-binary strings. This means that the columns of the BINARY and VARBINARY types store byte strings instead of character strings. The columns of BINARY and VARBINARY types do not use character sets. The values in these columns are compared and sorted based on the numeric values that are converted from the byte values in the columns.

The maximum value length for a column of the BINARY or VARBINARY data type is the same as that for a column of the CHAR or VARCHAR data type. The maximum value length for a column of the BINARY or VARBINARY type is measured in bytes instead of characters.

18.5.4.4. Date and time data types

[Date and time data types](#) describes the date and time data types that are supported by ApsaraDB for OceanBase SQL.

Date and time data types

Date and time data type	Format	Value range	Size (bytes)
DATE	YYYY-MM-DD	'1000-01-01' to '9999-12-31'	3
DATETIME	YYYY-MM-DD HH:MM:SS	'1000-01-01 00:00:00' to '9999-12-31 23:59:59'	8
TIMESTAMP	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:00' to '2037-12-31 23:59:59'	8
TIME	HH:MM:SS	'-838:59:59' to '838:59:59'	3
YEAR	YYYY (default format)	'1901 to 2155' and '0000'	1
	YY	'70 to 69'. This value range represents years 1970 to 2069.	

By default, the precision for the values of DATETIME, TIMESTAMP, and TIME data types is seconds in date and time functions. You can set the fsp parameter to specify the precision for the fractional second part.

You can specify the precision based on your storage requirements. The fractional second part occupies 0 to 3 bytes based on the specified precision. The most fine-grained time granularity for the fractional second part is microseconds. If you use the microsecond precision, the fractional second part occupies 3 bytes.

The syntax is `type_name(fsp)`.

The `type_name` parameter specifies the data type, such as DATETIME, TIMESTAMP, and TIME.

The `fsp` parameter specifies the precision of the fractional second part. The values of this parameter range from 0 to 6. The default value is 0. The largest value is 6. The value 6 indicates that the precision of the fractional second part is microseconds.

DATE

The DATE data type stores values that consist of only the date part. The supported value range is `1000-01-01` to `9999-12-31`.

ApsaraDB for OceanBase uses the `YYYY-MM-DD` format for DATE values. ApsaraDB for OceanBase allows you to use strings or numbers to assign values to DATE columns.

DATETIME

`DATETIME[(fsp)]`

The DATETIME data type stores values that consist of a date part and a time part. The supported value range is `1000-01-01 00:00:00.000000` to `9999-12-31 23:59:59.000000`. ApsaraDB for OceanBase uses the `YYYY-MM-DD HH:MM:SS[.fraction]` format for DATETIME values. ApsaraDB for OceanBase allows you to use strings or numbers to assign values to DATETIME columns.

The `fsp` parameter specifies the precision of the fractional second part. The values of this parameter range from 0 to 6. The default value is 0. The largest value is 6. The value 6 indicates that the precision of the fractional second part is microseconds.

DATETIME and TIMESTAMP data types support `DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` clauses.

TIMESTAMP

`TIMESTAMP[(fsp)]`

The `TIMESTAMP` data type in ApsaraDB for OceanBase is inconsistent with that in MySQL. This is because the format of the `TIMESTAMP` data type in ApsaraDB for OceanBase complies with strict mode requirements.

Invalid values such as `0000-00-00 00:00:00` are disallowed in ApsaraDB for OceanBase.

The following string values are invalid:

```
'2012^12^32'
'20070523'
'070523'
'071332'
```

The following integer values are invalid:

```
19830905
830905
```

`DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` clauses

If you use `DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` clauses, the `TIMESTAMP` or `DATETIME` column uses the current timestamp as the default value. The timestamp in the column is automatically updated to the current timestamp.

If you specify the value in the first `TIMESTAMP` column in a table as the `default value`, the `DEFAULT` clause cannot be ignored. You can specify the `current timestamp` or a constant date and time value as the default value.

- For the first `TIMESTAMP` column in a table, the `DEFAULT NULL` clause is the same as the `DEFAULT CURRENT_TIMESTAMP` clause. For the other `TIMESTAMP` columns in the table, the `DEFAULT NULL` clause is considered as `DEFAULT 0`.
- In the `CREATE TABLE` statement, you can use the following methods to declare the first `TIMESTAMP` column:
 - If you use `DEFAULT CURRENT_TIMESTAMP` and `ON UPDATE CURRENT_TIMESTAMP` clauses, the column uses the current timestamp as the default value. The timestamp in the column is automatically updated to the current timestamp.
 - If you use the `DEFAULT CURRENT_TIMESTAMP` clause and do not use the `ON UPDATE` clause, the column uses the current timestamp as the default value. In this scenario, the timestamp in the column is not automatically updated to the current timestamp.
 - If you do not use the `DEFAULT` clause and use the `ON UPDATE CURRENT_TIMESTAMP` clause, the default value of the column is 0. The timestamp in the column is automatically updated to the current timestamp.
 - If you specify a constant in the `DEFAULT` clause, the column uses the specified constant as the default value. If the `ON UPDATE CURRENT_TIMESTAMP` clause is specified for the column, the timestamp of the column is automatically updated to the current timestamp. Otherwise, automatic updates are not performed.

In other words, you can use the current timestamp as either, neither, or both of the default value and the auto-update value.

For example, you can use `ON UPDATE` to enable automatic updates of timestamps and avoid automatic initialization for the column.

- In `DEFAULT` and `ON UPDATE` clauses, you can use `CURRENT_TIMESTAMP`, `CURRENT_TIMESTAMP()`, or `NOW()`. These functions are equivalent.

The order of the `DEFAULT` and `ON UPDATE` clauses in the statement does not affect the returned results. If the `DEFAULT` clause and the `ON UPDATE` clause are specified for a `TIMESTAMP` column at the same time, either of the clauses can occur first.

In the following examples, the statements are equivalent:

```
CREATE TABLE t (ts TIMESTAMP);

CREATE TABLE t (ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP);

CREATE TABLE t (ts TIMESTAMP ON UPDATE CURRENT_TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

- Assume that you need to specify a default value or enable automatic updates of timestamps for all the `TIMESTAMP` columns instead of only the first `TIMESTAMP` column. In this case, you must specify a constant in the `DEFAULT` clause for the first `TIMESTAMP` column to disable automatic initialization and updates. For example, you can specify `DEFAULT 0` or `DEFAULT '2003-01-01 00:00:00'`. The rules for the other `TIMESTAMP` columns are the same as those for the first `TIMESTAMP` column, except that `DEFAULT` and `ON UPDATE` clauses cannot be ignored for the other columns. If the `DEFAULT` and `ON` clauses are ignored, automatic initialization and updates cannot apply to the timestamps.

In the following examples, the statements are equivalent:

```
CREATE TABLE t (
  ts1 TIMESTAMP DEFAULT 0,
  ts2 TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP);

CREATE TABLE t (
  ts1 TIMESTAMP DEFAULT 0,
  ts2 TIMESTAMP ON UPDATE CURRENT_TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

NULL values in `TIMESTAMP` columns

You can specify the `NULL` attribute for `TIMESTAMP` columns so that the columns can store `NULL` values.

Examples

```

CREATE TABLE tstest
(
  ts1 TIMESTAMP NULL DEFAULT NULL,
  ts2 TIMESTAMP NULL DEFAULT 0,
  ts3 TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP
);
INSERT INTO tstest values(NULL, NULL, NULL);
SELECT * FROM tstest;
//Result
mysql> SELECT * FROM tstest;
+-----+-----+-----+
| ts1 | ts2 | ts3 |
+-----+-----+-----+
| NULL | NULL | NULL |
+-----+-----+-----+
1 row in set (0.00 sec)

```

TIME

TIME[(fsp)]

The TIME data type stores values that consist of only the time part. The supported value range is - 838:59:59.000000 to 838:59:59.000000. ApsaraDB for OceanBase uses the HH:MM:SS[.fraction] format for TIME values. ApsaraDB for OceanBase allows you to use strings or numbers to assign values to TIME columns.

The fsp parameter specifies the precision of the fractional second part. The values of this parameter range from 0 to 6. The default value is 0. The largest value is 6. The value 6 indicates that the precision of the fractional second part is microseconds.

D TIME values can be strings in the D HH:MM:SS[.fraction] format. You can use the following non-strict SQL syntax: H:MM:SS[.fraction], HH:MM:SS, HH:MM, D HH:MM:SS, D HH:MM, D HH, or SS. D represents days. The valid values range from 0 to 34.

TIME values can be strings in the HHMMSS format. In the format, no delimiters are used. The system assumes that the strings specify valid time. For example, '101112' is valid and represents '10:11:12'. However, '109712' is invalid because the minute part 97 is invalid. The system changes the invalid string to '00:00:00'.

TIME values can be numeric values in the HHMMSS format. The system assumes that the numeric values specify valid time. For example, 101112 represents '10:11:12'. The system also supports numeric values in the following formats: SS, MMSS, HHMMSS, and HHMMSS.fraction.

TIME values can be the results that are returned by functions, such as CURRENT_TIME. The results must comply with the TIME value requirements.

Assume that a TIME value is a string where the hour, minute, and second parts are separated with colons (:). If the value for one of the parts is smaller than 10, you do not need to specify a two-digit value for the part. For example, '8:3:2' and '08:03:02' are equivalent.

When you assign an abbreviated value to a TIME column, you must pay attention to the impacts of colons (:). If a TIME value does not contain colons (:), the system identifies the two rightmost digits in the TIME value as the second part. In this case, the system does not interpret the TIME value as the time on the current day. For example, you may interpret '1112' and 1112 as 11:12:00 : 12 minutes past 11. However, the system interprets '1112' and 1112 as 00:11:12 : 11 minutes and 12 seconds. Similarly, the system interprets '12' and 12 as 00:00:12 . If a TIME value contains colons (:), the system interprets the TIME value as the time on the current day. For example, the system interprets '11:12' as '11:12:00' instead of '00:11:12' .

If valid values fall out of the TIME value range, the values are converted into the closest boundaries of the value range. For example, '-850:00:00' is converted into '-838:59:59' and '850:00:00' is converted into '838:59:59' .

Invalid TIME values are converted into '00:00:00' . Note that '00:00:00' is a valid TIME value. Based on only the stored values in the table, you cannot distinguish the original '00:00:00' values from the '00:00:00' values that are converted from invalid values.

YEAR

The YEAR data type stores two-digit or four-digit values that represent years. By default, YEAR values use the four-digit format. In the four-digit format, valid YEAR values consist of 0000 and values that range from 1901 to 2155. In the two-digit format, valid YEAR values range from 70 to 69. The values represent years 1970 to 2069. YEAR values are displayed in the YYYY format. You can use strings or numbers to assign values to YEAR columns.

You can specify YEAR values in the following formats:

- Four-digit strings: The value range is '1901' to '2155' .
- Four-digit numbers: The value range is 1901 to 2155.
- Two-digit strings: The value range is '00' to '99' . The values in the range of '00' to '69' are converted into years 2000 to 2069. The values in the range of '70' to '99' are converted into years 1970 to 1999.
- Two-digit integers: The value range is 1 to 99. The values in the range of 1 to 69 are converted into years 2001 to 2069. The values in the range of 70 to 99 are converted into years 1970 to 1999. The difference between the range of two-digit integers and that of two-digit strings is that the range of two-digit integers does not include 0. This is because 0 cannot be specified as a number or interpreted as 2000. To use 0 to represent the year 2000, you must specify 0 as the string '0' or '00' or enable the system to interpret 0 as 0000.
- Results that are returned by functions such as NOW(): The returned results must comply with the YEAR value requirements.

Invalid YEAR values are converted into 0000.

For DATETIME, DATE, TIMESTAMP, and YEAR data types, ApsaraDB for OceanBase use the following rules to interpret the dates that have ambiguous YEAR values:

- YEAR values in the range of 00 to 69 are converted into years 2000 to 2069.
- YEAR values in the range of 70 to 99 are converted into years 1970 to 1999.

You can use the ORDER BY clause to sort the two-digit YEAR values or the TIMESTAMP values that contain two-digit YEAR values.

You can use some functions such as MIN() and MAX() to convert TIMESTAMP values or YEAR values into numbers. Two-digit YEAR values are not applicable to these functions. In this case, you can convert the year parts of the TIMESTAMP values or the YEAR values into four-digit values. You can also use MIN(DATE_ADD(TIMESTAMP, INTERVAL 0 DAYS)) .

18.5.5. Character sets

A character set is a set of symbols and encodings. A collation is a set of rules that are used to compare characters in a character set.

Supported character sets

ApsaraDB for OceanBase supports the UTF8MB4 character set. UTF8MB4 is a superset of UTF8 and uses a maximum of four bytes for each character.

Compared with UTF8, UTF8MB4 supports new characters in the iOS operating system, such as emojis. Characters that are supported by UTF8 are known as Basic Multilingual Plane (BMP) characters. The new characters that are supported by UTF8MB4 are known as supplementary characters.

You can specify character sets at different levels: tenant, database, table, field, and session. ApsaraDB for OceanBase supports only the UTF8MB4 character set. By default, the UTF8MB4 character set is used. In most cases, you do not need to specify the character set.

 **Note** In ApsaraDB for OceanBase, the UTF8MB4 character set allows you to use UTF8. UTF8 is an alias for UTF8MB3.

Collation

In ApsaraDB for OceanBase, the supported collations for the UTF8MB4 character set are utf8mb4_bin and utf8mb4_general_ci. The default collation is utf8mb4_general_ci.

One of the main differences of the two collations is that they have different impacts on sorting orders and string comparisons. The utf8mb4_bin collation is case-sensitive and the utf8mb4_general_ci collation is not case-sensitive.

18.5.6. Auto-increment fields

18.5.6.1. Overview

This topic describes the AUTO_INCREMENT attribute of ApsaraDB for OceanBase.

ApsaraDB for OceanBase allows you to specify the AUTO_INCREMENT attribute for an integer field in a table. In this aspect, ApsaraDB for OceanBase is the same as MySQL. ApsaraDB for OceanBase can automatically generate unique values for auto-increment fields.

The following list describes the notes on the AUTO_INCREMENT attribute in ApsaraDB for OceanBase:

- You can specify the AUTO_INCREMENT attribute for fields of various integer data types, such as TINYINT, SMALLINT, MEDIUMINT, INT, INTEGER, BIGINT, FLOAT, and DOUBLE. You cannot specify the AUTO_INCREMENT attribute for fields of the other data types.

 **Note** In MySQL, you can specify the AUTO_INCREMENT attribute for fields of FLOAT, DOUBLE, and BOOLEAN data types. You cannot specify this attribute for fields of DECIMAL and BIT data types.

- You can specify the AUTO_INCREMENT attribute for only one field in a table.

```
mysql> create table t1 (id int auto_increment);
ERROR 1075 (42000): Incorrect table definition; there can be only one auto column and it must be defined as a key
#In MySQL, auto-increment fields must have indexes that can be primary key indexes or general indexes. Otherwise, the preceding error occurs. Indexes are not required for auto-increment fields in ApsaraDB for OceanBase. This is the difference between ApsaraDB for OceanBase and MySQL.
```

```
mysql> create table t1 (id int auto_increment,name VARCHAR(20) primary key,key(id));
Query OK, 0 rows affected (0.01 sec)
#The AUTO_INCREMENT attribute is specified for the id field. The id field is a general field that has an index. Indexes are not required for auto-increment fields in ApsaraDB for OceanBase. This is the difference between ApsaraDB for OceanBase and MySQL.
```

- In other cases, the following rules take effect:
 - i. If you insert a NULL value into an `AUTO_INCREMENT` column, MySQL automatically generates the next sequence number for the column.
 - ii. When you insert a row and do not explicitly specify a value for the `AUTO_INCREMENT` column, the system considers the value of the column as NULL. MySQL automatically generates the next sequence number for the column.
 - iii. In all the modes except `NO_AUTO_VALUE_ON_ZERO`, if you insert 0 into an `AUTO_INCREMENT` column, the system considers the value of the column as NULL. MySQL automatically generates the next sequence number for this column.

```
mysql> insert into t1 (id,name) values (null,'test');
Query OK, 1 row affected (0.00 sec)
mysql> select * from t1;
+----+-----+
| id | name |
+----+-----+
| 1 | test |
+----+-----+
1 row in set (0.00 sec)
```

The preceding statements are equivalent to the following statements:

```
mysql> insert into t1 (name) values ('test');

## Create a table.
mysql> CREATE TABLE t1 (id int AUTO_INCREMENT,PRIMARY KEY (id));

## Insert data into the table.
mysql> insert into t1 values (null),(null),(null);
Query OK, 3 rows affected (0.00 sec)

## Insert the value 7 into the table.
mysql> insert into t1 values (7);
Query OK, 1 row affected (0.00 sec)

## The value of the auto-increment column changes to 8.
mysql> show create table t1\G;
***** 1. row *****
      Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8
```

18.5.6.2. System variables for auto-increment columns

```
mysql> show variables like '%auto_increment%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 1 |
| auto_increment_offset | 1 |
+-----+-----+
```

You can specify global or session values for the preceding two system variables. The `auto_increment_increment` variable specifies the interval between two successive values in an auto-increment column. The `auto_increment_offset` variable specifies the start value of an auto-increment column.

The default value of the `auto_increment_increment` variable is 1. The values of this variable range from 1 to 65535.

The default value of the `auto_increment_offset` variable is 1. The values of this variable range from 1 to 65535.

Examples

- Example 1

```
auto_increment_increment=2
auto_increment_offset=1

mysql>create table t1(id int auto_increment primary key);
Query OK, 0 rows affected (0.00 sec)

mysql> set session auto_increment_increment=2;
Query OK, 0 rows affected (0.00 sec)

mysql> set session auto_increment_offset=1;
Query OK, 0 rows affected (0.00 sec)

mysql> show session variables like '%auto_incre%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| auto_increment_increment | 2   |
| auto_increment_offset  | 1   |
+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into t1 values (null),(null),(null),(null),(null),(null);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from t1;
+----+
| id |
+----+
| 1  |
| 3  |
| 5  |
| 7  |
| 9  |
| 11 |
+----+
6 rows in set (0.00 sec)
```

- Example 2

```
auto_increment_increment=2
auto_increment_offset=2

mysql> truncate t1;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> set session auto_increment_increment=2;
Query OK, 0 rows affected (0.00 sec)

mysql> set session auto_increment_offset=2;
Query OK, 0 rows affected (0.00 sec)

mysql> show session variables like '%auto_incre%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| auto_increment_increment | 2   |
| auto_increment_offset  | 2   |
+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into t1 values (null),(null),(null),(null),(null),(null);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from t1;
+----+
| id |
+----+
| 2 |
| 4 |
| 6 |
| 8 |
| 10 |
| 12 |
+----+
6 rows in set (0.00 sec)
```

- Example 3

```

auto_increment_increment=10
auto_increment_offset=5

mysql> truncate t1;
Query OK, 0 rows affected (0.00 sec)

mysql> set session auto_increment_increment=10;
Query OK, 0 rows affected (0.00 sec)

mysql> set session auto_increment_offset=5;
Query OK, 0 rows affected (0.00 sec)

mysql> show session variables like '%auto_incre%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| auto_increment_increment | 10   |
| auto_increment_offset  | 5    |
+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into t1 values (null),(null),(null),(null),(null),(null);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from t1;
+-----+
| id |
+-----+
| 5 |
| 15 |
| 25 |
| 35 |
| 45 |
| 55 |
+-----+
6 rows in set (0.00 sec)

```

18.5.6.3. Change start values of auto-increment columns

You can specify the start value of an auto-increment column when you execute `CREATE TABLE` to create a table. You can also execute `ALTER TABLE table_name AUTO_INCREMENT=n` to change the start value of an `AUTO_INCREMENT` column. The specified value `n` may be smaller than the current value of the `AUTO_INCREMENT` column. If this occurs, the system does not report an error for the executed statement, but the value `n` does not take effect.

Examples

Examples

```
mysql>select LAST_INSERT_ID();
+-----+
| LAST_INSERT_ID() |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
```

18.5.6.5. Limits

The following list describes the limits of auto-increment fields:

- The values of auto-increment fields must be unique. The values of auto-increment fields may not be continuous. For example, if you execute the `INSERT ...SELECT...` statement to insert data in batches, the values that are generated for an auto-increment field may not be continuous.
- You can execute a single `INSERT` statement on only one partition. If you execute a single `INSERT` statement across partitions, the system reports an error.

18.5.7. Functions

18.5.7.1. Overview

ApsaraDB for OceanBase supports date and time functions, string functions, type conversion functions, aggregate functions, flow control functions, mathematical functions, comparison functions, information functions, and other functions.

Expressions can be used in SQL statements and clauses, such as `ORDER BY` and `HAVING` clauses in `SELECT` statements, `WHERE` clauses in `SELECT`, `DELETE`, and `UPDATE` statements, and `SET` statements. You can define expressions by using literal values, column values, `NULL` values, functions, and operators.

In most cases, an expression that contains a `NULL` value returns a `NULL` value.

Supported functions describes the functions that ApsaraDB for OceanBase supports.

Supported functions

Function type	Function description	Function name
---------------	----------------------	---------------

Function type	Function description	Function name
Date and time functions	Return date and time information.	<ul style="list-style-type: none"> • CURRENT_TIME • CURTIME • CURRENT_TIMESTAMP • CURRENT_DATE • CURDATE • DATE_ADD • DATE_FORMAT • DATE_SUB • EXTRACT • NOW • STR_TO_DATE • TIME_TO_USEC • USEC_TO_TIME • UNIX_TIMESTAMP • DATEDIFF • TIMEDIFF • TIMESTAMPDIFF • PERIOD_DIFF • TO_DAYS • FROM_DAYS
String functions	Manipulate n-base numbers, strings, and expressions. For example, you can use a function of this type to retrieve the start position of a string or the number of returned strings.	<ul style="list-style-type: none"> • CONCAT • SUBSTRING • SUBSTR • TRIM • LENGTH • UPPER • LOWER • HEX • UNHEX • INT2IP • IP2INT • LIKE • REGEXP • REPEAT • SUBSTRING_INDEX • LOCATE • INSTR • REPLACE • FIELD • ELT
Type conversion functions	Convert values from a data type to another data type.	CAST

Function type	Function description	Function name
Aggregate functions	Perform calculations on a set of values and return a single value. In most cases, aggregate functions are used in conjunction with <code>GROUP BY</code> clauses of <code>SELECT</code> statements. If an aggregate function is used in conjunction with a <code>GROUP BY</code> clause, the aggregate functions return a single value for each group instead of the entire table.	<ul style="list-style-type: none"> • AVG • COUNT • MAX • MIN • SUM • GROUP_CONCAT
Mathematical functions	Perform mathematical calculations based on numeric expressions.	<ul style="list-style-type: none"> • ROUND • CEIL • FLOOR • ABS • NEG • SIGN • CONV • MOD • POW • POWER
Comparison functions	Compare input values.	<ul style="list-style-type: none"> • GREATEST • LEAST • ISNULL
Flow control functions	Control flows.	<ul style="list-style-type: none"> • CASE • IF • IFNULL • NULLIF
Information functions	Retrieve dynamic database information.	<ul style="list-style-type: none"> • FOUND_ROWS • LAST_INSERT_ID
Other functions	The other functions.	<ul style="list-style-type: none"> • COALESCE • NVL • DECODE

18.5.7.2. Date and time functions

The functions of this type return date and time information.

CURRENT_TIME() and **CURRENT_TIMESTAMP()**

 **Notice** You can pass a number that ranges from 0 to 6 into a `CURRENT_TIME()` or a `CURRENT_TIMESTAMP()` function. The number specifies the precision of the fractional second part.

The following list describes the notes on `CURRENT_TIME()` and `CURRENT_TIMESTAMP()` functions:

- `CURRENT_TIME()` and `CURRENT_TIMESTAMP()` functions return the current time of the system. The two functions return the time in different formats.

- `CURRENT_TIME()` returns the current time in the `HH:MI:SS` format. The returned value excludes the date part.
- `CURRENT_TIMESTAMP()` returns the current date and time in the `YYYY-MM-DD HH:MI:SS` format.

Examples

```
root@test 04:09:10>SELECT CURRENT_TIME(), CURRENT_TIMESTAMP();
+-----+-----+
| CURRENT_TIME() | CURRENT_TIMESTAMP() |
+-----+-----+
| 16:24:09      | 2014-10-31 16:24:09 |
+-----+-----+
1 row in set (0.00 sec)
root@(none) 09:59:30>select CURRENT_TIME(1);
#You can specify the precision of the fractional second part.
+-----+
| CURRENT_TIME(1) |
+-----+
| 09:59:32.4      |
+-----+
1 row in set (0.00 sec)
root@(none) 09:59:32>select CURRENT_TIME(7);
ERROR 1426 (42000): Too big precision 7 specified for column 'curtime'. Maximum is 6.
#The value that you can specify for the precision ranges from 0 to 6. If the value that you specify is not in this range, the system reports the 1426 (42000) error.
```

 **Note** If `CURRENT_TIME()` or `CURRENT_TIMESTAMP()` is run for different transactions in the same session, the returned results may not be incremental in chronological order. This is because the time of one server is different from that of another server.

CURTIME()

`CURTIME()` is a synonym for `CURRENT_TIME()` or `CURRENT_TIME`.

```
mysql> select curtime(), current_time(), current_time;
+-----+-----+-----+
| curtime() | current_time() | current_time |
+-----+-----+-----+
| 14:45:37 | 14:45:37      | 14:45:37     |
+-----+-----+-----+
1 row in set (0.00 sec)
```

CURRENT_DATE()

`CURRENT_DATE()` returns the current date in the `'YYYY-MM-DD'` or `YYYYMMDD` format.

The returned date format varies based on the date format that is specified in the function. If the date format in the function is a string, the returned date is a string. If the date format in the function is a numeric value, the returned date is a numeric value.

Examples

```
mysql> select current_date, current_date+5;
+-----+-----+
| current_date | current_date+5 |
+-----+-----+
| 2015-08-27 | 20150832 |
+-----+-----+
1 row in set (0.00 sec)
```

CURDATE()

CURDATE() is a synonym for CURRENT_DATE() and CURRENT_DATE .

```
mysql> select curdate(), current_date(), current_date;
+-----+-----+-----+
| curdate() | current_date() | current_date |
+-----+-----+-----+
| 2015-08-27 | 2015-08-27 | 2015-08-27 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

DATE_ADD(date, INTERVAL exprunit)

This function adds a specified interval to a date. The *date* parameter specifies the start date to which an interval is added. The *expr* parameter specifies the interval to be added. The values of *expr* can be negative. Based on the internal configurations and the time zone, the operating system determines whether to use daylight saving time (DST) for the DATE_ADD() function.

The following list describes the notes on the DATE_ADD() function:

- The data types of *date* values must be string or date and time data types, such as DATETIME and TIMESTAMP. The other data types are not supported. If you use string data types, the strings must represent time values.
- The valid format of the *date* parameter is YYYY-MM-DD HH:MM:SS.SSSSSS .

MySQL supports non-strict syntax for parsing date strings. If a string contains digits and non-digit characters, MySQL parses only the digits as time values and assigns the values to the year, month, and day parts in sequence. For example, the Ywwe1990d07 09,12:45-08&900 string represents the same time as 1990-07-09 12:45:08.900 . Strict syntax is applied in ApsaraDB for OceanBase. The system reports errors for invalid date types. For example, if you set the date parameter to the invalid date ABC , the system reports an error.

- You must specify the date part in a *date* string. The time part in a date string is optional. If you do not specify the time part, the default values are used. For example, 1990-07-09 is valid. By default, the time values that follow 1990-07-09 are set to 0 values: 1990-07-09 00:00:00.000000 . 1990-07 and 1990 are invalid date values.
- The DATE_ADD() function cannot parse TIMESTAMP strings, such as 990309 .
- In ApsaraDB for OceanBase, you can use the results of invoking other system functions as the values of *date*.
- ApsaraDB for OceanBase does not support fuzzy match for two-digit years. For example, MySQL interprets the 12 year as the 2012 year, but ApsaraDB for OceanBase interprets the 12 year as the year 12.
- The values of *expr* can be negative. If you specify a negative value for the *expr* parameter, the function subtracts the corresponding interval from the start date. You can use the results of invoking system functions as *expr* values. All the results are processed as strings.

- The *unit* parameter specifies the unit of the interval that is to be added or subtracted. The valid values of the parameter are MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, SECOND_MICROSECOND, MINUTE_MICROSECOND, MINUTE_SECOND, HOUR_MICROSECOND, HOUR_SECOND, HOUR_MINUTE, DAY_MICROSECOND, DAY_SECOND, DAY_MINUTE, DAY_HOUR, and YEAR_MONTH. QUARTER represents quarters.
- If the *unit* value is a compound unit, you must enclose *expr* values in single quotation marks (').

 **Note** In MySQL command-line clients, if a single line is excessively long and compromises read experience, you can add `\G` to the end of the SELECT statement. `\G` is used to align the query results in a vertical way.

Examples

```
mysql> SELECT DATE_ADD(now(), INTERVAL 5 DAY),
-> DATE_ADD('2014-01-10', INTERVAL 5 MICROSECOND),
-> DATE_ADD('2014-01-10', INTERVAL 5 SECOND),
-> DATE_ADD('2014-01-10', INTERVAL 5 MINUTE),
-> DATE_ADD('2014-01-10', INTERVAL 5 HOUR),
-> DATE_ADD('2014-01-10', INTERVAL 5 DAY),
-> DATE_ADD('2014-01-10', INTERVAL 5 WEEK),
-> DATE_ADD('2014-01-10', INTERVAL 5 MONTH),
-> DATE_ADD('2014-01-10', INTERVAL 5 QUARTER),
-> DATE_ADD('2014-01-10', INTERVAL 5 YEAR),
-> DATE_ADD('2014-01-10', INTERVAL '5.000005' SECOND_MICROSECOND),
-> DATE_ADD('2014-01-10', INTERVAL '05:05.000005' MINUTE_MICROSECOND),
-> DATE_ADD('2014-01-10', INTERVAL '05:05' MINUTE_SECOND),
-> DATE_ADD('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND),
-> DATE_ADD('2014-01-10', INTERVAL '05:05:05' HOUR_SECOND),
-> DATE_ADD('2014-01-10', INTERVAL '05:05' HOUR_MINUTE),
-> DATE_ADD('2014-01-10', INTERVAL '01 05:05:05.000005' DAY_MICROSECOND),
-> DATE_ADD('2014-01-10', INTERVAL '01 05:05:05' DAY_SECOND),
-> DATE_ADD('2014-01-10', INTERVAL '01 05:05' DAY_MINUTE),
-> DATE_ADD('2014-01-10', INTERVAL '01 05' DAY_HOUR),
-> DATE_ADD('2014-01-10', INTERVAL '1-01' YEAR_MONTH) \G
***** 1. row *****
      DATE_ADD(now(), INTERVAL 5 DAY): 2016-03-19 13:56:45
      DATE_ADD('2014-01-10', INTERVAL 5 MICROSECOND): 2014-01-10 00:00:00.000005
      DATE_ADD('2014-01-10', INTERVAL 5 SECOND): 2014-01-10 00:00:05
      DATE_ADD('2014-01-10', INTERVAL 5 MINUTE): 2014-01-10 00:05:00
      DATE_ADD('2014-01-10', INTERVAL 5 HOUR): 2014-01-10 05:00:00
      DATE_ADD('2014-01-10', INTERVAL 5 DAY): 2014-01-15
      DATE_ADD('2014-01-10', INTERVAL 5 WEEK): 2014-02-14 00:00:00
      DATE_ADD('2014-01-10', INTERVAL 5 MONTH): 2014-06-10
      DATE_ADD('2014-01-10', INTERVAL 5 QUARTER): 2015-04-10 00:00:00
      DATE_ADD('2014-01-10', INTERVAL 5 YEAR): 2019-01-10
      DATE_ADD('2014-01-10', INTERVAL '5.000005' SECOND_MICROSECOND): 2014-01-10 00:00:05.000005
      DATE_ADD('2014-01-10', INTERVAL '05:05.000005' MINUTE_MICROSECOND): 2014-01-10 00:05:05.000005
      DATE_ADD('2014-01-10', INTERVAL '05:05' MINUTE_SECOND): 2014-01-10 00:05:05
      DATE_ADD('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND): 2014-01-10 05:05:05.000005
      DATE_ADD('2014-01-10', INTERVAL '05:05:05' HOUR_SECOND): 2014-01-10 05:05:05
      DATE_ADD('2014-01-10', INTERVAL '05:05' HOUR_MINUTE): 2014-01-10 05:05:00
      DATE_ADD('2014-01-10', INTERVAL '01 05:05:05.000005' DAY_MICROSECOND): 2014-01-11 05:05:05.000005
      DATE_ADD('2014-01-10', INTERVAL '01 05:05:05' DAY_SECOND): 2014-01-11 05:05:05
      DATE_ADD('2014-01-10', INTERVAL '01 05:05' DAY_MINUTE): 2014-01-11 05:05:00
      DATE_ADD('2014-01-10', INTERVAL '01 05' DAY_HOUR): 2014-01-11 05:00:00
      DATE_ADD('2014-01-10', INTERVAL '1-01' YEAR_MONTH): 2015-02-10
1 row in set (0.01 sec)
```

In the `DATE_ADD()` function, you can place addition (+) or subtraction operators (-) before `INTERVAL`, as shown in the following examples:

```
date + INTERVAL expr unit
date - INTERVAL expr unit
```

For addition operations, `INTERVAL expr unit` can be placed on the right of the *date* parameter. The value of the *date* parameter can be a date value or a date and time value. For subtraction operations, `INTERVAL expr unit` can be placed only on the right of the *date* parameter. If `INTERVAL expr unit` is placed on the left of the *date* parameter, a date value is subtracted from an interval. This computing process does not return a valid result.

Examples

```
mysql> SELECT '2008-12-31 23:59:59' + INTERVAL 1 SECOND , '2005-01-01' - INTERVAL 1 SECOND\G;
***** 1. row *****
'2008-12-31 23:59:59' + INTERVAL 1 SECOND: 2009-01-01 00:00:00
'2005-01-01' - INTERVAL 1 SECOND: 2004-12-31 23:59:59
1 row in set (0.00 sec)
```

DATE_FORMAT(*date*, *format*)

`DATE_FORMAT()` is an inverse function of `STR_TO_DATE()`. `DATE_FORMAT()` converts a specified value of the *date* parameter into a time string based on the specified *format*.

The *date* parameter specifies the time value to be converted. The *date* values can be date and time values of the `STRING` data type. For more information, see the description of the *date* parameter in `DATE_ADD()`.

Values of the *format* parameter in `DATE_FORMAT()` describes the *format* values that can be used in a format string.

Values of the format parameter in DATE_FORMAT()

Format	Description	Returned value range or format
%a	The abbreviation of the day of the week.	Sun..Sat
%b	The abbreviation of the month.	Jan to Dec.
%c	The month of the numeric type.	1 to 12.
%D	The day that ends with an ordinal indicator: st, nd, rd, or th.	1st to 31st.
%d	The day of the numeric type in the month.	01 to 31.
%e	The day of the numeric type in the month.	1 to 31.
%f	The microsecond.	000000 to 999999.
%H	The hour.	00 to 23.
%h	The hour.	01 to 12.
%I	The hour.	01 to 12.
%i	The minute.	00 to 59.

Format	Description	Returned value range or format
%j	The sequence number of the day in the year.	000 to 366.
%k	The hour.	0 to 23.
%l	The hour.	01 to 12.
%M	The name of the month.	January to December.
%m	The month of the numeric type.	01 to 12.
%p	The morning or the afternoon.	AM and PM.
%r	The time in the 12-hour clock.	hh:mm:ss AM/PM
%S	The seconds.	00 to 59.
%s	The seconds.	00 to 59.
%T	The time in the 24-hour clock.	hh:mm:ss
%U	The sequence number of the week in the year when Sunday is the first day of each week.	00 to 53.
%u	The sequence number of the week in the year when Monday is the first day of each week.	00 to 53.
%V	The sequence number of the week in the year when Sunday is the first day of each week. This option is used in conjunction with %X. Notes: A date may fall within the first week of a year or the last week of the previous year. If this occurs, the sequence number of the week varies based on the format values. Wednesday, January 01, 2014 is used as an example to explain this rule. If the format value is %U or %u, the date falls within week 00 in 2014. If the format value is %V or %v, the date falls within week 52 in 2013.	01 to 53.
%v	The sequence number of the week in the year when Monday is the first day of each week. This option is used in conjunction with %x.	01 to 53.
%W	The day of the week.	Sunday to Saturday.
%w	The sequence number of the day in the week.	0 to 6. The value 0 represents Sunday. The value 6 represents Saturday. Similar rules apply to values 1 to 5.

Format	Description	Returned value range or format
%X	The year of the week when Sunday is the first day of each week. The year is represented by a number that consists of four digits. %X is used in conjunction with %V.	-
%x	The year of the week when Monday is the first day of each week. The year is represented by a number that consists of four digits. %x is used in conjunction with %v.	-
%Y	The year in the four-digit format.	-
%y	The year in the two-digit format.	-
%%	The literal character %.	-

Examples

```
mysql> SELECT DATE_FORMAT('2014-01-01', '%Y-%M-%d'),
DATE_FORMAT('2014-01-01', '%X-%V'),DATE_FORMAT('2014-01-01', '%U') \G
***** 1. row *****
DATE_FORMAT('2014-01-01', '%Y-%M-%d'): 2014-January-01
DATE_FORMAT('2014-01-01', '%X-%V'): 2013-52
DATE_FORMAT('2014-01-01', '%U'): 00
1 row in set (0.00 sec)
```

DATE_SUB(*date*, INTERVAL *expr unit*)

This function subtracts a specified interval from a date. The *date* parameter specifies the start date from which an interval is subtracted. The *expr* parameter specifies the interval to be subtracted from the start date. The values of *expr* can be negative. If you specify a negative value for the *expr* parameter, the function adds the corresponding interval to the start date.

For more information about the parameter description, see `DATE_ADD()` .

Examples

```
mysql> SELECT DATE_SUB('2014-01-10', INTERVAL 5 HOUR),
DATE_SUB('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND)\G;
***** 1. row *****
DATE_SUB('2014-01-10', INTERVAL 5 HOUR): 2014-01-09 19:00:00
DATE_SUB('2014-01-10', INTERVAL '05:05:05.000005' HOUR_MICROSECOND): 2014-01-09 18:54:54.999995
1 row in set (0.00 sec)
```

EXTRACT(*unit* FROM *date*)

This function extracts parts from a date based on the specified units. The *date* parameter specifies the date. The *unit* parameter specifies the units.

For more information about the parameter description, see `DATE_ADD()` .

- The data type of the results that the EXTRACT function returns is BIGINT.
- For a single unit, such as microseconds to years, the EXTRACT function returns the corresponding integer.

- If the *unit* value is WEEK, the function returns the sequence number of the week in the year of the date that you specify in the *date* expression. ApsaraDB for OceanBase considers the first Sunday of the year as the beginning of the first week in the year. If the first Sunday of a year is not January 1, January 1 and the days between the first Sunday and January 1 are in week 0. For example, the first Sunday in 2013 is January 6. Therefore, `SELECT EXTRACT(WEEK FROM '2013-01-01')` returns 0 and `SELECT EXTRACT(WEEK FROM '2013-01-06')` returns 1.
- For compound units such as SECOND_MICROSECOND, ApsaraDB for OceanBase combines the values for the specified units and returns a single value. For example, `SELECT EXTRACT(YEAR_MONTH FROM '2012-03-09')` returns 201203 .

```
mysql> SELECT EXTRACT(WEEK FROM '2013-01-01'),
EXTRACT(WEEK FROM '2013-01-06'),
EXTRACT(YEAR_MONTH FROM '2012-03-09'),
EXTRACT(DAY FROM NOW())\G;
***** 1. row *****
EXTRACT(WEEK FROM '2013-01-01'): 0
EXTRACT(WEEK FROM '2013-01-06'): 1
EXTRACT(YEAR_MONTH FROM '2012-03-09'): 201203
EXTRACT(DAY FROM NOW()): 18
1 row in set (0.00 sec)
```

NOW()

This function returns the current system time that is accurate to seconds. The time format is `YYYY-MM-DD HH:MI:SS` .

 **Notice** You can pass a number that ranges from 0 to 6 into a NOW() function. The number specifies the precision of the fractional second part. By default, `NOW()` is equivalent to `NOW(0)` .

Examples

```

mysql> SELECT NOW();
+-----+
| NOW()          |
+-----+
| 2014-02-17 11:46:15|
+-----+
1 row in set (0.00 sec)
root@(none) 09:52:46>select now(0);
+-----+
| now(0)         |
+-----+
| 2014-11-03 09:55:01 |
+-----+
1 row in set (0.00 sec)
root@(none) 09:55:01>select now(1);
+-----+
| now(1)         |
+-----+
| 2014-11-03 09:55:04.2 |
+-----+
1 row in set (0.00 sec)
root@(none) 09:55:04>select now(2);
+-----+
| now(2)         |
+-----+
| 2014-11-03 09:55:06.57 |
+-----+
1 row in set (0.00 sec)
root@(none) 09:55:06>select now(3);
+-----+
| now(3)         |
+-----+
| 2014-11-03 09:55:09.576 |
+-----+
1 row in set (0.00 sec)
root@(none) 09:55:09>select now(7);
ERROR 1426 (42000): Too big precision 7 specified for column 'now'. Maximum is 6.

```

STR_TO_DATE(*str*,*format*)

This function converts a string that is specified by *str* into a value of the DATETIME, DATE, or TIME type based on the *format* string. If the format string contains date and time parts, STR_TO_DATE() returns a DATETIME value. If the format string contains only the date part, this function returns a DATE value. If the format string contains only the time part, this function returns a TIME value.

The format of the DATE, TIME, or DATETIME value in *str* must be specified in *format*. If the *str* value contains an invalid DATE, TIME, or DATETIME value, STR_TO_DATE() returns NULL. In this case, the system returns a warning message for the invalid value.

Values of the format parameter in STR_TO_DATE() describes the *format* values that can be used in a format string.

Values of the format parameter in STR_TO_DATE()

Format	Description	Returned value range or format
%b	The abbreviation of the month.	Jan to Dec
%c	The month of the numeric type.	1 to 12
%D	The day that ends with an ordinal indicator: st, nd, rd, or th.	1st to 31st
%d	The day of the numeric type in the month.	01 to 31
%e	The day of the numeric type in the month.	1 to 31
%f	The microsecond.	000000 to 999999
%H	The hour.	00 to 23
%h	The hour.	01 to 12
%l	The hour.	01 to 12
%i	The minute.	00 to 59
%k	The hour.	0 to 23
%l	Hours.	01 to 12
%M	The name of the month.	January to December
%m	The month of the numeric type.	01 to 12
%p	The morning or the afternoon.	AM and PM
%r	The time in the 12-hour clock.	hh:mm:ss AM/PM
%S	The seconds.	00 to 59
%s	The seconds.	00 to 59
%T	The time in the 24-hour clock.	hh:mm:ss
%Y	The year in the four-digit format.	-

Examples

```
mysql> SELECT STR_TO_DATE('2014-Jan-1st 5:5:5', '%Y-%b-%D');
+-----+
| STR_TO_DATE('2014-Jan-1st 5:5:5 pm', '%Y-%b-%D') |
+-----+
| 2014-01-01 05:05:05          |
+-----+
1 row in set (0.00 sec)
```

TIME_TO_USEC(*date*)

This function converts the internal time of ApsaraDB for OceanBase into the number of microseconds. The returned result of this function indicates the number of microseconds from 1970-01-01 00:00:00 to the time that is specified by *date*. The returned result is the UTC time and uses the UTC+0 time zone.

The following list describes the notes on the TIME_TO_USEC(*date*) function:

- The *date* parameter specifies the date that is to be converted into the number of microseconds. The date uses the time zone that is specified in the system. The value of the date parameter is a string of the `TIMESTAMP` or `TIME` data type.
- The `TIME_TO_USEC` function can use the results of invoking other functions as the values of the date parameter. Note that the results must be strings of the `TIMESTAMP` or `TIME` data type.
- The returned values of the `TIME_TO_USEC` function are measured in microseconds. The data type of the returned values is `INT`.

Examples

```
mysql> SELECT TIME_TO_USEC('2014-03-25'), TIME_TO_USEC(now(6));
+-----+-----+
| TIME_TO_USEC('2014-03-25') | TIME_TO_USEC(now(6)) |
+-----+-----+
|      1395676800000000 | 1395735415207794 |
+-----+-----+
1 row in set (0.00 sec)
```

USEC_TO_TIME(*usec*)

This function is an inverse function of `TIME_TO_USEC(date)`. The `USEC_TO_TIME(usec)` function adds the value of *usec* to 1970-01-01 00:00:00 and returns a result that uses the required time zone. For example, if you invoke the `USEC_TO_TIME(1)` function in the UTC+8 time zone, the function returns 1970-01-01 08:00:01.

The following list describes the notes on the `USEC_TO_TIME(usec)` function:

- The *usec* parameter specifies the number of microseconds.
- This function returns a value of the `TIMESTAMP` data type.

Examples

```
mysql> SELECT USEC_TO_TIME(1);
+-----+
| USEC_TO_TIME(1) |
+-----+
| 1970-01-01 08:00:00.000001 |
+-----+
1 row in set (0.00 sec)
```

UNIX_TIMESTAMP(),UNIX_TIMESTAMP(*date*)

If you do not pass arguments into the `UNIX_TIMESTAMP()` function, the function returns a `UNIX timestamp`. A `UNIX timestamp` is the number of seconds that have elapsed since '1970-01-01 00:00:00' GMT. The returned result is an unsigned integer.

If you pass a *date* value into the UNIX_TIMESTAMP() function, the function returns the value as the number of seconds that have elapsed since '1970-01-01 00:00:00' GMT . The values of the *date* parameter can be DATE strings, DATETIME strings, TIMESTAMP strings, or numbers in the YYMMDD or YYYYMMDD format. Note that the numbers represent the local time.

Examples

```
mysql> SELECT UNIX_TIMESTAMP();
+-----+
| UNIX_TIMESTAMP() |
+-----+
|      1427176668 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT UNIX_TIMESTAMP('1997-10-04 22:23:00')
+-----+
| UNIX_TIMESTAMP('1997-10-04 22:23:00') |
+-----+
|                875974980 |
+-----+
1 row in set (0.00 sec)
```

If the values in the `TIMESTAMP` column are used as the values of the date parameter for the `UNIX_TIMESTAMP` function, the function returns internal timestamps.

If you pass a date that does not fall within the basic date ranges into the UNIX_TIMESTAMP() function, the function returns 0. The function checks the specified date values against only the basic date ranges. In the basic date ranges, year values range from 1970 to 2037, month values range from 01 to 12, and day values range from 01 to 31.

DATEDIFF(*expr1*,*expr2*)

The DATEDIFF() function returns the number of days between the start date that is specified by *expr1* and the end date that is specified by *expr2*. *expr1* and *expr2* are date or date and time expressions. Only the date parts in the expressions are used to calculate the returned result.

You must specify two arguments for this function. If the number of the specified arguments is not two, the system reports an error.

Examples

```
mysql> select datediff('2015-06-19','1994-12-17'), datediff('2015-06-19','1998-06-27 10:10:10'), datediff(now(), '2014-01-02')\G;
***** 1. row *****
      datediff('2015-06-19','1994-12-17'): 7489
      datediff('2015-06-19','1998-06-27 10:10:10'): 6201
      datediff(now(), '2014-01-02'): 533
1 row in set (0.00 sec)
```

TIMEDIFF(*expr1*,*expr2*)

The TIMEDIFF() function returns the time interval between the start time that is specified by *expr1* and the end time that is specified by *expr2*. *expr1* and *expr2* are time or date and time expressions. The data types of *expr1* and *expr2* values must be the same.

The result that is returned by the `TIMEDIFF()` function must fall within the valid range of time values. You can also use the `TIMESTAMPDIFF()` and `UNIX_TIMESTAMP()` functions. These functions return integer values.

Examples

```
mysql> select timediff(now(), '2017-06-06 11:11:22'), timediff('2015-06-06 12:12:12', '2014-06-05 11:11:11')\G;
***** 1. row *****
      timediff(now(), '2017-06-06 11:11:22'): 315:00:15
      timediff('2015-06-06 12:12:12', '2014-06-05 11:11:11'): 838:59:59
1 row in set, 1 warning (0.00 sec)
```

TIMESTAMPDIFF(*unit*,*datetime_expr1*,*datetime_expr2*)

This function returns the difference between the value that is specified by *datetime_expr1* and the value that is specified by *datetime_expr2*. The returned result is an integer. The *unit* parameter specifies the unit of the returned result.

Valid values of the *unit* parameter are `MICROSECOND` (microsecond) , `SECOND`, `MINUTE`, `HOUR`, `DAY`, `WEEK`, `MONTH`, `QUARTER`, and `YEAR`.

Examples

```
mysql> select timestampdiff(second,now(), '2011-01-01 11:11:11'), timestampdiff(second, '2011-01-01 11:11:11', now())
\G;
***** 1. row *****
      timestampdiff(second,now(), '2011-01-01 11:11:11'): -140843995
      timestampdiff(second, '2011-01-01 11:11:11', now()): 140843995
1 row in set (0.00 sec)
```

PERIOD_DIFF(*p1*,*p2*)

This function returns the number of months between the *p1* period and the *p2* period. *p1* and *p2* values must use the YYMM or YYYYMM format.

 **Notice** *p1* and *p2* values are not date values.

Examples

```
mysql> select period_diff(20150702, 20790503), period_diff(150702, 790503);
+-----+-----+
| period_diff(20150702, 20790503) | period_diff(150702, 790503) |
+-----+-----+
|          -76777 |          -76777 |
+-----+-----+
1 row in set (0.00 sec)
```

TO_DAYS(*date*)

This function returns the number of days between the zero year and the date that is specified by the *date* parameter.

Examples

```
mysql> SELECT TO_DAYS('2015-11-04'), TO_DAYS('20151104');
+-----+-----+
| TO_DAYS('2015-11-04') | TO_DAYS('20151104') |
+-----+-----+
|          736271 |          736271 |
+-----+-----+
1 row in set (0.00 sec)
```

TO_DAYS() is not used for the dates that are earlier than the year 1582 when the Gregorian calendar was first introduced. This is because some days were lost when the Julian calendar was switched to the Gregorian calendar. The lost days are not considered in this function.

Two-digit year values in dates are converted into four-digit year values. For example, '2015-11-04' and '15-11-04' represent the same date.

Examples

```
mysql> SELECT TO_DAYS('2015-11-04'), TO_DAYS('151104');
+-----+-----+
| TO_DAYS('2015-11-04') | TO_DAYS('151104') |
+-----+-----+
|          736271 |          736271 |
+-----+-----+
1 row in set (0.00 sec)
```

The system considers the zero date '0000-00-00' as an invalid date. The TO_DAYS() function returns the following result for the zero date:

```
mysql> SELECT TO_DAYS('0000-00-00');
+-----+
| TO_DAYS('0000-00-00') |
+-----+
|          NULL |
+-----+
1 row in set, 1 warning (0.01 sec)
mysql> SHOW WARNINGS;
+-----+-----+-----+
| Level | Code | Message |
+-----+-----+-----+
| Warning | 1292 | Incorrect datetime value: '0000-00-00' |
+-----+-----+-----+
1 row in set (0.00 sec)
```

FROM_DAYS(N)

The function returns a DATE value based on the number of days that you specify by N.

Examples

```
mysql> SELECT FROM_DAYS(736271), FROM_DAYS(700000);
+-----+-----+
| FROM_DAYS(736271) | FROM_DAYS(700000) |
+-----+-----+
| 2015-11-04   | 1916-07-15   |
+-----+-----+
1 row in set (0.00 sec)
```

 **Note** Use caution when you use the FROM_DAYS() function to process ancient dates. This function is not used to process the dates that are earlier than the 1582 year when the Gregorian calendar was first introduced.

FROM_UNIXTIME(unix_timestamp[,format])

```
FROM_UNIXTIME(unix_timestamp) , FROM_UNIXTIME(unix_timestamp,format)
```

This function converts a unix_timestamp value into a value in the 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS format. The format depends on whether the function uses strings or numbers as arguments.

If you set the format parameter to a format string, the function returns the result based on the specified format. The format values in this function support those for the format input parameter in the DATE_FORMAT() function.

Values of the format parameter in FROM_UNIXTIME() describes the format values that can be used in a format string.

Values of the format parameter in FROM_UNIXTIME()

Format	Description
%a	The abbreviation of the day of the week. The returned values range from Sun to Sat.
%b	The abbreviation of the day of the week. The returned values range from Jan to Dec.
%c	The month of the numeric data type. The returned values range from 0 to 12.
%D	The day that ends with an ordinal indicator: st, nd, rd, or th. The returned values range from 0th to 31st.
%d	The day of the numeric data type in the month. The returned values range from 00 to 31.
%e	The day of the numeric data type in the month. The returned values range from 0 to 31.
%f	The microsecond. The returned values range from 000000 to 999999.
%H	The hour. The returned values range from 00 to 23.
%h	The hour. The returned values range from 01 to 12.
%l	The hour. The returned values range from 01 to 12.

Format	Description
%i	The minute of the numeric data type. The returned values range from 00 to 59.
%j	The number of days in the year. The returned values range from 001 to 366.
%k	The hour. The returned values range from 0 to 23.
%l	The hour. The returned values range from 1 to 12.
%M	The name of the month. The returned values range from January to December.
%m	The month of the numeric data type. The returned values range from 00 to 12.
%p	The morning or the afternoon. The returned value for the morning is AM. The returned value for the afternoon is PM.
%r	The time in the 12-hour clock. The returned value uses the hh:mm:ss format and ends with AM or PM. hh represents hours, mm represents minutes, and ss represents seconds.
%S	The seconds. The returned values range from 00 to 59.
%s	The seconds. The returned values range from 00 to 59.
%T	The time in the 24-hour clock. The returned value uses the hh:mm:ss format.
%U	The sequence number of the week in the year when Sunday is the first day of each week. The returned values range from 00 to 53.
%u	The sequence number of the week in the year when Monday is the first day of each week. The returned values range from 00 to 53.
%V	The sequence number of the week in the year when Sunday is the first day of each week. This option is used in conjunction with %X. The returned values range from 01 to 53.
%v	The sequence number of the week in the year when Monday is the first day of each week. This option is used in conjunction with %x. The returned values range from 01 to 53.
%W	The day of the week. The returned values range from Sunday to Saturday.
%w	The day of the week. The returned values range from 0 to 6. The value 0 represents Sunday. The value 6 represents Saturday. Similar rules apply to numbers 1 to 5.

Format	Description
%X	The year of the week when Sunday is the first day of each week. The year is represented by a number that consists of four digits. This option is used in conjunction with %V.
%x	The year of the week when Monday is the first day of each week. The year is represented by a number that consists of four digits. This option is used in conjunction with %v.
%Y	The year of the numeric data type. The year is represented by a number that consists of four digits.
%y	The year of the numeric data type. The year is represented by a number that consists of two digits.
%%	The literal character %.

All the other characters are copied into the returned result and do not need to be interpreted.

 **Note** The % characters must be placed before the format values.

Examples

```
mysql> SELECT FROM_UNIXTIME(875996580);
+-----+
| FROM_UNIXTIME(875996580) |
+-----+
| 1997-10-05 04:23:00   |
+-----+
1 row in set (0.00 sec)

mysql> SELECT FROM_UNIXTIME(875996580) + 0;
+-----+
| FROM_UNIXTIME(875996580) + 0 |
+-----+
|          19971005042300 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT FROM_UNIXTIME(UNIX_TIMESTAMP(), '%Y %D %M %h:%i:%s %x');
+-----+
| FROM_UNIXTIME(UNIX_TIMESTAMP(), '%Y %D %M %h:%i:%s %x') |
+-----+
| 2016 6th January 10:18:22 2016          |
+-----+
1 row in set (0.01 sec)
```

18.5.7.3. String functions

 **Note** ApsaraDB for OceanBase supports only the UTF8MB4 character set.

CONCAT(*str1*,..., *strN*)

This function concatenates one or more strings into a single string. Each input value in CONCAT functions must be a string or a NULL value. Otherwise, the system reports an error. If the concatenation is successful, the function returns the concatenated string. If an argument is NULL, the function returns NULL.

 **Note** If the data types of the *str* parameter are numeric data types, the system converts numeric values into strings in an implicit way.

Examples

```
mysql> select concat('test'), concat('test','OceanBase'), concat('test', 'OceanBase', '1.0'), concat('test','OceanBase','1.0',
NULL)\G;
***** 1. row *****
      concat('test'): test
      concat('test','OceanBase'): testOceanBase
      concat('test', 'OceanBase', '1.0'): testOceanBase1.0
      concat('test','OceanBase','1.0', NULL): NULL
1 row in set (0.00 sec)
```

Errors

- If the syntax of the function is invalid, the system reports the following error: **ERROR 1064 (42000): You have an error in your SQL syntax; .**
- If the format of a data type is invalid, the system reports the following error: **ERROR 1054 (42S22): Unknown column .** For example, if you do not enclose a string in double quotations marks ("), the system reports the error.

SUBSTRING

SUBSTRING(*str*,*pos*)

SUBSTRING(*str* FROM *pos*)

SUBSTRING(*str*,*pos*,*len*)

SUBSTRING(*str* FROM *pos* FOR *len*)

This function has the same semantics as the SUBSTR function.

SUBSTR

```
SUBSTR(str,pos,len)
```

```
SUBSTR(str,pos)
```

```
SUBSTR(str FROM pos)
```

```
SUBSTR (str FROM pos FOR len)
```

This function extracts a substring from a specified string. The *pos* parameter specifies the start position of the returned substring. The *len* parameter specifies the length of the returned substring. To use the standard SQL syntax to specify the start position, specify the *pos* parameter in the FROM clause.

- The values of *str* must be strings. The values of *pos* and *len* must be integers. If an argument is NULL, the function returns NULL.
- The Chinese characters that you specify in the values of *str* are identified as byte streams.
- If you do not specify the *len* parameter in the function, the returned substring starts from the position *pos* and ends with the last character of the source string.
- If you specify the *pos* parameter as a negative value, the function determines the start position of the returned substring based on the right-of-left order. If you specify the *pos* parameter as 0, the system considers 0 as 1. This means that the returned substring starts with the first character of the source string.
- If the *len* value is smaller than or equal to 0 or no character exists at the *pos* position, the returned result is an empty string.

```
mysql> SELECT SUBSTR('abcdefg',3), SUBSTR('abcdefg',3,2), SUBSTR('abcdefg',-3), SUBSTR('abcdefg',3,-2), SUBSTR('abcdefg' from -4 for 2)\G;
***** 1. row *****
SUBSTR('abcdefg',3): cdefg
SUBSTR('abcdefg',3,2): cd
SUBSTR('abcdefg',-3): efg
SUBSTR('abcdefg',3,-2):
SUBSTR('abcdefg' from -4 for 2): de
1 row in set (0.00 sec)
```

Errors

If the syntax is invalid, the system reports the following error: **ERROR 1064 (42000): You have an error in your SQL syntax; .**

TRIM([BOTH | LEADING | TRAILING]) [remstr] FROM] str)

This function can remove leading and trailing characters from a specified string at a time. This function can also remove only leading or trailing characters from a sting at a time.

- The values of *remstr* and *str* parameters must be strings or NULL. If an argument is NULL, the function returns NULL.
- If you do not specify BOTH, LEADIN, or TRAILING, BOTH is used by default.
- The *remstr* parameter is optional. If you do not specify this parameter, the function removes spaces from a specified string.

```
mysql> SELECT TRIM(' bar '),
TRIM(LEADING 'x' FROM 'xxxbarxxx'),
TRIM(BOTH 'x' FROM 'xxxbarxxx'),
TRIM(TRAILING 'x' FROM 'xxxbarxxx')\G;
***** 1. row *****
TRIM(' bar '): bar
TRIM(LEADING 'x' FROM 'xxxbarxxx'): barxxx
TRIM(BOTH 'x' FROM 'xxxbarxxx'): bar
TRIM(TRAILING 'x' FROM 'xxxbarxxx'): xxxbar
1 row in set (0.00 sec)
```

LENGTH(*str*)

This function returns the length of a specified string. The returned length is measured in bytes. The values of the input parameter must be strings or NULL. Otherwise, the system reports an error. If the operation is successful, the function returns an integer of the INT data type. The returned integer indicates the string length. If an argument is NULL, the function returns NULL.

If the data types of the *str* parameter are numeric data types, the system converts numeric values into strings in an implicit way.

```
mysql> SELECT LENGTH('text');
+-----+
| LENGTH('text') |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)
mysql> select length(-1.23);
+-----+
| length(-1.23) |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
mysql> select length(1233e);
mysql> select length(1233e);
ERROR 1054 (42S22): Unknown column '1233e' in 'field list'
```

Errors

If the data type of the *str* parameter is invalid, the system reports the following error: **ERROR 1054 (42S22):** Unknown column 'XXXX' in 'field list' .

UPPER(*str*)

This function converts a specified string into uppercase characters. The values of the *str* parameter must be strings. If the *str* value is NULL, the function returns NULL.

If the data types of the *str* parameter are numeric data types, the system converts numeric values into strings in an implicit way.

The byte range of the Chinese character set does not overlap with that of the ASCII character set. Therefore, you can specify Chinese characters in the input values of the UPPER function.

```
mysql> SELECT UPPER ('Hello, OceanBase!') ;
+-----+
| UPPER('Hello, OceanBase!') |
+-----+
| HELLO, OCEANBASE!        |
+-----+
1 row in set (0.00 sec)

mysql> select upper(e);
ERROR 1054 (42S22): Unknown column 'e' in 'field list'

mysql> select upper(1.235.) ;
ERROR 1064 (42000): You have an error in your SQL syntax;
```

Errors

- If the data type of the `str` parameter is invalid, the system reports the following error: **ERROR 1064 (42000):** You have an error in your SQL syntax; .
- If you do not use double quotation marks (") to enclose a string, the system reports the following error: **ERROR 1054 (42S22):** Unknown column 'XX' in 'field list' . This error is different from the following error: **ERROR 1166 (42703):** Unknown column Name that ApsaraDB for OceanBase reports.

LOWER(*str*)

This function converts a specified string into lowercase characters. The values of the `str` parameter must be strings. If the `str` value is NULL, the function returns NULL.

If the data types of the `str` parameter are numeric data types, the system converts numeric values into strings in an implicit way.

The byte range of the Chinese character set does not overlap with that of the ASCII character set. Therefore, you can specify Chinese characters in the input values of the LOWER function.

Examples

```
mysql> SELECT LOWER ('Hello, OceanBase!') ;
+-----+
| LOWER('Hello, OceanBase!') |
+-----+
| hello, oceanbase!        |
+-----+
1 row in set (0.00 sec)
mysql> select lower(1.23) ;
+-----+
| lower(1.23) |
+-----+
| 1.23      |
+-----+
1 row in set (0.00 sec)
mysql> select lower(1.23h);
ERROR 1583 (42000): Incorrect parameters in the call to native function 'lower'
mysql> select lower(1.23e);
ERROR 1582 (42000): Incorrect parameter count in the call to native function 'lower'
```

Errors

- If the syntax is invalid, the system reports the following error: `ERROR 1064 (42000): You have an error in your SQL syntax; .`
- If the specified value of an input parameter is invalid, the system reports the following errors: `ERROR 1582 (42000): Incorrect parameter count in the call to native function 'lower'` or `ERROR 1583 (42000): Incorrect parameters in the call to native function 'lower'`.

HEX(*str*)

This function converts a specified string into a hexadecimal string. If the *str* value is NULL, the function returns NULL.

If the *str* value is a numeric value, this function converts the integer part of the numeric value into a hexadecimal string.

If the *str* value is a string, the function returns a hexadecimal string for the *str* value. Each character in the *str* value is converted into two hexadecimal digits.

Examples

```
mysql> SELECT HEX(255);
-> 'FF'
mysql> SELECT HEX('abc');
-> 616263
mysql> SELECT HEX('OceanBase'),
HEX(123),
HEX(0x0123);
+-----+-----+-----+
| HEX('OceanBase') | HEX(123) | HEX(0x0123) |
+-----+-----+-----+
| 4F6365616E42617365 | 7B   | 0123   |
+-----+-----+-----+
1 row in set (0.00 sec)
mysql> select hex(0x012);
+-----+
| hex(0x012) |
+-----+
| 0012   |
+-----+
1 row in set (0.00 sec)
```

Errors

If the syntax is invalid, the system reports the following error: **ERROR 1064 (42000): You have an error in your SQL syntax;** .

UNHEX(*str*)

This function is an inverse function of HEX(*str*). The UNHEX(*str*) function interprets each pair of hexadecimal digits in the *str* value as a number and converts this number into a character. The returned value is a binary string.

The values of *str* must be strings or NULL. If the *str* value is a valid hexadecimal string, the UNHEX(*str*) function uses an algorithm to convert hexadecimal values into byte streams. If the *str* value is not a hexadecimal string, the function returns NULL. If the *str* value is NULL, the function returns NULL.

Examples

```
mysql> SELECT HEX('OceanBase'),
UNHEX('4f6365616e42617365'),
UNHEX(HEX('OceanBase')),
UNHEX(NULL)\G;
***** 1. row *****
HEX('OceanBase'): 4F6365616E42617365
UNHEX('4f6365616e42617365'): OceanBase
UNHEX(HEX('OceanBase')): OceanBase
UNHEX(NULL): NULL
1 row in set (0.00 sec)
mysql> select unhex(abc);
ERROR 1054 (42S22): Unknown column 'abc' in 'field list';
```

Errors

If the data type of the `str` parameter is invalid, the system reports the following error: `ERROR 1054 (42522): Unknown column 'abc' in 'field list'` .

INT2IP(*int_value*)

 **Note** The `INT2IP` function is available only in ApsaraDB for OceanBase.

This function converts a specified integer into an IP address.

The data type of the `int_value` parameter must be `INT`. If the value of the `int_value` parameter is `NULL`, the function returns `NULL`. If the specified integer is greater than `MAX_INT32` or smaller than `0`, the function returns `NULL`. `MAX_INT32` is the maximum signed 32-bit integer.

Examples

```
mysql> SELECT INT2IP(16777216),
-> HEX(16777216),
-> INT2IP(1);
+-----+-----+-----+
| INT2IP(16777216) | HEX(16777216) | INT2IP(1) |
+-----+-----+-----+
| 1.0.0.0      | 1000000      | 0.0.0.1   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

IP2INT('ip_addr')

 **Note** The `IP2INT` function is available only in ApsaraDB for OceanBase.

This function converts an IP address of the `STRING` data type into an integer.

Pay attention to the following considerations when you use the function:

- The values of the `ip_addr` parameter must be strings.
If the value of the `ip_addr` parameter is `NULL`, the function returns `NULL`. If the specified IP address is invalid, the function returns `NULL`. For example, the IP address contains non-digit characters, or the number in each segment of an IP address is larger than `256`.
- This function supports only IPv4 addresses.

Examples

```
mysql> SELECT IP2INT('0.0.0.1'),
HEX(IP2INT('0.0.0.1')),
HEX(IP2INT('1.0.0.0')),
IP2INT('1.0.0.257')\G;
***** 1. row *****
IP2INT('0.0.0.1'): 1
HEX(IP2INT('0.0.0.1')): 1
HEX(IP2INT('1.0.0.0')): 1000000
IP2INT('1.0.0.257'): NULL
1 row in set (0.01 sec)
```

[NOT] LIKE *str2* [ESCAPE *str3*]

This function compares strings based on wildcards. The arguments on the left of the LIKE or NOT LIKE keyword and those on the right of the keyword must be strings or NULL. Otherwise, the system reports an error. If the left argument matches the right argument, the LIKE or NOT LIKE function returns TRUE. If the left argument does not match the right argument, the LIKE or NOT LIKE function returns FALSE. If an argument is NULL, the function returns NULL.

The function supports the following wildcards: percent signs (%) and underscores (_).

- A percent sign (%) matches a string of zero or more characters.
- An underscore (_) matches a single character. The matched character must exist.

If you need to search for `a_c` instead of `abc`, you can use `a_c`. ApsaraDB for OceanBase uses double backslashes (\\) as escape characters. In `a_c`, the double backslashes (\\) are used to escape the underscore (_).

You can use the ESCAPE clause to specify an escape character. If the *str2* value contains the *str3* value, the *str2* characters that follow the *str3* value are processed as general characters. For example, in `LIKE 'abc%' ESCAPE 'c'`, `c` is an escape character. In this case, the percent sign (%) is a general character instead of an escape character. The matched string for this SQL statement is `ab%`.

```
mysql> SELECT 'ab%' LIKE 'abc%' ESCAPE 'c';
+-----+
| 'ab%' LIKE 'abc%' ESCAPE 'c' |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

 **Note** When you use the ESCAPE clause to specify an escape character, the escape character must be a one-character string. In the ESCAPE clause, percent signs (%) or underscores (_) cannot be specified as escape characters.

Examples

```
mysql> select 'a_c' like 'a\\_c';
+-----+
| 'a_c' like 'a\\_c' |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

mysql> select 'abc_' like 'abcdd_' escape 'dd';
ERROR 1064 (42000): Incorrect arguments to ESCAPE
```

expr* [NOT] REGEXP | RLIKE *pat

This function compares a specified string expression with a pattern. The *expr* parameter specifies the string. The *pat* parameter specifies the pattern.

If the *expr* value matches the *pat* value, the function returns 1. Otherwise, the function returns 0. If the *expr* value or the *pat* value is NULL, the function returns NULL.

RLIKE is a synonym for REGEXP.

The values of *expr* and *pat* must be strings or NULL. If the values of the two parameters are numeric values, the system converts the values into strings in an implicit way. If the data type of the *expr* or *pat* parameter is invalid, the system reports an error.

The specified pattern must be a valid regular expression. Otherwise, the system reports an error.

```
mysql> select 1234 regexp 1;
+-----+
| 1234 regexp 1 |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
mysql> select 'hello' rlike 'h%';
+-----+
| 'hello' rlike 'h%' |
+-----+
|          0 |
+-----+
1 row in set (0.00 sec)
mysql> select 1234 regexp ^y;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '^y' at line 1
mysql> select yunzhi regexp ^y;
ERROR 1054 (42522): Unknown column 'yunzhi' in 'field list'
mysql> select 'hello' not rlike '*h*';
ERROR 1139 (42000): Got error 'repetition-operator operand invalid' from regexp
mysql> select 'hello' not rlike '*h*';
ERROR 1139 (42000): Got error 'repetition-operator operand invalid' from regexp
```

Errors

- If the syntax of is invalid, the system reports the following error: `ERROR 1064 (42000): You have an error in your SQL syntax; .`
- If the data type of the *expr* or *pat* argument is invalid, the system reports the following error: `ERROR 1054 (42522): Unknown column 'yunzhi' in 'field list' .`
- If the system cannot identify the specified regular expression, the system reports the following error: `ERROR 1139 (42000): Got error 'repetition-operator operand invalid' from regexp .`

REPEAT(*str*, *count*)

This function repeats a string for a specified number of times. The *str* parameter specifies the string to be repeated. The *count* parameter specifies the number of times. If the specified *count* value is smaller than or equal to 0 , the function returns an empty string.

If the value of *str* or *count* is NULL, the function returns NULL.

If the data types of the *str* parameter are numeric data types, the system converts numeric values into strings in an implicit way.

The system can convert the data type of the *count* parameter into a numeric data type in an implicit way. If the conversion fails, the systems considers the *count* value as 0.

Examples

```
mysql> select repeat('1',-1), repeat(null,null),repeat('test',4);
+-----+-----+-----+
| repeat('1',-1) | repeat(null,null) | repeat('test',4) |
+-----+-----+-----+
|          | NULL          | testtesttesttest |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select repeat(11111,'2');
+-----+
| repeat(11111,'2') |
+-----+
| 1111111111      |
+-----+
1 row in set (0.00 sec)
```

SUBSTRING_INDEX(*str*, *delim*, *count*)

This function returns a specified substring of a string. The returned substring is located before the position where a specified delimiter last occurs in the specified string. In this function, the *delim* parameter specifies the delimiter. The *count* parameter specifies the number of occurrences for the delimiter. The *str* parameter specifies the string from which you want to extract the substring.

If the *count* value is positive, the function determines the last delimiter in the string based on the left-to-right order. In this case, the function returns the characters that are located on the left of the last delimiter. If the *count* value is negative, the function determines the last delimiter in the string based on the right-to-left order. In this case, the function returns the characters that are located on the right of the last delimiter.

If an argument is NULL, the function returns NULL. If the value of the *str* or *delim* parameter is an empty string, the function returns an empty string. If the *count* value is 0, the function returns an empty string.

If the data types of the *str*, *delim*, and *count* parameters are numeric data types, the system converts numeric values into strings in an implicit way.

Examples

```
mysql> select substring_index('abcdabc', 'abc', 0), substring_index('abcdabc', 'abc', 1), substring_index('abcdabc', 'abc',
2), substring_index('abcdabc', 'abc', 3), substring_index('abcdabc', 'abc', -1), substring_index('abcdabc', 'abc', -2), substri
ng_index('abcdabc', 'abc', -3)\G;
***** 1. row *****
substring_index('abcdabc', 'abc', 0):
substring_index('abcdabc', 'abc', 1):
substring_index('abcdabc', 'abc', 2): abcd
substring_index('abcdabc', 'abc', 3): abcdabc
substring_index('abcdabc', 'abc', -1):
substring_index('abcdabc', 'abc', -2): dabc
substring_index('abcdabc', 'abc', -3): abcdabc
1 row in set (0.00 sec)
```

LOCATE(*substr*,*str*) , LOCATE(*substr*,*str*,*pos*)

The `LOCATE(substr,str)` function returns the position of the first occurrence for a specified substring in a string. The `substr` parameter specifies the substring and the `str` parameter specifies the source string. The `LOCATE(substr,str,pos)` function also returns the position of the first occurrence for a specified string in a string. However, this function adds the following input parameter to the `LOCATE(substr,str)` function: `pos`. The `pos` parameter specifies the position where the function starts to search for the substring. The `substr` parameter specifies the substring and the `str` parameter specifies the source string. If the `substr` substring is not included in the `str` source string, the function returns 0.

Examples

```
mysql> SELECT LOCATE('bar', 'foobarbar');
      -> 4
mysql> SELECT LOCATE('xbar', 'foobar');
      -> 0
mysql> SELECT LOCATE('bar', 'foobarbar',5);
      -> 7
```

INSTR(*str,substr*)

This function returns the position of the first occurrence for a specified substring in a string that is specified by `str`. This function is the same as the `LOCATE(substr,str)` function, except that the order of the arguments is reversed.

Examples

```
mysql> SELECT INSTR('foobarbar', 'bar');
      -> 4
mysql> SELECT INSTR('xbar', 'foobar');
      -> 0
```

REPLACE(*str, from_str, to_str*)

This function replaces each specified substring in a string with a new substring. The `from_str` parameter specifies the substring to be replaced. The `to_str` parameter specifies the substring that replaces the specified substring. The `str` parameter specifies the source string.

Examples

```
mysql> SELECT REPLACE('abc.efg.gpg.nowdew.abc.dabc.e', 'abc.', 'www');
+-----+
| REPLACE('abc.efg.gpg.nowdew.abc.dabc.e', 'abc.', 'www') |
+-----+
| wwwefg.gpg.nowdew.wwwdwww                               |
+-----+
1 row in set (0.00 sec)
```

FIELD(*str,str1,str2,str3,...*)

This function returns the index position of a specified string in a list of strings. In this function, `str` specifies the string and `str1,str2,str3,...` specifies the list of strings. The sequence numbers for index positions start from 1. If the `str` value is not found, the function returns 0.

If all the input values in the `FIELD()` function are strings, the function compares the input values based on the strings. If all the input values are numbers, the function compares the input values based on the numbers. In other scenarios, the function compares the input values based on the `DOUBLE` numbers.

If the *str* value is NULL, the function returns 0. This is because NULL cannot be compared with the other values. FIELD() is a supplement to ELT().

Examples

```
mysql> select field('abc','abc1','abc2','abc','abc4','abc'), field(NULL, 'null1', NULL);
+-----+-----+
| field('abc','abc1','abc2','abc','abc4','abc') | field(NULL, 'null1', NULL) |
+-----+-----+
|          3 |          0 |
+-----+-----+
1 row in set (0.00 sec)
```

ELT(N, *str1*, *str2*, *str3*,...)

This function returns a specified string in a list of strings. The returned string is specified by the index number N. For example, if you set N to 1, the function returns the *str1* value. If you set N to 2, the function returns the *str2* value. If N is smaller than 1 or greater than the number of strings in the function, the function returns NULL. ELT() is a supplement to FIELD().

```
mysql> select elt(3, 'abc1', 'abc2', 'abc', 'abc4', 'abc'), elt(0, 'null1', NULL);
+-----+-----+
| elt(3, 'abc1', 'abc2', 'abc', 'abc4', 'abc') | elt(0, 'null1', NULL) |
+-----+-----+
| abc          | NULL          |
+-----+-----+
1 row in set (0.00 sec)
```

INSERT (*str1*,*pos*,*len*,*str2*)

This function replaces a specified substring of a string with a new substring. The *str1* parameter specifies the source string and the *str2* parameter specifies the substring that replaces the original substring. The *pos* parameter specifies the start position of the original substring and the *len* parameter specifies the length of the original substring. If the *pos* value is greater than the length of the source string, the function returns the source string. If the *len* value is greater than the length of the *str1* or *str2* string, the function replaces the original substring that starts at the *pos* position. If an argument is NULL, the function returns NULL. This function supports multibyte characters.

- The values of *str1* and *str2* must be strings. The values of *pos* and *len* must be integers. If an argument is NULL, the function returns NULL.
- The text characters in *str1* and *str2* are identified as byte streams.
- If the *pos* value is negative or greater than the length of the *str1* value, the function returns the *str1* value.
- If the *len* value is smaller than 0 or greater than the length of the *str1* value, the function returns a string that consists of the *str2* value and a substring of the *str1* string. The substring starts from the first character of the *str1* value and ends at the position that is specified by the *pos* parameter.

Examples

```
mysql> select insert('Quadratic',-2,100,'What'), insert('Quadratic',7,3,'What'), insert('Quadratic',1,3,'What'), insert('Quadratic',10,3,'What'), insert('Quadratic',5,-1,''), insert('Quadratic',7,-1,'What')\G;
***** 1. row *****
insert('Quadratic',-2,100,'What'): Quadratic
insert('Quadratic',7,3,'What'): QuadraWhat
insert('Quadratic',1,3,'What'): QWhatratic
insert('Quadratic',10,3,'What'): Quadratic
insert('Quadratic',5,-1,''): Quad
insert('Quadratic',7,-1,'What'): QuadraWhat
1 row in set (0.00 sec)
```

18.5.7.4. Type conversion functions

CAST(*expr AS type*)

This function converts a value of *expr* to a value of a data type that is specified by *type*. For more information about data types, see [Data types](#).

Parameter description

expr specifies the valid SQL expression.

AS separates the two parameters. The parameter before AS specifies the data to be processed. The parameter after AS specifies the destination data type.

type specifies the destination data type. This function converts the *expr* value into a value of the specified data type. You can use one of the following data types as the destination data type:

- CHAR[(N)]. If you use CHAR[N] as the destination data type in the CAST function, the value of the CHAR data type cannot exceed N characters in length.
- DATE
- DATETIME
- DECIMAL
- SIGNED [INTEGER]
- TIME
- UNSIGNED [INTEGER]

The CAST function is applicable when one of the following conditions is met:

- The data types of the two expressions are the same.
- The data types of the two expressions can be converted in an implicit way.
- The data types must be converted in an explicit way.

If an attempt is made to perform an invalid conversion, ApsaraDB for OceanBase returns an error message.

If the length of a data type is not specified, the system uses the maximum length that is supported for the data type in ApsaraDB for OceanBase. For example, the maximum length for the VARCHAR data type is 262,143 bytes. The maximum length for a numeric data type is 65 bits for floating-point numbers.

You can use the CAST function to convert signed and unsigned 64-bit values. If you use a numeric operator such as a plus sign (+) and one of the operands is an unsigned integer, the function returns an unsigned value. To override the numeric operator, you can use the SIGNED or UNSIGNED cast operator. The SIGNED operator converts a value into a signed 64-bit integer and the UNSIGNED cast operator converts a value into an unsigned 64-bit integers.

If an operand is a floating-point value, the result is a floating-point value.

Examples

```
mysql> select cast(1-2 as unsigned), cast(cast(1-2 as unsigned) as signed);
+-----+-----+
| cast(1-2 as unsigned) | cast(cast(1-2 as unsigned) as signed) |
+-----+-----+
| 18446744073709551615 | -1 |
+-----+-----+
1 row in set (0.00 sec)
mysql> SELECT CAST(1 AS UNSIGNED) - 2.0;
+-----+
| CAST(1 AS UNSIGNED) - 2.0 |
+-----+
| -1.0 |
+-----+
1 row in set (0.00 sec)
mysql> select cast(0 as date);
+-----+
| cast(0 as date) |
+-----+
| 0000-00-00 |
+-----+
1 row in set (0.00 sec)
```

18.5.7.5. Aggregate functions

Aggregate functions perform calculations on a set of values and return a single value. Aggregate functions ignore NULL values. In most cases, aggregate functions are used in conjunction with GROUP BY clauses of SELECT statements.

All the aggregate functions are deterministic. If aggregate functions are invoked by using the same set of input values, the aggregate functions return the same value.

In ApsaraDB for OceanBase, you can specify only one argument for each aggregate function. For example, `COUNT(c1, c2)` is not supported and `COUNT(c1)` is supported.

AVG(**[DISTINCT]** *expr*)

This function returns the average for a specified data set. This function ignores the NULL values in the specified data set. The DISTINCT keyword is used to return the average of distinct expr values. If no matched rows are found, the AVG() function returns NULL.

Examples

```
mysql> select * from oceanbasetest;
+----+-----+-----+
| id | ip | ip2 |
+----+-----+-----+
| 1 | 4 | NULL |
| 3 | 3 | NULL |
| 4 | 3 | NULL |
+----+-----+-----+
3 rows in set (0.01 sec)

mysql> select avg(ip2), avg(ip), avg(distinct(ip)) from oceanbasetest;
+-----+-----+-----+
| avg(ip2) | avg(ip) | avg(distinct(ip)) |
+-----+-----+-----+
| NULL | 3.3333 | 3.5000 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select avg(distinct(ip)),avg(ip),avg(ip2) from oceanbasetest;
+-----+-----+-----+
| avg(distinct(ip)) | avg(ip) | avg(ip2) |
+-----+-----+-----+
| 3.5000 | 3.3333 | NULL |
+-----+-----+-----+
1 row in set (0.00 sec)
```

COUNT([DISTINCT] *expr*)

COUNT([DISTINCT] *expr*) returns the number of non-NULL values in the rows that are retrieved by a specified SELECT statement. If no matched rows are found, COUNT() returns 0. The DISTINCT keyword is used to return the number of distinct *expr* values.

COUNT(*) returns the number of retrieved rows. The retrieved rows may contain NULL values.

Examples

```
mysql> select * from oceanbasetest;
+----+-----+-----+
| id | ip  | ip2 |
+----+-----+-----+
| 1  | 4  | NULL |
| 3  | 3  | NULL |
| 4  | 3  | NULL |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql> select count(ip2), count(ip), count(distinct(ip)), count(*) from oceanbasetest;
+-----+-----+-----+-----+
| count(ip2) | count(ip) | count(distinct(ip)) | count(*) |
+-----+-----+-----+-----+
| 0 | 3 | 2 | 3 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

MAX([DISTINCT] *expr*)

This function returns the maximum value among a specified data set.

You can pass strings into the MAX() function as arguments. If this occurs, the strings are sorted in alphabetical order and the maximum string is returned. You can use the DISTINCT keyword in the function to obtain the maximum value among distinct *expr* value. The function returns the same maximum value if you do not use the DISTINCT keyword.

For example, the a table has three rows of data: `id=1,num=10, id=2,num=20, and id=3,num=30` .

```
mysql> SELECT MAX(num) FROM a;
+-----+
| MAX(num) |
+-----+
| 30 |
+-----+
1 row in set (0.00 sec)
```

MIN([DISTINCT] *expr*)

This function returns the minimum value among a specified data set.

You can pass strings into the MIN() function as arguments. If this occurs, the strings are sorted in alphabetical order and the minimum string is returned. You can use the DISTINCT keyword in the function to obtain the minimum value of distinct *expr* values. The function returns the same minimum value if you do not use the DISTINCT keyword.

For example, the a table has three rows of data: `id=1,num=10, id=2,num=20, and id=3,num=30` .

```
mysql> SELECT MIN(num) FROM a;
+-----+
| MIN(num) |
+-----+
|      10 |
+-----+
1 row in set (0.00 sec)
```

SUM([DISTINCT] *expr*)

This function returns the sum of *expr* values. If no rows are found for *expr*, the SUM() function returns NULL. You can use the DISTINCT keyword to obtain the sum of distinct *expr* values.

If no matched rows are found, the SUM() function returns NULL.

Examples

```
mysql> select * from oceanbasetest;
+-----+-----+-----+
| id | ip | ip2 |
+-----+-----+-----+
|  1 |  4 | NULL |
|  3 |  3 | NULL |
|  4 |  3 | NULL |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select sum(ip2),sum(ip),sum(distinct(ip)) from oceanbasetest;
+-----+-----+-----+
| sum(ip2) | sum(ip) | sum(distinct(ip)) |
+-----+-----+-----+
|  NULL |  10 |      7 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

GROUP_CONCAT([DISTINCT] *expr*)

This function concatenates non-NULL strings from a specified group into a single string.

- Syntax

```
GROUP_CONCAT([DISTINCT] expr [,expr ...]
[ORDER BY {unsigned_integer | col_name | expr}
[ASC | DESC] [,col_name ...]]
[SEPARATOR str_val])
```

- Examples

```

mysql> select * from book; //The table named book (book number, book title, publisher) +-----+-----+
-----+-----+
| bookid | bookname          | publishname      |
+-----+-----+-----+
| 1 | git help          | alibaba group publisher |
| 2 | MySQL Optimization | Zhejiang University Press |
| 3 | Java Programming Guide | Machinery Industry Press |
| 3 | Java Programming Guide | Machinery Industry Press |
| 4 | Large-Scale Distributed Storage System | Machinery Industry Press |
+-----+-----+-----+
5 rows in set (0.00 sec)

//Retrieve book titles.
mysql> select group_concat(bookname) from book group by bookname;
+-----+
| group_concat(bookname) |
+-----+
| git help              |
| Java Programming Guide, Java Programming Guide |
| MySQL Optimization    |
| Large-Scale Distributed Storage System |
+-----+
4 rows in set (0.00 sec)

//Retrieve distinct book titles.
mysql> select group_concat(distinct(bookname)) from book group by bookname;
+-----+
| group_concat(distinct(bookname)) |
+-----+
| git help                        |
| Java Programming Guide          |
| MySQL Optimization              |
| Large-Scale Distributed Storage System |
+-----+
4 rows in set (0.01 sec)

mysql> select bookname, group_concat(publishname order by publishname desc separator ';') from book group by
bookname;
+-----+-----+-----+
| bookname          | group_concat(publishname order by publishname desc separator ';') |
+-----+-----+-----+
| git help          | alibaba group publisher |
+-----+-----+-----+
+-----+-----+-----+
4 rows in set (0.00 sec)

```

18.5.7.6. Mathematical functions

The functions of this type perform mathematical calculations based on numeric expressions.

ROUND(X), ROUND(X,D)

The ROUND() function returns a number that is rounded to the specified length or precision.

This function rounds the X argument to the nearest integer. If you specify two arguments for the function, the function rounds X to D decimal places. The Dth digit is obtained by rounding. To keep D digits of the X value to the left of the decimal point, set D to a negative value.

The data type of the returned value is the same as that of the first argument. You can assume that the value is an integer, double-precision floating-point number, or decimal value. This means that the returned result for an integer argument is also an integer. No fractional part is included in the returned result.

- For exact-value numbers, the ROUND() function rounds values to the nearest integer.
 - If the fractional part of a positive value is `.5` or greater than `.5`, the function rounds the positive value up to the next integer. If the fractional part of a negative value is greater than or equal to `.5`, the function rounds the negative value down to the next integer. In other words, the value is rounded based on the distance between zero and the specified value along the X-axis.
 - If the fractional part of a positive value is smaller than `.5`, the function rounds the value down to the next integer. If the fractional part of a negative value is smaller than `.5`, the function rounds the negative value down to the next integer.
- For approximate-value numbers, the ROUND() function follows the bankers rounding rule. If a value has a fractional part, the function rounds the value to the nearest even integer based on the rule.

Examples

```
mysql> select round(2.15,2);
+-----+
| round(2.15,2) |
+-----+
|      2.15 |
+-----+
1 row in set (0.00 sec)

mysql> select round(2555e-2,1);
+-----+
| round(2555e-2,1) |
+-----+
|      25.6 |
mysql> select round(25e-1), round(25.3e-1),round(35e-1);
+-----+-----+-----+
| round(25e-1) | round(25.3e-1) | round(35e-1) |
+-----+-----+-----+
|      3 |      3 |      4 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

CEIL(expr)

This function rounds a specified expression value up to the next largest integer.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted into a numeric value: 1 for TRUE or 0 for FALSE.

If the value of `expr` is `NULL`, the function returns `NULL`.

If you specify a string of numbers, the system converts the string into a numeric value in an implicit way.

The returned value is converted into a `BIGINT` number.

Examples

```
mysql> select ceil(1.2), ceil(-1.2), ceil(1+1.5), ceil(1=1),ceil(1<1),ceil(null);
+-----+-----+-----+-----+-----+-----+
| ceil(1.2) | ceil(-1.2) | ceil(1+1.5) | ceil(1=1) | ceil(1<1) | ceil(null) |
+-----+-----+-----+-----+-----+-----+
|      2 |      -1 |          3 |          1 |          0 |        NULL |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select ceil(name);
ERROR 1166 (42703): Unkown column name 'name'

mysql> select ceil('2');
+-----+
| ceil('2') |
+-----+
|      2 |
+-----+
1 row in set (0.00 sec)
```

FLOOR(*expr*)

This function is similar to the `CEIL(expr)` function. This function rounds a specified expression value down to the next smallest integer.

The function supports comparison operations. The comparison result is a `BOOLEAN` value. The `BOOLEAN` value is converted into a numeric value: 1 for `TRUE` or 0 for `FALSE`.

If the value of `expr` is `NULL`, the function returns `NULL`.

If you specify a string of numbers, the system converts the string into a numeric value in an implicit way.

The returned value is converted into a `BIGINT` number.

Examples

```
mysql> select floor(1.2), floor(-1.2), floor(1+1.5), floor(1=1), floor(1<1), floor(null);
+-----+-----+-----+-----+-----+-----+
| floor(1.2) | floor(-1.2) | floor(1+1.5) | floor(1=1) | floor(1<1) | floor(null) |
+-----+-----+-----+-----+-----+-----+
| 1 | -2 | 2 | 1 | 0 | NULL |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select floor(name);
ERROR 1166 (42703): Unkown column name 'name'

mysql> select floor('2');
+-----+
| floor('2') |
+-----+
| 2 |
+-----+
1 row in set (0.00 sec)
```

ABS(*expr*)

This function returns the absolute value for a specified numeric expression. The data type of the returned value is the same as that of the expression value.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted into a numeric value: 1 for TRUE or 0 for FALSE.

If the value of *expr* is NULL, the function returns NULL.

If you specify a string of numbers, the system converts the string into a numeric value in an implicit way.

The returned value is converted into a BIGINT number.

Examples

```
mysql> select abs(5), abs(-5.777), abs(0), abs(1/2), abs(1-5);
+-----+-----+-----+-----+-----+
| abs(5) | abs(-5.777) | abs(0) | abs(1/2) | abs(1-5) |
+-----+-----+-----+-----+-----+
| 5 | 5.777 | 0 | 0.5 | 4 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

NEG(*expr*)

This function is a negation function that subtracts a specified operand from zero and returns the final result.

The function supports comparison operations. The comparison result is a BOOLEAN value. The NEG function performs the NOT operation. The returned result is 1 or 0: 1 for TRUE and 0 for FALSE.

Examples

```
mysql> select neg(1),neg(1+1),neg(2*3),neg(1=1),neg(5<1);
+-----+-----+-----+-----+-----+
| neg(1) | neg(1+1) | neg(2*3) | neg(1=1) | neg(5<1) |
+-----+-----+-----+-----+-----+
| -1 | -2 | -6 | -1 | 0 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

SIGN(X)

This function returns the sign of a specified number. The function returns -1, 0, or 1 based on whether the specified X value is negative, zero, or positive.

The function supports comparison operations. The comparison result is a BOOLEAN value. The BOOLEAN value is converted into a numeric value: 1 for TRUE and 0 for FALSE.

If the value of expr is NULL, the function returns NULL.

The function supports floating-point numbers and hexadecimal numbers.

Examples

```
mysql> SELECT SIGN(-32);
-> -1
mysql> SELECT SIGN(0);
-> 0
mysql> SELECT SIGN(234);
-> 1
mysql> select sign(null),sign(false),sign(0x01);
+-----+-----+-----+
| sign(null) | sign(false) | sign(0x01) |
+-----+-----+-----+
| NULL | 0 | 1 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

CONV(N, from_base, to_base)

This function converts a number from one base to another base.

The function converts a number from the *from_base* based to the *to_base* base. The returned result is a string. The value of the N input parameter can be an integer or a string. The minimum base is 2 and the maximum base is 36. If the *to_base* value is a negative number, N is processed as a signed number. Otherwise, N is processed as an unsigned number. If the *from_base* value is a negative number, the value is processed as an integer and the value sign is ignored. The data types of the N argument must be INT or STRING.

The values of *from_base* and *to_base* parameters must be decimal INT values. The value range is the union of [-36,-2] and [2,36].

Invalid input values result in errors. In the following scenarios, the input values are invalid:

- The values of the *from_base* or *to_base* parameter are not valid decimal INT values.
- The values of the *from_base* or *to_base* parameter do not fall within the valid value range: union of [-36,-2] and [2,36] .
- N is an invalid numeric value. For example, N falls out of the following character range: 0 to 9, a to z, or A to Z.


```
mysql> select mod(29,19), 29 mod 19, 29 % 19;
```

```
+-----+-----+-----+
| mod(29,19) | 29 mod 19 | 29 % 19 |
+-----+-----+-----+
|    10 |    10 |    10 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select mod(19.5, 29);
```

```
+-----+
| mod(19.5, 29) |
+-----+
|    19.5 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select mod(29, null);
```

```
+-----+
| mod(29, null) |
+-----+
|    NULL |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select mod(100,0);
```

```
+-----+
| mod(100,0) |
+-----+
|    NULL |
+-----+
1 row in set (0.00 sec)
```

POW(X, Y)

This function raises X to the power of Y.

Examples

```
mysql> select pow(4,2), pow(4,-2), pow(1,null);
```

```
+-----+-----+-----+
| pow(4,2) | pow(4,-2) | pow(1,null) |
+-----+-----+-----+
|    16 |    0.0625 |    NULL |
+-----+-----+-----+
1 row in set (0.00 sec)
```

POWER(X, Y)

POWER(X, Y) and POW(X, Y) are equivalent.

RAND([N])

The RAND([N]) function accepts zero or one argument N and returns a random floating-point number that falls within the range of [0,1.0). N is called a random seed. If you need to retrieve a random integer that falls within the range of [i, j), you can use the expression FLOOR(I + RAND() * (j - i)).

If you do not specify N, a random seed is generated during initialization. The RAND() function generates a random number based on this random seed. Therefore, the RAND() function generates a different random number each time the function is invoked.

Examples

```
mysql> select a, b, rand() from t3;
+-----+-----+-----+
| a | b | rand() |
+-----+-----+-----+
| 1 | 1 | 0.641815407799385 |
| 2 | 2 | 0.16825051248841966 |
| 3 | 3 | 0.9158063697775886 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select a, b, rand() from t3;
+-----+-----+-----+
| a | b | rand() |
+-----+-----+-----+
| 1 | 1 | 0.07428034215632857 |
| 2 | 2 | 0.6239826321825224 |
| 3 | 3 | 0.897072165177271 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

If you specify N, N is used as the random seed to generate random numbers. The function generates random numbers based on whether N is a constant:

- If N is a constant, N is used as the random seed during initialization. Then, the RAND(N) function generates a random number based on the initialized value. If the values of N are the same, the generated random number sequences are the same.

```
mysql> select a, b, rand(3) from t3;
+-----+-----+-----+
| a | b | rand(3) |
+-----+-----+-----+
| 1 | 1 | 0.9057697559760601 |
| 2 | 2 | 0.37307905813034536 |
| 3 | 3 | 0.14808605345719125 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select a, b, rand(3) from t3;
+-----+-----+-----+
| a | b | rand(3) |
+-----+-----+-----+
| 1 | 1 | 0.9057697559760601 |
| 2 | 2 | 0.37307905813034536 |
| 3 | 3 | 0.14808605345719125 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

- If N is a variable, such as a column name, N is used as the random seed to generate a random number in each function call. If the values of N are the same, the generated random numbers are the same.

```
mysql> select a, b, rand(a), rand(b) from t3;
+-----+-----+-----+-----+
| a | b | rand(a) | rand(b) |
+-----+-----+-----+-----+
| 1 | 1 | 0.40540353712197724 | 0.40540353712197724 |
| 2 | 2 | 0.6555866465490187 | 0.6555866465490187 |
| 3 | 3 | 0.9057697559760601 | 0.9057697559760601 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

The `RAND([N])` function can be used in SELECT statements and the following clauses: `WHERE` , `ORDER BY` , and `GROUP BY` . This function in the clauses runs based on the preceding rules.

For example, if you need to sort a table in a random way, you can execute the following SQL statement: `select * from t1 order by rand()` . If you need to sample 100 rows of a table in a random way, you can execute the following SQL statement: `select * from t1 order by rand() limit 100` .

18.5.7.7. Comparison functions

Comparison functions compare input values of parameters. The values that can be compared include numbers, characters, and dates.

GREATEST(*value1*, ...)

This function returns the maximum value among the specified input values. This function performs the opposite operation of the `LEAST()` function.

You must specify at least two arguments. If an argument is NULL, the function returns NULL.

If the specified arguments contain numeric values and strings, the system converts the strings to numeric values in an implicit way. If the conversion fails, the system reports errors.

Examples

```
mysql> select greatest('2',1,0), greatest('a','b','c'), greatest('a', NULL, 'c'), greatest('2014-05-15','2014-06-01')\G;
***** 1. row *****
      greatest('2',1,0): 2
      greatest('a','b','c'): c
      greatest('a', NULL, 'c'): NULL
      greatest('2014-05-15','2014-06-01'): 2014-06-01
1 row in set (0.00 sec)
```

LEAST(*value1*, ...)

This function returns the minimum value among the specified arguments. This function performs the opposite operation of the `GREATEST()` function.

You must specify at least two arguments. If an argument is NULL, the function returns NULL.

If the specified arguments contain numeric values and strings, the system converts the strings to numeric values in an implicit way. If the conversion fails, the system reports errors.

Examples

```
mysql> select least('2',4,9), least('a','b','c'), least('a',NULL,'c'), least('2014-05-15','2014-06-01')\G;
***** 1. row *****
      least('2',4,9): 2
      least('a','b','c'): a
      least('a',NULL,'c'): NULL
      least('2014-05-15','2014-06-01'): 2014-05-15
1 row in set (0.00 sec)
```

ISNULL(*expr*)

This function checks whether a specified value of an expression is NULL. If the *expr* value is NULL, the `ISNULL()` function returns 1. Otherwise, the function returns 0.

Examples

```
mysql> SELECT ISNULL(null), ISNULL('test'), ISNULL(123.456), ISNULL('10:00');
+-----+-----+-----+-----+
| ISNULL(null) | ISNULL('test') | ISNULL(123.456) | ISNULL('10:00') |
+-----+-----+-----+-----+
|      1 |      0 |      0 |      0 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT ISNULL(null+1);
+-----+
| ISNULL(null+1) |
+-----+
|      1 |
+-----+
1 row in set (0.00 sec)
```

The ISNULL() function can be used as an alternative to the equal sign (=) comparison. You can use an equal sign (=) operator to check whether an expression value is NULL.

Note If the equal sign (=) operator is used to check whether an expression value is NULL, the returned result is invalid in most cases.

The ISNULL() function provides some features that are the same as the IS NULL operator.

18.5.7.8. Flow control functions

CASE

You can use the following syntax:

```
CASE value WHEN [compare-value] THEN result [WHEN [compare-value] THEN result ...] [ELSE result] END
```

You can also use the following syntax:

```
CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END
```

For the former syntax, the function returns `value=compare-value` .

For the latter syntax, the function returns the result if the first condition is met. If no matched results are available, the function returns the result of the ELSE part. If the ELSE part is unavailable, the function returns NULL.

Examples

```
mysql> select CASE 'b' when 'a' then 1 when 'b' then 2 END;
+-----+
| CASE 'b' when 'a' then 1 when 'b' then 2 END |
+-----+
|                2 |
+-----+
1 row in set (0.00 sec)

mysql> select CASE concat('a','b') when concat('ab') then 'a' when 'b' then 'b' end;
+-----+
| CASE concat('a','b') when concat('ab') then 'a' when 'b' then 'b' end|
+-----+
| a                |
+-----+
1 row in set (0.00 sec)

mysql> select case when 1>0 then 'true' else 'false' end;
+-----+
| case when 1>0 then 'true' else 'false' end |
+-----+
| true                |
+-----+
1 row in set (0.00 sec)
```

IF(*expr1*,*expr2*,*expr3*)

If the *expr1* condition is met, such as *expr1*<>0 and *expr1*<>NULL, the function returns the *expr2* value. Otherwise, the function returns the *expr3* value.

The IF() function returns numeric values or strings based on the data types of *expr2* and *expr3* values.

Examples

```
mysql> select if(5>6, 'T','F'), if (5>6, 1, 0), if(null, 'True', 'False'), if(0, 'True', 'False')\G;
***** 1. row *****
if(5>6, 'T','F'): F
if (5>6, 1, 0): 0
if(null, 'True', 'False'): False
if(0, 'True', 'False'): False
1 row in set (0.00 sec)
```

If one of *expr2* and *expr3* values is NULL, the data type of the IF() function result is the same as that of the non-null expression value.

Default data types of returned values describes the default data types of the values that are returned by the IF() function.

Default data types of returned values

Expression value	Default data type of the returned values
<i>expr2</i> or <i>expr3</i> returns a string.	String
<i>expr2</i> or <i>expr3</i> returns a floating-point number.	Float-pointing number

Expression value	Default data type of the returned values
<i>expr2</i> or <i>expr3</i> returns an integer.	Integer

If *expr2* and *expr3* values are strings and one of the strings is case-sensitive, the returned value is case-sensitive.

IFNULL(*expr1*,*expr2*)

If the *expr1* value is not NULL, the IFNULL() function returns the *expr1* value. Otherwise, the function returns the *expr2* value.

The IFNULL() function returns numeric values or strings based on the data types of arguments.

Examples

```
mysql> SELECT IFNULL('abc', null), IFNULL(NULL+1, NULL+2), IFNULL(1/0, 0/1);
+-----+-----+-----+
| IFNULL('abc', null) | IFNULL(NULL+1, NULL+2) | IFNULL(1/0, 0/1) |
+-----+-----+-----+
| abc          |          NULL |          0.0000 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

NULLIF(*expr1*,*expr2*)

For *expr1=expr2*, the function returns NULL. Otherwise, the function returns the *expr1* value. This rule is the same as that for CASE WHEN *expr2=expr2* THEN NULL ELSE *expr1* END .

Examples

```
mysql> SELECT NULLIF('ABC', 123), NULLIF('123',123), NULLIF(NULL, 'abc');
+-----+-----+-----+
| NULLIF('ABC', 123) | NULLIF('123',123) | NULLIF(NULL, 'abc') |
+-----+-----+-----+
| ABC          | NULL          | NULL          |
+-----+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

 Note

18.5.7.9. Information functions

FOUND_ROWS()

You may use a LIMIT clause in a SELECT statement to limit the number of rows that are returned from the database server to the client.

In some cases, you need to retrieve the actual number of rows that the statement returns if the statement does not have the LIMIT clause. To retrieve the actual number of rows, you can execute the statement that does not have the LIMIT clause again. If you do not want to execute the statement again, you can use `SQL_CALC_FOUND_ROWS` in the SELECT statement. This way, you can invoke the `FOUND_ROW()` function to retrieve the actual number of rows that are returned by the SELECT statement that does not have the LIMIT clause.

Examples

```
mysql> SELECT SQL_CALC_FOUND_ROWS * FROM tbl_name
-> WHERE id > 100 LIMIT 10;
mysql> SELECT FOUND_ROWS();
```

The second SELECT statement returns a number that may be different from that for the first SELECT statement. The number indicates how many rows are returned by the first SELECT statement if the first statement does not have the LIMIT clause.

 **Note** Assume that `SQL_CALC_FOUND_ROWS` is not used in the first SELECT statement. When you use the LIMIT clause in the statement, the `FOUND_ROWS()` function returns a value. The returned value may be different from the value that is returned when you do not use the LIMIT clause.

For the SELECT `SQL_CALC_FOUND_ROWS` statement in the preceding example, the returned value of the `FOUND_ROWS()` function is valid only for a short period. The returned value becomes invalid when the statement that follows the `SELECT SQL_CALC_FOUND_ROWS` statement is executed. If you need to use the returned number of rows later, you must save the number.

Examples

```
mysql> SELECT SQL_CALC_FOUND_ROWS * FROM ... ;
mysql> SET @rows = FOUND_ROWS();
```

If you use `SQL_CALC_FOUND_ROWS` in a query, the system calculates the number of rows in the full result set. This process requires less time than the process of running another query that does not use the LIMIT clause. This is because the result set in the former process does not need to be sent to the client.

Assume that you want to limit the number of rows that a query returns and do not want to run another query to retrieve the number of rows in the full result set. In this case, you can use `SQL_CALC_FOUND_ROWS` and `FOUND_ROWS()`. For example, you can use a web script for a paged display. The displayed information contains the links to pages for the other parts of the query results. You can use the `FOUND_ROWS()` function to determine the number of additional pages that are required to show the remaining results.

The implementation of `SQL_CALC_FOUND_ROWS` and `FOUND_ROWS()` in UNION queries is more complex than that in simple SELECT statements. This is because a UNION query may contain more than one LIMIT clause. For example, you may use LIMIT clauses in the SELECT statements of a UNION query, or you may use the clauses to limit the UNION results.

If `SQL_CALC_FOUND_ROWS` is used for a UNION query, the expected returned result is the row count that is not limited by the global LIMIT clause.

If you need to use `SQL_CALC_FOUND_ROWS` in a UNION query, make sure the following requirements are met:

- The `SQL_CALC_FOUND_ROWS` keyword must appear in the first SELECT statement of the UNION query.
- The UNION ALL syntax is used. The returned value of the `FOUND_ROWS()` function is accurate only if `UNION ALL` is used.
- If the UNION query does not contain LIMIT clauses, `SQL_CALC_FOUND_ROWS` is ignored. In this case, the query returns the number of rows in the temporary table that is created to process the UNION query.

LAST_INSERT_ID()

This function returns the auto-increment field value that is latest inserted in the current session. If you insert multiple rows into the table in the latest operation, the `LAST_INSERT_ID()` function returns the auto-increment field value of the first row.

Examples

```
mysql>select LAST_INSERT_ID();
+-----+
| LAST_INSERT_ID() |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
```

18.5.7.10. Other functions

COALESCE(*expr, expr, expr,...*)

The function evaluates the argument expressions in sequence until the function finds a non-NULL value. Then, the function returns the non-NULL value. If the values of all the expressions are NULL, the function returns a NULL value.

The data types of all the expressions must be the same. If the data types are different, the system converts the data types into the same type in an implicit way.

```
mysql> SELECT COALESCE(NULL,NULL,3,4,5), COALESCE(NULL,NULL,NULL);
+-----+-----+
| COALESCE(NULL,NULL,3,4,5) | COALESCE(NULL,NULL,NULL) |
+-----+-----+
|          3 |          NULL |
+-----+-----+
1 row in set (0.00 sec)
```

NVL(*str1,replace_with*)

If the *str1* value is NULL, this function replaces the value with the *replace_with* value.

If you set *str1* to NULL, the function returns a value that you specify for the *replace_with* parameter. This helps you retrieve complete outputs. In most cases, the *str1* value is a column name. No limits are placed on the values of `replace_with`. For example, you can specify hard-coded values, references to other columns, or expressions for the *replace_with* parameter.

Examples

```
mysql> SELECT NVL(NULL, 0), NVL(NULL, 'a');
+-----+-----+
| NVL(NULL, 0) | NVL(NULL, 'a') |
+-----+-----+
|          0 | a |
+-----+-----+
1 row in set (0.00 sec)
```

ORA_DECODE()

The `ORA_DECODE()` function runs in the same way as the Oracle `DECODE()` function.

Note ApsaraDB for OceanBase is compatible with MySQL and supports some functions of Oracle. The names of the Oracle functions that are used in ApsaraDB for OceanBase start with `ORA_`.

```
ora_decode (condition, value 1, returned value 1, value 2, returned value 2, ... value n, returned value n, default value)
```

The following section describes the meaning of this function:

```
IF condition = value 1 THEN
....RETURN (returned value 1)
ELSIF condition = value 2 THEN
....RETURN (returned value 2)
.....
ELSIF condition = value n THEN
....RETURN (returned value n)
ELSE
....RETURN (default value)
END IF
```

SLEEP(duration)

This function suspends an SQL query for a specified duration that is measured in seconds. The function returns 0 after the suspension ends.

If only the SLEEP function is run in a query that is not interrupted, the function returns 0, as shown in the following example:

```
mysql> SELECT SLEEP(1000);
+-----+
| SLEEP(1000) |
+-----+
| 0 |
+-----+
```

If only the SLEEP function is run in a query that is interrupted, the function returns 1 and does not return error codes, as shown in the following example:

```
mysql> SELECT SLEEP(1000);
+-----+
| SLEEP(1000) |
+-----+
| 1 |
+-----+
```

If the SLEEP function is part of a query and the query is interrupted due to the suspension, the function returns the error code 1317, as shown in the following example:

```
mysql> SELECT 1 FROM t1 WHERE SLEEP(1000);
ERROR 1317 (70100): Query execution was interrupted
```

18.5.7.11. Full-text search functions

```
MATCH (col1,col2,...) AGAINST (expr [search_modifier])
search_modifier:
{
  IN NATURAL LANGUAGE MODE
| IN NATURAL LANGUAGE MODE WITH QUERY EXPANSION
| IN BOOLEAN MODE
| WITH QUERY EXPANSION
}
```

ApsaraDB for OceanBase allows you to use full-text search functions to search for full-text indexes.

When you use the full-text search functions, make sure that the following requirements are met:

- Full-text indexes are created on the columns that are specified in the `MATCH(col1,col2,...)` function. The function supports only `FULLTEXT CTXCAT` indexes.
- For `FULLTEXT CTXCAT` indexes, the columns in the `MATCH(col1,col2,...)` function must include the columns that are specified for the CTXCAT indexes. For example, you can create the `FULLTEXT INDEX(c1, c2, c3) CTXCAT(c2, c3)` index. In this scenario, the retrieved data can be matched only if the function is `MATCH(c2,c3)`.
- You can use search modifiers that are included in the preceding statement to specify the search mode for full-text search functions. The default search mode is `NATURAL LANGUAGE MODE`.

 **Note** ApsaraDB for OceanBase supports only two search modes: `NATURAL LANGUAGE MODE` and `BOOLEAN MODE`.

Full-text search: NATURAL LANGUAGE MODE

If you use the default mode or specify the `IN NATURAL LANGUAGE MODE` search modifier, the `MATCH...AGAINST` function uses the `NATURAL LANGUAGE` mode to run a full-text search. In the `NATURAL LANGUAGE` mode, specify the `AGAINST` argument as a search string. In the index, the function searches for the string by comparing strings based on the character set. For each row in the table, the `MATCH` function returns a relevance value. The value indicates the relevance between the search string and the row data. To be more specific, the value indicates the similarity between the text of the search string and the text in the data table.

By default, string columns in ApsaraDB for OceanBase are not case-sensitive. Therefore, the keywords for full-text searches are not case-sensitive. If you need to run case-sensitive full-text searches, you can specify case-sensitive data types for the columns on which full-text indexes is created. For example, you can specify the data type of the columns as `UTF8MB4_BIN`.

If the `MATCH...AGAINST` function is used in the `WHERE` clause, `MATCH` is used to filter data that is irrelevant to the keywords in the function. ApsaraDB for OceanBase supports only `MATCH...AGAINST=0` and `MATCH...AGAINST>0`. `MATCH...AGAINST=0` indicates that no data matches the keywords and `MATCH...AGAINST>0` indicates that at least one keyword is matched.

You can specify multiple keywords for the `AGAINST` parameter. You must separate the keywords with spaces and enclose the keywords in a single quotation mark ('). This means that the `OR` logical operator is applied on the specified keywords. If the text in the table matches one of the keywords, a match occurs.

Full-text search: BOOLEAN MODE

In ApsaraDB for OceanBase, you can run boolean full-text searches by using the `IN BOOLEAN MODE` search modifier. In this mode, some special operators at the beginning of the keywords in search strings have special meanings.

Examples:

```
SELECT * FROM t1 WHERE MATCH (a, b)
    AGAINST ('Chrysanthemum Jasmine' IN BOOLEAN MODE);
```

id	a	b
1	Alipay	Chrysanthemum tea
2	Taobao	Jasmine

```
SELECT * FROM t1 WHERE MATCH (a, b)
    AGAINST ('+Chrysanthemum -Jasmine tea' IN BOOLEAN MODE);
```

id	a	b
1	Alipay	Chrysanthemum tea

In ApsaraDB for OceanBase, boolean full-text searches support the following operators:

- The plus sign (+) represents the AND operator. The AND operator indicates that the search results must include the keyword that is preceded by the plus sign (+).
- The minus sign (-) represents the NOT operator. The NOT operator indicates that the search results must exclude the keyword that is preceded by the minus sign (-).
- If *no operator* is specified, the OR operator is applied on the specified keywords. The OR operator indicates that the search results must include at least one of the specified keywords.

When you run boolean full-text searches, make sure the following requirements are met:

- All the operators must be placed at the beginning of keywords. The operators at the end of keywords are ignored. For example, the plus sign (+) in `+Chrysanthemums` is valid and takes effect. The plus sign (+) in `Chrysanthemums+` is invalid and ignored.
- Operators and keywords cannot be separated with other characters. If operators and keywords are separated with other characters, the operators are ignored. For example, the plus sign (+) in `+ Chrysanthemums` is ignored.

18.5.8. Operators and precedences

18.5.8.1. Overview

ApsaraDB for OceanBase supports a wide range of operators: arithmetic operators, comparison operators, vector comparison operators, logical operators, and bitwise operators. The other topics in the "Operators and precedences" chapter describe each of the listed operators and their precedences.

18.5.8.2. Logical operators

In ApsaraDB for OceanBase, the logical operators convert left and right operands into the values of the BOOLEAN data type before the operators perform calculations. If Error is returned for logical operations, calculation errors occur.

In ApsaraDB for OceanBase, you must comply with the following rules when you convert the specified values into the values of the BOOLEAN data type:

- Only the following strings can be converted into the values of the BOOLEAN data type: True, False, 1, and 0. The strings True and 1 are converted into True Boolean values. The strings False and 0 are converted into False Boolean values.
- If a numeric value of the INT, FLOAT, DOUBLE, or DECIMAL data type is converted into a value of the BOOLEAN data type and the numeric value is not zero, the result value is True. If the numeric value to be converted is zero, the result value is False.

NOT!

The logical NOT operator. If you need to use the operator to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for the operator.

INT	FLOAT	DOUBLE	TIMESTAMP	VARCHAR	BOOL	NULL
True/False	True/False	True/False	Error	True/False/E	True/False	NULL

Example:

```
mysql> SELECT NOT 0, NOT 1, NOT NULL;
+-----+-----+-----+
| NOT 0 | NOT 1 | NOT NULL |
+-----+-----+-----+
|  1   |  0   |  NULL   |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select not current_timestamp;
+-----+
| not current_timestamp |
+-----+
|          0           |
+-----+
1 row in set (0.00 sec)

mysql> select not now();
+-----+
| not now() |
+-----+
|    0     |
+-----+
1 row in set (0.00 sec)
```

AND &&

The logical AND operator. If you need to use the operator to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for the operator.

- INT FLOAT DOUBLE TIMESTAMP VARCHAR BOOL NULL INT True/False True/False True/False Error
 True/False/Error True/False False/NULL FLOAT - True/False True/False Error True/False/Error True/False
 False/NULL DOUBLE - - True/False Error True/False/Error True/False False/NULL TIMESTAMP - - - Error Error
 True/False Error VARCHAR - - - - True/False/Error True/False/Error False/NULL BOOL - - - - True/False
 False/NULL NULL - - - - - NULL

Examples:

```
mysql> SELECT (0 AND 0), (0 AND 1), (1 AND 1), (1 AND NULL);
+-----+-----+-----+-----+
| (0 AND 0) | (0 AND 1) | (1 AND 1) | (1 AND NULL) |
+-----+-----+-----+-----+
| 0 | 0 | 1 | NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

OR ||

The logical OR operator. If you need to use the operator to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for the operator.

-	INT	FLOAT	DOUBLE	TIMESTAMP	VARCHAR	BOOL	NULL	INT	True/False	True/False	True/False	Error	True/False/Error	True/False	True/NULL	FLOAT	-	True/False	True/False	Error	True/False/Error	True/False	True/NULL	DOUBLE	-	-	True/False	Error	True/False/Error	True/False	True/NULL	TIMESTAMP	-	-	Error	Error	Error	Error	VARCHAR	-	-	-	-	True/False/Error	True/False/Error	True/NULL	BOOL	-	-	-	-	True/False	True/NULL	NULL	-	-	-	-	NULL
---	-----	-------	--------	-----------	---------	------	------	-----	------------	------------	------------	-------	------------------	------------	-----------	-------	---	------------	------------	-------	------------------	------------	-----------	--------	---	---	------------	-------	------------------	------------	-----------	-----------	---	---	-------	-------	-------	-------	---------	---	---	---	---	------------------	------------------	-----------	------	---	---	---	---	------------	-----------	------	---	---	---	---	------

Example:

```
mysql> SELECT (0 OR 0), (0 OR 1), (1 OR 1), (1 AND NULL);
+-----+-----+-----+-----+
| (0 OR 0) | (0 OR 1) | (1 OR 1) | (1 AND NULL) |
+-----+-----+-----+-----+
| 0 | 1 | 1 | NULL |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

18.5.8.3. Arithmetic operators

In ApsaraDB for OceanBase, you can perform arithmetic operations on only the values of the numeric data types or the VARCHAR data type. If you perform arithmetic operations on the values of other data types, errors are reported. If you perform arithmetic operations on strings and the strings cannot be converted to the values of the DOUBLE data type, errors are reported. For example, if you perform the '3.4he' + 3 operation, an error is reported.

Strings can be converted into the values of the DOUBLE data type only in two scenarios. In one of the scenarios, each character in the strings is a digit. In the other scenario, the strings start with plus signs (+) or minus signs (-) and the characters that follow the plus signs (+) or the minus signs (-) are digits.

Supported arithmetic operators describes the arithmetic operators that ApsaraDB for OceanBase supports.

Supported arithmetic operators

Expression	Description	Example
+	The addition operator.	SELECT 2+3;
-	The minus operator.	SELECT 2-3;
*	The multiplication operator.	SELECT 2*3;

Expression	Description	Example
/	The division operator. The returned results are quotients. If the divisor is 0, NULL is returned.	SELECT 2/3;
% or MOD	The modulo operator. The returned results are remainders. If the divisor is 0, NULL is returned.	SELECT 2%3, 2 MOD 3;
^	Raises the specified number to the power of another number.	SELECT 2^2

If you need to use the addition (+), minus (-), and asterisk (*) operators to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for these operators. - INT FLOAT DOUBLE TIMESTAMP VARCHAR BOOL INT INT DOUBLE DOUBLE Error DOUBLE/Error Error FLOAT - DOUBLE DOUBLE Error DOUBLE/Error Error DOUBLE - - DOUBLE Error DOUBLE/Error Error TIMESTAMP - - - Error Error Error VARCHAR - - - - DOUBLE/Error Error BOOL - - - - Error

If you need to use the division (/) operator to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for the operator. If the divisor is zero, an error is returned.

- INT FLOAT DOUBLE TIMESTAMP VARCHAR BOOL NULL INT DOUBLE/Error DOUBLE/Error DOUBLE/Error Error DOUBLE/Error Error NULL/Error FLOAT - DOUBLE/Error DOUBLE/Error Error DOUBLE/Error Error NULL/Error DOUBLE - - DOUBLE/Error Error DOUBLE/Error Error NULL/Error TIMESTAMP - - - Error Error Error Error VARCHAR - - - - DOUBLE/Error Error NULL/Error BOOL - - - - Error Error NULL NULL

If you need to use the percent sign (%) and MOD operators to perform calculations, you must convert the data types of input values based on the specified rules. The following table describes the rules of converting the data types for these operators. - INT FLOAT DOUBLE TIMESTAMP VARCHAR BOOL NULL INT INT DOUBLE DOUBLE Error DOUBLE/Error Error NULL FLOAT - DOUBLE DOUBLE Error DOUBLE/Error Error NULL DOUBLE - - DOUBLE Error DOUBLE/Error Error NULL T IMESTAMP - - - Error Error Error Error VARCHAR - - - - DOUBLE/Error Error NULL/Error BOOL - - - - Error Error NULL NULL

18.5.8.4. Comparison operators

In ApsaraDB for OceanBase, operands are compared after they are converted into the values of the same type. You can use comparison operators to compare the operands. All the comparison operators return NULL or the values of the BOOLEAN data type. If the comparison result is true, 1 is returned. If the comparison result is false, 0 is returned. If the comparison result is unknown, NULL is returned.

Compare numeric values

You can use a comparison operator to compare two numeric values.

[Comparison operators](#) describes the comparison operators that ApsaraDB for OceanBase supports.

Comparison operators

Expression	Description	Example
=	Equal to	SELECT 1=0, 1=1, 1=NULL;
>=	Greater than or equal to	SELECT 1>=0, 1>=1, 1>=2, 1>=NULL;
>	Greater than	SELECT 1>0, 1>1, 1>2, 1>NULL;

Expression	Description	Example
<=	Less than or equal to	<code>SELECT 1<=0, 1<=1, 1<=2, 1<=NULL;</code>
<	Less than	<code>SELECT 1<0, 1<1, 1<2, 1<NULL;</code>
!= or <>	Not equal to	<code>SELECT 1!=0, 1!=1, 1<>0, 1<>1, 1!=NULL, 1<>NULL;</code>

If you need to use the comparison operators to perform calculations, you must convert the data types of the input values based on the specified rules. The following table describes the rules of converting the data types for the comparison operators.

- INT FLOAT DOUBLE TIMESTAMP VARCHAR BOOL NULL
 INT INT FLOAT DOUBLE Error ? INT Error NULL FLOAT - FLOAT DOUBLE Error ? FLOAT Error NULL DOUBLE - - DOUBLE Error ? DOUBLE Error NULL
 TIMESTAMP - - - TIMESTAMP ? TIMESTAMP Error NULL VARCHAR - - - - VARCHAR ? BOOL NULL BOOL - - - - - BOOL NULL NULL NULL

Note

- ? ApsaraDB for OceanBase attempts to convert the specified data types to the data types that start with question marks (?) before the comparisons are performed. If the type conversion fails, errors are reported.
- If you compare the values of the BOOLEAN data type, False is smaller than True.

[NOT] BETWEEN ... AND ...

Checks whether the specified value falls in or out of a range of values.

```
mysql> SELECT 2 BETWEEN 1 AND 2,
3 NOT BETWEEN 1 AND 2,
1 BETWEEN null AND 0,
1 NOT BETWEEN null AND 0\G;
***** 1. row *****
2 BETWEEN 1 AND 2: 1
3 NOT BETWEEN 1 AND 2: 1
1 BETWEEN null AND 0: 0
1 NOT BETWEEN null AND 0: 1
1 row in set (0.00 sec)
```

Example:

The following two tables are available: emp that stores the employee information and salgrade that stores the salary information. The employee information includes employee names and salaries. The salary information includes salary ranges and salary levels. In this example, you can execute the following statements to query the employee names, salaries, and salary levels.

```
mysql> select * from emp; //emp is the table that stores the employee information.
+-----+-----+
| ename | sal  |
+-----+-----+
| Jerry | 25000.00 |
| Larry | 40000.00 |
| Maggie | 46000.00 |
| Micky | 15000.00 |
+-----+-----+
4 rows in set (0.00 sec)
mysql> select * from salgrade; //salgrade is the table that stores the salary information.
+-----+-----+-----+
| grade | losal | hisal |
+-----+-----+-----+
| 1 | 10000.00 | 20000.00 |
| 2 | 20001.00 | 30000.00 |
| 3 | 30001.00 | 40000.00 |
| 4 | 40001.00 | 50000.00 |
| 5 | 50001.00 | 90000.00 |
+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> select a1.ename, a1.sal, a2.grade from emp a1, salgrade a2 where a1.sal between a2.losal and a2.hisal; //The
BETWEEN...AND... operator specifies the query criteria.
+-----+-----+-----+
| ename | sal  | grade |
+-----+-----+-----+
| Micky | 15000.00 | 1 |
| Jerry | 25000.00 | 2 |
| Larry | 40000.00 | 3 |
| Maggie | 46000.00 | 4 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

[NOT] IN

Checks whether the specified value falls within a specified range.

Example:

```
mysql> SELECT 2 IN (1, 2), 3 IN (1, 2)\G;
***** 1. row *****
2 IN (1, 2): 1
3 IN (1, 2): 0
1 row in set (0.00 sec)
```

IS [NOT] NULL | TRUE | FALSE | UNKNOWN

Checks whether a value is NULL, true, false, or unknown. If the operation is successful, TRUE or FALSE is returned. In this scenario, NULL is not returned.

Example:

```
mysql> SELECT 0 IS NULL,
NULL IS NULL, NULL IS TRUE,
(0>1) IS FALSE,
NULL IS UNKNOWN,
0 IS NOT NULL,
NULL IS NOT NULL,
NULL IS NOT TRUE,
(0>1) IS NOT FALSE,
NULL IS NOT UNKNOWN\G;
***** 1. row *****
      0 IS NULL: 0
      NULL IS NULL: 1
      NULL IS TRUE: 0
      (0>1) IS FALSE: 1
      NULL IS UNKNOWN: 1
      0 IS NOT NULL: 1
      NULL IS NOT NULL: 0
      NULL IS NOT TRUE: 1
      (0>1) IS NOT FALSE: 0
      NULL IS NOT UNKNOWN: 0
1 row in set (0.00 sec)
```

Vector comparison operators

Vector comparison operators compare two vectors or rows. The supported operators are `<`, `>`, `=`, `<=`, `>=`, `!=`, `<=>`, `IN`, and `NOT IN`. All these operators are binary operators. Each two vectors to be compared must have the same number of dimensions.

The `<=>` operator represents `NULL-safe equal`. The `<=>` operator is similar to the equal to (`=`) operator and performs equality comparisons. However, if both operands are `NULL`, the `<=>` operator returns 1 instead of `NULL`. If only one operand is `NULL`, the `<=>` operator returns 0 instead of `NULL`.

The expressions `(1,2)` and `ROW(1,2)` are also known as row constructors. The expressions `(1,2)` and `ROW(1,2)` are equivalent. The two expressions are valid in other contexts. For example, the following two statements are equivalent. However, only the second statement can be optimized.

```
SELECT * FROM t1 WHERE (column1, column2) = (1, 1);
SELECT * FROM t1 WHERE column1 = 1 AND column2 = 1;
```

If comparison results are returned after the first `i` scalars of each two operands are compared, vector comparison operators do not continue to compare the other numeric values of the operands. This is the main difference between vector comparison operators and general operators.

When you use vector comparison operators, pay attention to the following considerations:

- To compare `n`-tuple vectors, you can omit the `ROW` keyword. Note that `n` must be larger than or equal to three. For example, `ROW(1,2,3) < ROW(1,3,5)` is equivalent to `(1,2,3) < (1,3,5)`.
- The `IN` and `NOT IN` operators can be used only in vector operations. The `IN` operator indicates that the left parameter value is included in the right value set. The `NOT IN` operator indicates that the left parameter value is excluded from the right value set. Each right value set is enclosed in parentheses `()`. For example, you can specify `1 in (2, 3, 1)` for comparison.

- For `IN` or `NOT IN` operators, the scalar operands in the corresponding positions must be comparable. Otherwise, errors are reported. For example, `ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4))` is valid and `ROW(1,2) in (ROW(2,1), ROW(2,3), ROW(1,3,4))` is invalid.

Examples:

```
mysql> SELECT ROW(1,2) < ROW(1, 3),
-> ROW(1,2,10) < ROW(1, 3, 0),
-> ROW(1,null) < ROW(1,0),
-> ROW(null, 1) < ROW(null, 2),
-> ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4), ROW(4,5)),
-> 1 in (1,2,3),
-> 1 not in (2,3,4),
-> ROW(1,2) not in (ROW(2,1),ROW(2,3), ROW(3,4)),
-> NULL = NULL,
-> NULL <=> NULL,
-> NULL <=> 1,
-> 1 <=> 0 \G;
***** 1. row *****
      ROW(1,2) < ROW(1, 3): 1
      ROW(1,2,10) < ROW(1, 3, 0): 1
      ROW(1,null) < ROW(1,0): NULL
      ROW(null, 1) < ROW(null, 2): NULL
ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4), ROW(4,5)): 1
      1 in (1,2,3): 1
      1 not in (2,3,4): 1
      ROW(1,2) not in (ROW(2,1),ROW(2,3), ROW(3,4)): 1
      NULL = NULL: NULL
      NULL <=> NULL: 1
      NULL <=> 1: 0
      1 <=> 0: 0
1 row in set (0.00 sec)
```

18.5.8.5. Vector comparison operators

Vector comparison operators compare two vectors or rows. The supported operators are less than (`<`), greater than (`>`), equal to (`=`), less than or equal to (`<=`), greater than or equal to (`>=`), not equal to (`!=`), `<=>`, `IN`, and `NOT IN`. All these operators are binary operators. Each two vectors to be compared must have the same number of dimensions.

The `<=>` operator represents NULL-safe equal. The `<=>` operator is similar to the equal to the (`=`) operator and performs equality comparisons. However, if both operands are NULL, the `<=>` operator returns 1 instead of NULL. If only one operand is NULL, the `<=>` operator returns 0 instead of NULL.

The expressions `(1,2)` and `ROW(1,2)` are also known as row constructors. The expressions `(1,2)` and `ROW(1,2)` are equivalent. The two expressions are valid in other contexts.

For example, the following two statements are equivalent. However, only the second statement can be optimized.

```
SELECT * FROM t1 WHERE (column1, column2) = (1, 1);
```

```
SELECT * FROM t1 WHERE column1 = 1 AND column2 = 1;
```

Vector comparison operators differ from general operators in the following aspects:

- If comparison results are returned after the first i scalars of each two operands are compared, vector comparison operators do not continue to compare the other numeric values of the operands.
- When you use vector comparison operators, pay attention to the following considerations:
 - To compare n -tuple vectors, you can omit the ROW keyword. Note that n must be larger than or equal to three.

For example, `ROW(1,2,3) < ROW(1,3,5)` is equivalent to `(1,2,3) < (1,3,5)`.

- The IN and NOT IN operators can be used only in vector operations. The IN operator indicates that the left parameter value is included in the right value set. The NOT IN operator indicates that the left parameter value is excluded from the right value set. Each right value set is enclosed in parentheses ().

For example, you can specify `1 in (2, 3, 1)` for comparison.

- For IN or NOT IN operators, the scalar operands in the corresponding positions must be comparable. Otherwise, errors are reported.

For example, `ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4))` is valid and `ROW(1,2) in (ROW(2,1), ROW(2,3), ROW(1,3, 4))` is invalid.

Examples

```
mysql> SELECT ROW(1,2) < ROW(1, 3),
-> ROW(1,2,10) < ROW(1, 3, 0),
-> ROW(1,null) < ROW(1,0),
-> ROW(null, 1) < ROW(null, 2),
-> ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4), ROW(4,5)),
-> 1 in (1,2,3),
-> 1 not in (2,3,4),
-> ROW(1,2) not in (ROW(2,1),ROW(2,3), ROW(3,4)),
-> NULL = NULL,
-> NULL <=> NULL,
-> NULL <=> 1,
-> 1 <=> 0 \G;
***** 1. row *****
      ROW(1,2) < ROW(1, 3): 1
      ROW(1,2,10) < ROW(1, 3, 0): 1
      ROW(1,null) < ROW(1,0): NULL
      ROW(null, 1) < ROW(null, 2): NULL
ROW(1,2) in (ROW(1,2), ROW(2,3), ROW(3,4), ROW(4,5)): 1
      1 in (1,2,3): 1
      1 not in (2,3,4): 1
      ROW(1,2) not in (ROW(2,1),ROW(2,3), ROW(3,4)): 1
      NULL = NULL: NULL
      NULL <=> NULL: 1
      NULL <=> 1: 0
      1 <=> 0: 0
1 row in set (0.00 sec)
```

18.5.8.6. Bitwise operators

In ApsaraDB for OceanBase, bit operations are performed on the values of the BIGINT data type. BIGINT is a data type that stores 64-bit integers. For bitwise operators, the maximum number of bits for each value is 64.

Bitwise operators describes the bitwise operators that ApsaraDB for OceanBase supports.

Bitwise operators

Expression	Description	Example
BIT_COUNT(N)	Returns the number of bits that are specified by the parameter N.	SELECT BIT_COUNT(29);-> 4
&	The bitwise AND operator.	SELECT 29 & 15;-> 13 The returned value is an unsigned 64-bit integer.

Expression	Description	Example
~	Inverts all the bits.	<pre>SELECT 29 & ~15;-> 16</pre> <p>The returned value is an unsigned 64-bit integer.</p>
	The bitwise OR operator.	<pre>SELECT 29 ~15;-> 31</pre> <p>The returned value is an unsigned 64-bit integer.</p>
^	The bitwise XOR operator.	<pre>SELECT 1 ^ 1;-> 0</pre> <p>The returned value is an unsigned 64-bit integer.</p>
<<	Shifts the specified value of the BIGINT data type to the left by two bits.	<pre>SELECT 1 << 2;-> 4</pre> <p>The returned value is an unsigned 64-bit integer.</p>
>>	Shifts the specified value of the BIGINT data type to the right by two bits.	<pre>SELECT 4 << 2;-> 1</pre> <p>The returned value is an unsigned 64-bit integer.</p>

18.5.8.7. Operator precedences

If you need to use ApsaraDB for OceanBase operators to perform mixed operations, you must familiarize yourself with the operator precedences.

Operator precedences describes the operators that ApsaraDB for OceanBase supports. The following table lists the operators based on precedences in descending order.

Operator precedences

Precedence	Operator
15	!
14	- (unary minus) and ~
13	^
12	*, /, %, and MOD
11	+, -
10	<<, >>
9	&

Precedence	Operator
8	
7	= (comparison operator: equal to), <=>, >, >=, <, <=, <>, !=, IS, LIKE, REGEXP, and IN
6	BETWEEN
5	NOT
4	AND and &&
3	XOR
2	OR and
1	= (assignment operator)

 **Note** You can use parentheses () to enclose the operations that you want to perform before the other operations. This also helps you identify the operations that have high precedences.

18.5.9. Escape characters

An escape character is a character sequence that is prefixed with a backslash (\) in a string and invokes an alternative interpretation on the subsequent characters.

Escape characters are case-sensitive. For example, `\b` represents the backspace and `\B` represents the B character.

Escape characters describes the escape characters that can be recognized by ApsaraDB for OceanBase.

Escape characters

Escape character	Description
<code>\b</code>	The backspace.
<code>\f</code>	The form feed.  Note MySQL does not support this escape character.
<code>\n</code>	The line feed.
<code>\r</code>	The carriage return.
<code>\t</code>	The tab character.
<code>\\</code>	The backslash (\).
<code>\'</code>	The single quotation mark (').
<code>\"</code>	The double quotation mark (").
<code>_</code>	The underscore (_).
<code>\%</code>	The percent sign (%).
<code>\0</code>	The NULL character.

Escape character	Description
\Z	<p>The ASCII 26 character. The corresponding key combination is Ctrl+Z.</p> <p>In Windows, ASCII 26 is the end-of-file indicator. You can encode the ASCII 26 character as \Z so that ASCII 26 is not interpreted as the end-of-file indicator.</p>

18.5.10. DDL statements

18.5.10.1. Overview

Data definition language (DDL) statements allow you to manage basic database components. For example, you can execute DDL statements to create, modify, and delete tables.

ApsaraDB for OceanBase supports a wide range of DDL statements. For example, the following statements are supported: CREATE DATABASE , ALTER DATABASE , DROP DATABASE , CREATE TABLE , DROP TABLE , ALTER TABLE , CREATE INDEX , DROP INDEX , CREATE VIEW , DROP VIEW , ALTER VIEW , TRUNCATE TABLE .

18.5.10.2. CREATE DATABASE

Syntax

```
CREATE DATABASE [IF NOT EXISTS] dbname
    [create_specification_list];

create_specification_list:
    create_specification [create_specification...]

create_specification:
    [DEFAULT] CHARACTER SET [=] charsetname
    | [DEFAULT] COLLATE [=] collationname
    | REPLICAS [=] num
    | PRIMARY_ZONE [=] zone
    | DEFAULT TABLEGROUP [=] {NULL | tablegroupname}
```

The CREATE DATABASE statement allows you to create a database. When you create the database, you can specify the default attributes for the database, such as the default character set and collation.

Notes:

- REPLICAS specifies the number of replicas.
- PRIMARY_ZONE specifies the primary zone of the database. Before you specify this attribute, make sure that the primary zone is included in the zone list of the tenant.
- DEFAULT TABLEGROUP specifies the default table group of the database. If you do not specify this parameter, the default value NULL is used.

Examples

```
root@(none) 01:36:27>create database test2 default CHARACTER SET UTF8;
Query OK, 1 row affected (0.00 sec)
root@(none) 01:36:44>create database test3 CHARACTER SET UTF8;
Query OK, 1 row affected (0.00 sec)
```

Errors

- If the database that you want to create already exists and you do not add `IF NOT EXISTS` to the statement, the system returns the following error: `ERROR 1007 (HY000): Can't create database 'test3'; database exists` .
- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax` .
- If the length of the specified database name exceeds the maximum length, the system returns the following error: `ERROR 1059 (42000): Identifier name 'XXXX' is too long` .

18.5.10.3. ALTER_DATABASE

Syntax

```
ALTER DATABASE [dbname]
    alter_specification_list;

alter_specification_list:
    [SET] alter_specification [alter_specification...]

alter_specification:
    [DEFAULT] {CHARACTER SET | CHARSET} [=] charsetname
    | [DEFAULT] COLLATE [=] collationname
    | REPLICA_NUM [=] num
    | PRIMARY_ZONE [=] zonenumber
    | {READ ONLY | READ WRITE}
    | DEFAULT TABLEGROUP [=] {NULL | tablegroupname}
```

The `ALTER_DATABASE` statement allows you to change the attributes of a specified database. For example, you can change the following attributes: character set, collation, the number of replicas, primary zone, database permissions, and default table group. `REPLICA_NUM` specifies the number of replicas. `PRIMARY_ZONE` specifies the primary zone. `READ ONLY| READ WRITE` specifies whether the database is read-only. `READ ONLY` indicates that users can have only read access to the database, and `READ WRITE` indicates that users can have read and write access to the database. `DEFAULT TABLEGROUP` specifies the default table group of the database. If the value is `NULL`, the default table group is deleted.

The database name is optional. If you do not specify the database name, the statement takes effect on the current database.

Examples

```
alter database test2 DEFAULT CHARACTER SET UTF8;
```

Errors

- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax .`
- If the specified database name is invalid or the database does not exist, the system returns the following error: `ERROR 1049 (42000): Unknown database .`

```
mysql> alter database notest default character set utf8;
ERROR 1049 (42000): Unknown database
```

18.5.10.4. DROP DATABASE

Syntax

```
DROP DATABASE [IF EXISTS] dbname;
```

The `DROP DATABASE` statement allows you to drop all the tables in a specified database and delete the database.

If you add `IF EXISTS` to the statement and the specified database does not exist, no errors are reported.

Examples

```
mysql> drop database notest;
ERROR 1008 (HY000): Can't drop database 'notest'; database doesn't exist
mysql> drop database if exists notest;
Query OK, 0 rows affected, 1 warning (0.00 sec)
mysql> show warnings;
+-----+-----+-----+-----+
| Level | Code | Message                               |      |
+-----+-----+-----+-----+
| Note  | 1008 | Can't drop database 'notest'; database doesn't exist |      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Errors

- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax .`
- If the specified database does not exist, the system returns the following error: `ERROR 1008 (HY000): Can't drop database 'XXX'; database doesn't exist .`

18.5.10.5. CREATE TABLE

The `CREATE TABLE` statement allows you to create a table in a specified ApsaraDB for OceanBase database.

Syntax

```
CREATE TABLE [IF NOT EXIST] tblname
(create definition )
```

```

[create_definition,...]
[table_options]
[partition_options];

CREATE TABLE [IF NOT EXISTS] tblname
LIKE oldtblname

create_definition:
colname column_definition
| PRIMARY KEY (index_col_name [, index_col_name...]) [index_type] [index_options]...
  | {INDEX|KEY} [indexname] (index_col_name,...) [index_type] [index_options]...
  | UNIQUE [INDEX|KEY] [indexname] (index_col_name,...) [index_type] [index_options]...
| FULLTEXT [INDEX|KEY] [indexname] (index_col_name,...) CTXCAT(index_col_name,...) [index_options]...

column_definition:
  data_type [NOT NULL | NULL] [DEFAULT defaultvalue]
  [AUTO_INCREMENT] [UNIQUE [KEY]] | [[PRIMARY] KEY]
  [COMMENT 'string']
| [data_type] [GENERATED ALWAYS] AS (expression)
  [VIRTUAL | STORED] [UNIQUE [KEY]] [COMMENT comment]
  [NOT NULL | NULL] [[PRIMARY] KEY]

data_type:
TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME[(fsp)]
| TIMESTAMP[(fsp)]
| DATETIME[(fsp)]
| YEAR
| CHAR[(length)]
  [CHARACTER SET charsetname] [COLLATE collationname]
| VARCHAR(length)
  [CHARACTER SET charsetname] [COLLATE collationname]
| BINARY[(length)]
| VARBINARY(length)

index_col_name:

```

```

colname [(length)] [ASC | DESC] ApsaraDB for OceanBase does not support prefix indexes.[(length)]

index_type:
    USING BTREE

index_options:
    index_option [index_option...]

index_option:
GLOBAL [LOCAL]
    |COMMENT 'string'
    |COMPRESSION [=] {NONE | LZ4_1.0 | LZO_1.0 | SNAPPY_1.0 | ZLIB_1.0}
    |BLOCK_SIZE [=] size
    |STORING(columnname_list)
| VISIBLE [INVISIBLE]

columnname_list:
    colname [, colname...]

table_options:
    table_option [table_option]...

table_option:
[DEFAULT] {CHARACTER SET| CHARSET} [=] charsetname
| [DEFAULT] COLLATE [=] collationname
| COMMENT [=] 'string'
| COMPRESSION [=] {NONE | LZ4_1.0 | LZO_1.0 | SNAPPY_1.0 | ZLIB_1.0}
| EXPIRE_INFO [=] expr
| REPLICANUM [=] num
| TABLE_ID [=] id
| BLOCK_SIZE [=] size
| USE_BLOOM_FILTER [=] {True | False}
| STEP_MERGE_NUM [=] num
    | TABLEGROUP [=] 'tablegroupname'
| PRIMARY_ZONE [=] zonelist
| AUTO_INCREMENT [=] num
| PCTFREE [=] integer
    | LOCALITY [=] locality

partition_options:
    PARTITION BY
    HASH(expr)
    |KEY(column_list)
    PARTITIONS num
    [partition_definition ...]

partition_definition:

```

```
COMMENT [=] 'string'
```

 **Note** B-tree indexes are created for the data that is stored in ApsaraDB for OceanBase. The data is sorted based on primary keys. In ApsaraDB for OceanBase, you do not need to specify primary keys because the system automatically generates the primary keys.

The `CREATE TABLE` statement supports the `UNIQUE` constraint. The `CREATE TABLE` statement does not support temporary tables or the `CHECK` constraint. When you execute the statement to create a table, you cannot import data from other tables to the table that you want to create.

Notes:

- If you add `IF NOT EXISTS` to the statement and the table that you want to create already exists, the system does not return an error. If you do not add `IF NOT EXISTS` to the statement, the system returns an error.
- For more information about the `data_type` parameter, see [Data types](#).
- `NOT NULL`, `DEFAULT`, and `AUTO_INCREMENT` are used to implement integrity constraints on columns.
- [Parameters of table_option](#) describes the `table_option` parameters. The parameters are separated with commas (,).

Parameters of table_option

Parameter	Description	Example
CHARACTER SET	The character set that is used to encode the strings in the table. The character set is used to provide the metadata information. Set the value to utf8mb4.	<code>CHARACTER SET = 'utf8mb4'</code>
COMMENT	The description of the table.	<code>COMMENT='create by Bruce'</code>
TABLE_ID	The ID of the table. If the specified table ID is smaller than 50,000, turn on the <code>enable_sy</code> switch of the <code>s_table_ddl</code> of the RootServer. If you are a general user, we recommend that you do not specify the table ID.	<code>TABLE_ID =4000</code>
BLOCK_SIZE	The micro-block size of a partition.	The default size is 16 KB.
USE_BLOOM_FILTER	Specifies whether to use a Bloom filter when the data in the table is read. Valid values: <ul style="list-style-type: none"> ◦ False: indicates that the Bloom filter is not used. False is the default value. ◦ True: indicates that the Bloom filter is used. 	<code>USE_BLOOM_FILTER = False</code>
TABLEGROUP	The table group to which the table belongs.	-

Parameter	Description	Example
REPLICA_NUM	The number of replicas for the partitions in the table. The default value is 3.	<code>REPLICA_NUM = 3</code>
ZONE_LIST	The list of clusters.	-
PRIMARY_ZONE	The primary zone.	-
AUTO_INCREMENT	The start value for an auto-increment field.	<code>AUTO_INCREMENT = 5</code>
PCTFREE	The percentage of the idle space that is reserved in a database macro-block.	<p>When you use the parameter, pay attention to the following syntax details:</p> <ul style="list-style-type: none"> In the Oracle syntax, the PCTFREE parameter specifies the similar setting. To ensure compatibility with the Oracle syntax, ApsaraDB for OceanBase uses the same parameter name. Use the following syntax to create a table: <pre>CREATE TABLE table_name (column_definition) PCTFREE [=] integer</pre> Use the following syntax to modify a table: <pre>ALTER TABLE table_name PCTFREE [=] integer</pre> <p>Note</p> <ul style="list-style-type: none"> If you do not specify the PCTFREE parameter when you create a table, the default value 10 is used. The specified value of the PCTFREE parameter must be an integer. The valid value range is [0,50).

- In `index_option`, you can use the GLOBAL keyword to specify global indexes and use the LOCAL keyword to specify local indexes. By default, global indexes are used. When you create a partitioned table, you must specify the LOCAL keyword to use local indexes. If you do not specify the LOCAL keyword, the system returns

an error.

Examples

The following example is used to illustrate how to create a database table and view the table information.

1. Execute the following statement to create a database table:

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(50)) REPLICAS_NUM = 3, PRIMARY_ZONE = 'zone1';
```

You can also execute the following statement:

```
CREATE TABLE test (c1 int, c2 VARCHAR(50), primary key(c1)) REPLICAS_NUM = 3, PRIMARY_ZONE = 'zone1';
```

2. Execute the following statement to view the table information:

```
SHOW tables;
DESCRIBE test;
```

Errors

- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax .`
- If the table name already exists, the system returns the following error: `ERROR 1050 (42501): Table 'test' already exists .`
- If the length of the specified table name exceeds the maximum length, the system returns the following error: `ERROR 1059 (42000): Identifier name 'XXXX' is too long .`
- If duplicate partition names are found in the table to be created, the system returns the following error: `ERROR 1517 (HY000): Duplicate partition name XX .` The following example is provided to explain the error:

```
mysql> create table employeetest(id int) partition by range(id) (partition dPname values less than (10), partition dPname values less than (20));
ERROR 1517 (HY000): Duplicate partition name 'dPname'
```

18.5.10.6. ALTER TABLE

The ALTER TABLE statement allows you to update the schema of an existing table. For example, you can modify an existing table and table attributes, add columns, modify columns and column attributes, or delete columns.

Syntax

```
ALTER TABLE tblname
alter_specification [, alter_specification]...

alter_specification:
  ADD [COLUMN] colname column_definition
  | ADD [COLUMN] (colname column_definition,...)
  | ADD {INDEX | KEY} [indexname] (index_col_name,...) [index_tpye] [index_options]
  | ADD [CONSTRAINT [symbol]]
      PRIMARY KEY (index_col_name,...) [index_type] [index_options] #ApsaraDB for OceanBase does not support the AD
D PRIMARY KEY clause.
  | ADD [CONSTRAINT [symbol]]
      UNIQUE [INDEX | KEY] [indexname] (index_col_name,...) [index_type] [index_options]
```

```

| ALTER [COLUMN] colname {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] oldcolname newcolname column_definition [FIRST|AFTER col_name]
| MODIFY [COLUMN] colname column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] colname
| DROP PRIMARY KEY#ApsaraDB for OceanBase does not support the DROP PRIMARY KEY clause. You cannot use the
DROP PRIMARY KEY clause to drop a primary key.
| DROP {INDEX | KEY} indexname
| RENAME [TO | AS] newtblname
| ORDER BY colname
| CONVERT TO CHARACTER SET charsetname [COLLATE collationname]
| [DEFAULT] CHARACTER SET charsetname [COLLATE collationname]
| table_options
| partition_options
| ADD PARTITION partition_definition
| DROP PARTITION partition_names
| COALESCE PARTITION number
| REORGANIZE PARTITION partition_names INTO (partition_definitions)
| ANALYZE PARTITION partition_names
| CHECK PARTITION partition_names
| OPTIMIZE PARTITION partition_names
| REBUILD PARTITION partition_names
| REPAIR PARTITION partition_names
| DROP TABLEGROUP
| AUTO_INCREMENT [=] num

```

column_definition:

```

data_type [NOT NULL | NULL] [DEFAULT defaultvalue]
[AUTO_INCREMENT] [UNIQUE [KEY]] [[PRIMARY] KEY]
[COMMENT 'string']

```

table_options:

```
[SET] table_option [table_option]...
```

table_option:

```

[DEFAULT] {CHARACTER SET | CHARSET} [=] charsetname
| [DEFAULT] COLLATE [=] collationname
| COMMENT [=] 'string'
| COMPRESSION [=] {NONE | LZ4_1.0 | LZO_1.0 | SNAPPY_1.0 | ZLIB_1.0}
| EXPIRE_INFO [=] expr
| REPLICA_NUM [=] num
| TABLE_ID [=] id
| BLOCK_SIZE [=] size
| USE_BLOOM_FILTER [=] {True| False}
| STEP_MERGE_NUM [=] num
| TABLEGROUP [=] tablegroupname

```

```
| PRIMARY_ZONE [=] zoneid
| AUTO_INCREMENT [=] num
  | PCTFREE [=] integer
| {READ ONLY | READ WRITE}
  | LOCALITY [=] locality
```

partition_options:

```
PARTITION BY
  HASH(expr)
  | KEY(column_list)
  [PARTITIONS num]
  [partition_definition ...]
```

partition_definition:

```
COMMENT [=] 'commenttext'
```

ApsaraDB for OceanBase does not support the `alter_specification` clauses of the `ALTER TABLE` statement. In ApsaraDB for OceanBase, the table ID in `table_option` cannot be changed. However, if you attempt to change the table ID, the system does not return syntax errors. ApsaraDB for OceanBase supports the `alter_specification` clauses that are frequently used.

Add columns

```
ALTER TABLE tblname
  ADD [COLUMN] col_name column_definition;
```

For more information about the `data_type` parameter, see [Data types](#).

ApsaraDB for OceanBase does not allow you to add primary key columns.

Modify column attributes

```
ALTER TABLE tblname
  ALTER [COLUMN] colname
  [SET DEFAULT literal| DROP DEFAULT];
```

Drop columns

```
ALTER TABLE tblname
  DROP [COLUMN] colname;
```

ApsaraDB for OceanBase does not allow you to drop the primary key columns or the columns that have indexes.

Rename tables

```
ALTER TABLE tblname
  RENAME [TO] newtblname;
```

Rename columns

```
ALTER TABLE tblname  
    CHANGE [COLUMN] oldcolname newcolname column_definition;
```

 **Notice** For the columns of the VARCHAR data type, the values of only the VARCHAR data type can be increased. The other data types are not supported.

For example, execute the following statement to change the field name d in the t2 table to c and change the data type of the field.

```
ALTER TABLE t2 CHANGE COLUMN d c CHAR(10);
```

Specify the block size of a partitioned table

```
ALTER TABLE tblname  
    SET BLOCK_SIZE [=] blocksize;
```

Specify the number of replicas for a table

```
ALTER TABLE tblname  
    SET REPLICA_NUM [=] num;
```

REPLICA_NUM specifies the number of replicas for the specified table.

Specify the compression method for a table

```
ALTER TABLE tblname  
    SET COMPRESSION [=] '{NONE | LZ4_1.0 | LZ0_1.0 | SNAPPY_1.0 | ZLIB_1.0}';
```

Specify whether to use a Bloom filter

```
ALTER TABLE tblname  
    SET USE_BLOOM_FILTER [=] {True | False};
```

Add descriptions

```
ALTER TABLE tblname  
    SET COMMENT [=] 'commentstring';
```

Specify the number of macro-blocks that are to be merged at a time for progressive compaction

```
ALTER TABLE tblname  
    SET PROGRESSIVE_MERGE_NUM [=] num;
```

You can execute this statement to specify the number of macro-blocks that are to be merged at a time for progressive compaction. The value of the `PROGRESSIVE_MERGE_NUM` parameter ranges from 1 to 64.

Specify the zone for a table

```
ALTER TABLE tblname
  zone_specification...;
zone_specification:
  PRIMARY_ZONE [=] zone
```

Examples

Example 1:

1. Before you add columns, execute the following statement to view the table information. **Table information before you add columns** shows the table information:

```
DESCRIBE test;
```

Table information before you add columns

Field	Type	Null	Key	Default	Extra
c1	int(11)	NO	PRI	NULL	
c2	varchar(50)	YES		NULL	

2 rows in set (0.01 sec)

2. Execute the following statement to add the c3 column:

```
ALTER TABLE test ADD c3 int;
```

3. After you add the column, execute the following statement to view the table information that is shown in **Table information after you add the column**:

```
DESCRIBE test;
```

Table information after you add the column

Field	Type	Null	Key	Default	Extra
c1	int(11)	NO	PRI	NULL	
c2	varchar(50)	YES		NULL	
c3	int(11)	YES		NULL	

3 rows in set (0.02 sec)

4. Execute the following statement to delete the c3 column:

```
ALTER TABLE test DROP c3;
```

5. After you delete the column, execute the following statement to view the table information that is shown in **Table information after you delete the column**:

```
DESCRIBE test;
```

Table information after you delete the column

```

+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| c1    | int(11)       | NO   | PRI | NULL    |       |
| c2    | varchar(50)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
    
```

Example 2:

```
ALTER TABLE test SET REPLICA_NUM=2, ADD COLUMN c5 INT;
```

Errors

- If a syntax error occurs, the system returns the following error: **ERROR 1064 (42000): You have an error in your SQL syntax** .
- If the specified table does not exist, the system returns the following error: **ERROR 1146 (42S02): Table 'XXX' doesn't exist** .
- If you add primary key columns to an existing table, the system returns the following error: **ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table** .

```

mysql> select * from employees;
+-----+-----+-----+-----+-----+-----+
| id | frame | lname | hired   | separated | job_code | store_id |
+-----+-----+-----+-----+-----+-----+
| 4 | 4 | 4 | 2000-04-04 | 2044-04-04 | 1 | 4 |
| 5 | 5 | 5 | 2000-05-05 | 2022-05-05 | 123 | 5 |
| 8 | 8 | 8 | 2000-05-05 | 2001-08-08 | 123 | 9 |
| 2 | test | 2 | 2000-02-02 | 2024-02-02 | 2 | 2 |
| 7 | 7 | 7 | 1999-02-02 | 2007-07-07 | 7 | 10 |
| 3 | 3 | 3 | 2000-03-03 | 2034-03-03 | 3 | 3 |
| 11 | test | test | 2003-03-03 | 2003-05-05 | 11 | 11 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> alter table employees add id2 int primary key;
ERROR 1068 (42000): Multiple primary key defined
    
```

- If the column that you want to delete does not exist, the system returns the following error: **ERROR 1091 (42000): Can't DROP 'XXX'; check that column/key exists** . The following example is provided to explain the error.

```

mysql> alter table employees drop id3;
ERROR 1091 (42000): Can't DROP 'id3'; check that column/key exists
    
```

- If the specified data type does not meet the requirements of the default data length, the system returns the following error: **ERROR 1067 (42000): Invalid default value for 'XXX'** . The following example is provided to explain the error.

```
mysql> alter table test1 add colum1 VARCHAR(10) default 'tttttttttttttttttttt';
ERROR 1067 (42000): Invalid default value for 'column1'
```

18.5.10.7. DROP TABLE

The **DROP TABLE** statement allows you to drop tables that are stored in an ApsaraDB for OceanBase database.

Syntax

```
DROP TABLE [IF EXISTS] tbl_list;
```

tbl_list:

```
tblname [, tblname ...]
```

If you add **IF EXISTS** to the statement and the table that you want to delete does not exist, the system does not return an error. If you do not add **IF EXISTS** to the statement, the system returns an error.

If you need to drop multiple tables at a time, separate the tables with commas (,).

Examples

```
DROP TABLE IF EXISTS test;
```

Errors

- If a syntax error occurs, the system returns the following error: **ERROR 1064 (42000): You have an error in your SQL syntax** .
- If the table that you want to delete does not exist, the system returns the following error: **ERROR 1051 (42S02): Unknown table 'XXXX'** .

18.5.10.8. CREATE INDEX

Indexes are created on tables to sort the values of one or more columns of the tables. Indexes are used to reduce the response time of queries and reduce the performance overhead of database systems.

Note

1. In ApsaraDB for OceanBase, a new index takes effect only after a daily major freeze operation is performed. For example, you can create a unique index. Before the unique index takes effect, you can insert indexes that violate the uniqueness constraint. After the major freeze operation is performed, the unique index cannot take effect because duplicate indexes exist. In this scenario, the system changes the status of the unique index to `index_error`.
2. You can execute the `SHOW INDEX from <table>` statement to view the index status.

Syntax

```

CREATE [UNIQUE] INDEX indexname
    ON tblname (index_col_name,...)
    [index_type] [index_options]
| CREATE FULLTEXT INDEX indexname ON tblname (index_col_name,...) CTXCAT(index_col_name,...) [index_options]
index_type:
    USING BTREE

index_options:
    index_option [index_option...]

index_option:
    GLOBAL [LOCAL]
    | COMMENT 'string'
    | COMPRESSION [=] {NONE | LZ4_1.0 | LZ0_1.0 | SNAPPY_1.0 | ZLIB_1.0}
    | BLOCK_SIZE [=] size
    | STORING(columnname_list)

index_col_name:
    colname [(length)] [ASC | DESC]

columnname_list:
    colname [, colname...]

```

Notes:

- In `index_col_name`, you can specify ASC or DESC for each column name. ASC indicates that the values are sorted in ascending order and DESC indicates that the values are sorted in descending order. By default, the values are sorted in ascending order.
- In the CREATE INDEX statement, the indexes are first sorted based on the values of the first column in `index_col_name`. If the values in the first column are the same, the indexes are sorted based on the values in the next column. Similar rules apply to the other columns.
- You can execute the `SHOW INDEX FROM tblname` statement to view the created indexes.
- In `index_option`, you can use the GLOBAL keyword to specify global indexes and use the LOCAL keyword to specify local indexes. By default, `global indexes` are used.

When you create a partitioned table, you must specify the LOCAL keyword to use local indexes. If you do not specify the LOCAL keyword, the system returns an error. Separate the `index option` parameters with spaces.

- STORING is an optional field. This field indicates that the specified columns are stored in the index table for redundant storage. This improves the query performance of systems. The STORING field is available only in ApsaraDB for OceanBase.

 **Note** B-tree indexes are created for the data that is stored in ApsaraDB for OceanBase.

Examples

1. Execute the following statement to create a table that is named test:

```
CREATE TABLE test (c1 int primary key, c2 VARCHAR(10));
```

2. Execute the following statement to create indexes on the test table:

```
CREATE INDEX test_index ON test (c1, c2 DESC);
```

3. Execute the following statement to view the indexes of the test table:

```
SHOW INDEX FROM test;
```

Errors

- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax` .
- If the specified table name is invalid, the system returns the following error: `ERROR 1146 (42S02): Table 'XXX' doesn't exist` .
- If the length of the specified table name exceeds the maximum length, the system returns the following error: `ERROR 1059 (42000): Identifier name 'XXX' is too long` .
- If the specified column name is invalid, the system returns the following error: `ERROR 1072 (42000): Key column 'XXX' doesn't exist in table` .

Expression indexes

ApsaraDB for OceanBase allows you to use the specified syntax to create generated columns. Generated columns are divided into virtual columns and stored columns. Generated columns can be used to create indexes in the same way as general columns.

If INSERT, REPLACE, and UPDATE operations are performed on virtual columns, the values in the virtual columns are materialized into index tables. Stored columns are the same as general columns except that the values of the stored columns are obtained from expressions.

Prefix indexes

You can create indexes on the prefixes of string values that are stored in columns. You can use the `col_name(length)` syntax to specify the prefix length.

Notes:

- You can create prefix indexes on the columns of the CHAR, VARCHAR, BINARY, or VARBINARY data type.
- You must specify prefixes if you need to create indexes on the columns of the BLOB or TEXT data type.

FULLEXT CTXCAT indexes

ApsaraDB for OceanBase supports FULLTEXT CTXCAT indexes. The full-text indexes of this type can cover the columns that are used for text analysis and the columns that are not used for text analysis. General indexes can cover only the columns that are used for text analysis. You can use the following syntax to create FULLTEXT CTXCAT indexes.

```
CREATE FULLTEXT INDEX indexname ON tblname (index_col_name,...) CTXCAT(index_col_name,...) [index_options]
```

where:

- `tblname (index_col_name,...)` specifies the table columns on which the index is created. It also specifies the ordinal positions of the columns in the index.
- `CTXCAT(index_col_name,...)` specifies the columns that are used for full-text analysis.

In ApsaraDB for OceanBase, multiple columns can be used for full-text analysis. If multiple columns are used for full-text analysis, the text set for full-text analysis is the sum of the text sets that are stored in the corresponding columns. The OR logical operator is used to combine the text sets that are stored in the corresponding columns. Only the columns of the STRING data type can be used for full-text analysis.

In the column list for the index, you must declare the columns that are specified for full-text analysis if you execute a DDL statement to create a CTXCAT index. The ordinal positions of the specified columns that are used for full-text analysis must be continuous. The following examples are provided to explain these rules:

```
create table t1(a int primary key, b varchar(100), c varchar(100), d int);
create fulltext index i1 on t1(b, d) ctxcat(c);
ERROR 1072 (42000): Key column 'c' doesn't exist in table
```

In the preceding example, the specified column for full-text analysis is not included in the column list of the index.

```
create table t1(a int primary key, b varchar(100), c varchar(100), d int);
create fulltext index i1 on t1(b, d, c) ctxcat(b, c);
ERROR 5291 (HY000): The CTXCAT column must be contiguous in the index column list
```

In the column list of the index, the ordinal positions of the specified columns for full-text analysis are not continuous in the preceding example.

For more information about how to use `FULLTEXT CTXCAT` indexes, see `MATCH... AGAINST` in [Full-text search functions](#).

Invisible indexes

You can use invisible indexes in many scenarios. One example of the scenarios is that you do not want the optimizer to use an index. Another example is that an index has not been used by queries and is always in the idle state. However, you do not want to delete the index because the index may be used in subsequent queries or the risks of deleting the index cannot be assessed. In these scenarios, `invisible indexes` help you handle the challenges. You can mark the index as an invisible index so that the index is not used by the optimizer.

When you create an index, you can specify whether the index is an invisible index. The default status is visible. If you do not specify the index status, the default status is used.

You can specify whether indexes are invisible when you execute DDL statements to create tables and indexes or create only indexes. The following examples show how to specify the index status.

```
CREATE TABLE t1(c1 int primary key, c2 int, c3 int, c4 varchar(16), key idx1(c1) visible, index idx2(c2) invisible, unique key idx3(c3) visible, unique index idx4(c1, c2) invisible, unique idx5(c2,c3));
CREATE UNIQUE INDEX idx6 ON t1(c2) visible;
CREATE INDEX idx7 ON t1(c3) invisible;
CREATE INDEX idx8 ON t1(c4) ;
```

In the preceding example, the status of `idx1`, `idx3`, and `idx6` is specified as visible and the status of `idx2`, `idx4`, and `idx7` is specified as invisible. The status of `idx5` and `idx8` is not specified. Therefore, the default visible status is used for `idx5` and `idx8`.

You can also use the `ALTER TABLE ALTER INDEX` syntax to change the status of the created indexes.

`ALTER TABLE t1 ALTER INDEX idx8 invisible;` changes the status of the `idx8` index from visible to invisible. As a result, the optimizer cannot use the index.

18.5.10.9. DROP INDEX

High overheads are required to maintain a large number of indexes. We recommend that you drop the indexes that are not required.

 **Note** After you execute the DROP INDEX statement to drop the index, the index cannot be immediately dropped. You must wait for a period before the index is dropped.

Syntax

```
DROP INDEX indexname
ON tblname;
```

Examples

```
DROP INDEX test_index ON test;
```

Errors

- If a syntax error occurs, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax` .
- If the specified table name is invalid, the system returns the following error: `ERROR 1146 (42502): Table 'XXX' doesn't exist` .
- If the length of the specified table name exceeds the maximum length, the system returns the following error: `ERROR 1059 (42000): Identifier name 'XXX' is too long` .
- If the specified index is invalid, the system returns the following error: `ERROR 1091 (42000): Can't DROP 'XXX'; check that column/key exists` .

18.5.10.10. CREATE VIEW

Syntax

```
CREATE [OR REPLACE] VIEW viewname
[(column_list)] AS select_stmt;
```

The CREATE VIEW statement allows you to create a view. If you add the `OR REPLACE` clause to the statement, you can execute the statement to replace an existing view.

In the syntax, `select_stmt` specifies a SELECT statement. The SELECT statement defines the view by selecting data from base tables or other views.

In views, column names must be unique. This rule for views is the same as that for base tables. By default, the column names that are returned by the SELECT statement are used as the column names for the view. To specify the column names of the view, use the optional `column_list` clause. You can use this clause to specify the column names. The specified column names are separated with commas (,). The number of the column names in the `column_list` clause must be equal to the number of columns that are returned by the SELECT statement.

The columns that are returned by the SELECT statement can be the specified columns of base tables. The columns that are returned by the SELECT statement can also store the computing results of the expressions that use functions, constant values, or operators.

Views are not stored as physical tables in databases. Views are generated each time you send a request to access the views. Views are created based on the outputs of the SELECT statements that are specified in the `CREATE VIEW` statements.

ApsaraDB for OceanBase supports only the views that cannot be updated.

18.5.10.11. DROP VIEW

Syntax

```
DROP VIEW [IF EXISTS]
viewname [, viewname ...] ;
```

The `DROP VIEW` statement allows you to delete one or more views. To drop the views, you must have the `DROP` permission on each view.

If you add `IF EXISTS` to the statement and the specified views do not exist, no errors are reported.

18.5.10.12. ALTER VIEW

Syntax

```
ALTER VIEW viewname [(column_list)]
[(column_list)] AS select_statement
```

The `ALTER VIEW` statement allows you to change the definition of an existing view. The syntax of the statement is similar to that of the `CREATE VIEW` statement.



Notice

- ApsaraDB for OceanBase does not support the `ALTER VIEW` statement. This indicates that views cannot be modified.
- Views are logical tables. If you need to modify views, you can delete the views and create new views based on your business needs.

18.5.10.13. TRUNCATE TABLE

Syntax

```
TRUNCATE [TABLE] tblname;
```

The `TRUNCATE TABLE` statement allows you to delete a specified table and retain the table schema that includes the partition information about the table. The `TRUNCATE TABLE` statement implements the same logic as the `DELETE FROM` statement. You can execute the `DELETE FROM` statement to delete all the rows in a table. To execute the `TRUNCATE TABLE` statement, you must have the permissions to create and delete tables. The `TRUNCATE TABLE` statement is a DDL statement.

The `TRUNCATE TABLE` and `DELETE FROM` statements have the following differences:

- The `TRUNCATE TABLE` statement deletes and recreates the table. The response time for the `TRUNCATE TABLE` statement is shorter than that for the `DELETE FROM` statement. This is because the `DELETE FROM` statement deletes rows one after one.
- The output of the `TRUNCATE TABLE` statement shows that the number of affected rows is always 0.
- If you execute the `TRUNCATE TABLE` statement, each auto-incremented value is reset to the start value. The table manager does not store the latest auto-incremented value.
- You cannot execute the `TRUNCATE TABLE` statement when a transaction is processed or the table is locked. If you execute the statement in these scenarios, the system returns errors.
- If the file that defines the table is valid, you can execute the `TRUNCATE TABLE` statement to recreate the table as an empty table. This occurs even if the data or the index file has been corrupted.

In ApsaraDB for OceanBase, the specified table in the `TRUNCATE TABLE` statement can have only one partition.

ApsaraDB for OceanBase supports the `ALTER TABLE TRUNCATE PARTITION` statement. This allows you to implement distributed transactions. This occurs only if you can execute the `TRUNCATE TABLE` statement on tables that have multiple partitions in the later versions of ApsaraDB for OceanBase.

18.5.10.14. RENAME TABLE

Syntax

```
RENAME TABLE tblname TO newtblname  
[, tblname2 TO newtblname ...];
```

The `RENAME TABLE` statement allows you to rename one or more tables.

After you execute the `RENAME TABLE` statement, the specified tables are automatically renamed. When the tables are renamed, other threads cannot read data from the tables.

Examples

For example, if you have a source table that is named `oldtable`, you can create an empty table that has the same schema as the source table. You can name the empty table `newtable` and replace the source table with this empty table:

```
CREATE TABLE newtable(...);  
RENAME TABLE oldtable TO backuptable, newtable TO oldtable;
```

If you execute this statement to rename multiple tables, the tables are renamed based on the left-to-right order.

If you need to swap the names of two tables and the `temptable` table does not exist, execute the following statement:

```
RENAME TABLE oldtable TO temptable,  
newtable TO oldtable,  
temptable TO newtable
```

For the same tenant, you can rename database tables and move the tables from the databases to other databases.

```
RENANME TABLE currentdb.tblname TO otherdb.tblname
```

Before you execute the `RENAME` statement to rename tables, make sure that no tables are locked or involved in active transactions. To execute the `RENAME TABLE` statement, you must have `ALTER` and `DROP` permissions on the source table. You must also have `CREATE` and `INSERT` permissions on the destination table.

You can execute the `RENAME TABLE` statement to rename views. Note that the source view and the destination view must belong to the same database.

18.5.10.15. CREATE SYNONYM

Synonyms are aliases of objects in databases. You can define aliases for most database objects, such as tables, views, materialized views, sequences, functions, stored procedures, packages, and synonyms.

Synonyms simplify the SQL-based development of applications. If you need to use objects in SQL statements and no synonyms are specified for the objects, you must obtain the locations where the objects are stored. For example, you must obtain the information about the database that stores the specified table object. If you have specified synonyms for the objects, you do not need to concern yourself with this issue.

Synonyms also offer other benefits. The following example is provided to explain another benefit. In the example, a developer writes an SQL statement. The SQL statement involves a table that is used in the production system. If a synonym is specified for the table, the developer can associate the synonym with a mock table during the initial test stage. After the test is passed, the developer can associate the synonym with the table that is stored in the production system. This avoids code modifications and reduces the impact of the test on the production system.

Syntax

```
CREATE [OR REPLACE] [PUBLIC] SYNONYM
[DATABASE.]synonym_name
FOR [DATABASE.]object_name;
```

You can use the `CREATE SYNONYM` syntax to create synonyms for objects. To create synonyms for objects, you do not need to have the permissions on the objects. However, you must have the permissions to create synonyms in the specified database. If you specify the `PUBLIC` keyword when you create a synonym, the synonym belongs to no databases. Therefore, the synonym does not share the same namespace with databases and belongs to the public namespace.

18.5.10.16. DROP SYNONYM

Syntax

```
DROP [PUBLIC] SYNONYM
[DATABASE.]synonym_name
[FORCE]
```

You can use the `DROP SYNONYM` syntax to drop synonyms. If you need to delete a public synonym, you must specify the `PUBLIC` keyword. In this scenario, you cannot specify the database for the synonym that is specified by the `synonym_name` parameter.

18.5.11. DML statements

18.5.11.1. Overview

Data manipulation language (DML) statements allow you to write, delete, and update data in databases.

ApsaraDB for OceanBase supports the following DML statements: `INSERT`, `REPLACE`, `SELECT`, `UPDATE`, and `DELETE`.

The following DML limits apply to partitioned tables:

- ApsaraDB for OceanBase distributes data across partitions. To ensure system performance, `WHERE` clauses in `SELECT`, `UPDATE`, and `DELETE` statements must include `partition(p0, p1,...)`. If `WHERE` clauses in `SELECT`, `UPDATE`, and `DELETE` statements do not include `partition(partition_list)`, all the partitions are scanned before the statements return data.
- If `SELECT`, `UPDATE`, and `DELETE` statements do not include `partition(partition_list)`, the system does not return syntax errors. In this scenario, the statements return results only after a full table scan is completed.
- You can specify only one partition in each `REPLACE`, `INSERT`, `UPDATE`, or `DELETE` statement. If you specify multiple partitions, the system returns errors.

If you need to specify multiple partitions in each `SELECT` statement, pay attention to the following considerations:

- If you send requests to read data from multiple `partition groups`, you can read data across partitions and only weak consistency is supported.
- If you send requests to read data from the same `partition group`, you can read data across partitions and strong consistency is supported. When data is migrated at the backend of ApsaraDB for OceanBase, partitions in the same `partition group` may be distributed across servers for a short period of several milliseconds. If this occurs, strong consistency cannot be ensured when you read data across partitions.

18.5.11.2. INSERT

The INSERT statement allows you to add one or more rows to a specified table.

Syntax

```
INSERT [INTO] tblname
    [(colname,...)]
    {VALUES|VALUE} ({expr | DEFAULT},...)
    [ ON DUPLICATE KEY UPDATE
        colname=expr
        [, colname=expr] ... ] ;
```

Or

```
INSERT [INTO] tblname
    [(colname,...)]
    {VALUES|VALUE} (colvalues,...)
    [ON DUPLICATE KEY UPDATE
        colname=expr
        [, colname=expr] ... ] ;
```

```

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  [(col_name,...)]
  {VALUES | VALUE} {(expr | DEFAULT),...}(...),...
  [ ON DUPLICATE KEY UPDATE
    col_name=expr
    [, col_name=expr] ... ]

```

Or:

```

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  SET col_name={expr | DEFAULT}, ...
  [ ON DUPLICATE KEY UPDATE
    col_name=expr
    [, col_name=expr] ... ]

```

Or:

```

INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  [(col_name,...)]
  SELECT ...
  [ ON DUPLICATE KEY UPDATE
    col_name=expr
    [, col_name=expr] ... ]

```

where:

- `[(colname,...)]` specifies the columns into which the data is inserted.
- If you need to insert multiple columns at a time, separate the columns with commas (,).
- The `ON DUPLICATE KEY UPDATE` clause is supported.
- The `INSERT` statement cannot be followed by `SET` operations.

If you execute the `INSERT... ON DUPLICATE KEY UPDATE...` statement, the number of `affected rows` is calculated based on the following rules:

- If you do not specify the `CLIENT_FOUND_ROWS` flag in `client_capabilities`, the number of affected rows is calculated based on this rule:
 - If a row is inserted as a new row, `affected_row= 1` is returned.
 - If an inserted row conflicts with an existing row in the table and the data in the table remains the same after the update, `affected_row = 0` is returned. If the data in the table before the update is different from that after the update, `affected_row= 2` is returned.
- If you specify the `CLIENT_FOUND_ROWS` flag, the number of affected rows is calculated based on the following rules:

- If a row is inserted as a new row, `affected_row=1` is returned.
- If the data in the table remains the same after the update, `affected_row=1` is returned.
- If the data in the table before the update is different from that after the update, `affected_row=2` is returned.
- If you do not specify the `CLIENT_FOUND_ROWS` flag, the value of the `affected_row` parameter is the number of updated rows. If you specify the `CLIENT_FOUND_ROWS` flag, the value of the `affected_row` parameter is the number of touched rows that conflict with existing rows. The data in the touched rows may not be modified.

Examples

1. Execute the following statement to insert rows:

```
INSERT INTO test VALUES (1, 'hello alipay'),(2, 'hello ob');
```

2. Execute the following statement to view the inserted rows that are shown in **Inserted rows**:

```
SELECT * FROM test;
```

Inserted rows

```
+-----+-----+
| c1    | c2    |
+-----+-----+
|      1 | hello alipay |
|      2 | hello ob     |
+-----+-----+
2 rows in set (0.01 sec)
```

3. Execute the following statements to create a partitioned table and insert a row to a partition of the table:

```
mysql> create table employees( id int not null, frame VARCHAR(20),lname VARCHAR(20), hired date not null default '1970-01-01', separated date not null default '9999-12-31', job_code int, store_id int ) partition by hash(store_id) partitions 4;
Query OK, 0 rows affected (0.34 sec)
mysql> insert into employees partition(p2) values (7,'7','7','1999-02-02','2007-07-07',7,10);
Query OK, 1 row affected (0.00 sec)
```

Errors

1. If an SQL syntax error occurs, the system returns the `1064` error.

```
mysql> insert employees into values employees(1, '1','1',2003-03-03,2003-03-03,1,1);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'into values employees(1, '1','1',2003-03-03,2003-03-03,1,1)' at line 1
```

2. If the type of the specified data is inconsistent with the data type that is defined in the table, the system returns the 1366 error. For example, if the data type that is defined in a table is INT and the type of the specified data is VARCHAR, the system returns this error. The 1366 error code indicates the error that the type of the specified data is invalid.

```
mysql> desc employees ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | NO | | NULL | |
| frame | VARCHAR(20) | YES | | NULL | |
| lname | VARCHAR(20) | YES | | NULL | |
| hired | date | NO | | 1970-01-01 | |
| separated | date | NO | | 9999-12-31 | |
| job_code | int(11) | YES | | 1 | |
| store_id | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> insert into employees values (1000, 1, '1', '2003-03-03', '2003-03-03', 'test', 1);
ERROR 1366 (HY000): Incorrect integer value: 'test' for column 'job_code' at row 1
```

3. If the type of the specified data cannot be recognized, the system returns the 1054 error. The 1054 error code indicates unknown column errors.

```
mysql> insert into employees values (1000, 1, '1', '2003-03-03', '2003-03-03', 'test', 1);
ERROR 1054 (42S22): Unknown column 'test' in 'field list'
```

4. If the format of the specified time is invalid, the system returns the 1292 error. The 1292 error code indicates the error that the date value is invalid.

```
mysql> create table t2(id int, work date);
Query OK, 0 rows affected (0.08 sec)

mysql> insert into t2 values(1, 2003-03-03);
ERROR 1292 (22007): Incorrect date value: '1997' for column 'work' at row 1

mysql> insert into t2 values(1, '2003-03-03');
Query OK, 1 row affected (0.02 sec)

mysql> insert into t2 values(1, test);
ERROR 1054 (42S22): Unknown column 'test' in 'field list'

mysql> insert into t2 values(1, 2003);
ERROR 1292 (22007): Incorrect date value: '2003' for column 'work' at row 1

mysql> insert into t2 values(1, 20030101);
Query OK, 1 row affected (0.01 sec)

mysql> insert into t2 values(1, 2003-01-01);
ERROR 1292 (22007): Incorrect date value: '2001' for column 'work' at row 1
```

5. If the specified numeric value is not in the specified range, the system returns the 1264 error.
6. If the length of the specified characters exceeds the maximum length, the system returns the 1406 error.


```
mysql> insert into employees partition(p5) values(3,1,1,'2001-01-01','2001-10-10',1,1);
ERROR 1735 (HY000): Unknown partition 'p5' in table 'employees'
```

12. If the information about the partitions into which data is inserted is inconsistent with the partition information that is parsed by the corresponding function, the system returns the 1748 error.

```
mysql> show create table employees;
+-----+-----+
| Table | Create Table
+-----+-----+
| employees | CREATE TABLE `employees` (
  `id` int(11) NOT NULL,
  `frame` VARCHAR(20) DEFAULT NULL,
  `lname` VARCHAR(20) DEFAULT NULL,
  `hired` date NOT NULL DEFAULT '1970-01-01',
  `separated` date NOT NULL DEFAULT '9999-12-31',
  `job_code` int(11) DEFAULT '1',
  `store_id` int(11) DEFAULT NULL
) DEFAULT CHARSET = utf8mb4 COMPRESSION = 'lz4_1.0' REPLICA_NUM = 1 BLOCK_SIZE = 16384 USE_BLOOM_FILTER =
FALSE TABLET_SIZE = 134217728 PCTFREE = 10 partition by hash(store_id) partitions 4 |
+-----+-----+
1 row in set (0.01 sec)

mysql> insert into employees partition(p2) values(3,1,1,'2001-01-01','2001-10-10',1,1);
ERROR 1748 (HY000): Found a row not matching the given partition set
```

13. If the inserted data is not in the data range that is defined in the partitions, the system returns the following error: ERROR 1526 (HY000): Table has no partition for value XX .

```
mysql> CREATE TABLE employees3 (
-> id INT NOT NULL,
-> fname VARCHAR(30),
-> lname VARCHAR(30),
-> hired DATE NOT NULL DEFAULT '1970-01-01',
-> separated DATE NOT NULL DEFAULT '9999-12-31',
-> job_code INT NOT NULL,
-> store_id INT NOT NULL
-> )
-> PARTITION BY RANGE (store_id) (
-> PARTITION ptest VALUES LESS THAN (6),
-> PARTITION ptestk VALUES LESS THAN (11),
-> PARTITION ptestl VALUES LESS THAN (16),
-> PARTITION ptestf VALUES LESS THAN (21)
-> );
Query OK, 0 rows affected (0.54 sec)

mysql> insert into employees3 values(3,'Meagge','simith','2011-01-10','2019-10-01',10,21);
ERROR 1526 (HY000): Table has no partition for value 21

mysql> insert into employees3 values(4,'Meery','simith','2012-01-10','2019-10-01',10,22);
ERROR 1526 (HY000): Table has no partition for value 22
```

14. If the inserted column values violate the unique index constraint, the system returns the following error:

```
ERROR 1062 (23000): Duplicate entry 'XXX' for key 'XXX' .
```

```
mysql> create table estest(c1 int primary key, c2 int, c3 int, c4 int);
Query OK, 0 rows affected (0.28 sec)
mysql> insert into estest values(1, 2, 3, 5), (2, 3, 4, 5), (5, 6, 7, 8);
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> create unique index uniindex on estest(c2); //The unique index uniindex is created.
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> select * from estest;
+----+-----+-----+-----+
| c1 | c2 | c3 | c4 |
+----+-----+-----+-----+
| 1 | 2 | 3 | 5 |
| 2 | 3 | 4 | 5 |
| 5 | 6 | 7 | 8 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
mysql> insert into estest values(7,6,8,9); //The value 6 in the inserted row is the same as an existing value that is
stored in the unique index column. This violates the constraint of the unique index.
ERROR 1062 (23000): Duplicate entry '6' for key 'uniindex'
```

18.5.11.3. REPLACE

Syntax

The REPLACE statement is executed in a similar way to the INSERT statement. Assume that a `primary key` value or a value of the unique index column for an existing row is the same as that for the new row. In this scenario, the REPLACE statement deletes the existing row before the statement inserts the new row. This is the only difference between the REPLACE statement and the INSERT statement.

To execute the REPLACE statement, you must have INSERT and DELETE permissions on the specified table.

Syntax

```
REPLACE [INTO] tblname
[(colname,...)]
{VALUES|VALUE} ({expr | DEFAULT},...);
```

The following code block describes the MySQL syntax of the REPLACE statement:

```
REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  [(col_name,...)]
  {VALUES | VALUE} {(expr | DEFAULT),...},(...),...
```

Or:

```
REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  SET col_name={expr | DEFAULT}, ...
```

Or:

```
REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  [(col_name,...)]
  SELECT ...
```

where:

- `[(colname,...)]` specifies the columns into which the data is inserted.
- If you need to replace data in multiple columns at a time, separate the columns with commas (,).

If you execute the REPLACE statement, the number of `affected rows` is calculated based on the following rules:

- If a row is inserted as a new row, `affected_row=1` is returned.
- If the data in the table before the replacement is different from that after the replacement, `affected_row=2` is returned. In this scenario, only one existing row has the same primary key value or the same unique index value as the new row. If multiple existing rows have the same primary key value or unique index value as the new row, the number of `affected rows` equals the number of conflicting rows plus 1.
- If the data remains the same after the replacement, the number of affected rows are calculated based on the following rules:
 - If a conflicting row is caused by the unique index in the table, the foreign key constraint is not used, and `ON DELETE TRIGGER` is not added to the REPLACE statement, `affected_row=1` is returned.
 - In other scenarios, `affected_row = 2` is returned.

Examples

1. Execute the following statement to replace the specified rows in the test table:

```
REPLACE INTO test VALUES (1, 'hello alibaba'),(2, 'hello ob');
```

2. Execute the following statement to view the inserted rows that are shown in the following figure:

```
SELECT * FROM test;
```

Replace the table rows

```

+-----+-----+
| c1    | c2    |
+-----+-----+
|      1 | hello alibaba |
|      2 | hello ob      |
+-----+-----+
2 rows in set (0.01 sec)

```

Errors

1. If an SQL syntax error occurs, the system returns the 1064 error.
2. If the type of the specified data is inconsistent with the data type that is defined in the table, the system returns the 1366 error. For example, if the data type that is defined in a table is INT and the type of the specified data is VARCHAR, the system returns this error. The 1366 error code indicates the error that the type of the specified data is invalid.
3. If the type of the specified data cannot be recognized, the system returns the 1054 error. The 1054 error code indicates unknown column errors.
4. If the format of the specified time is invalid, the system returns the 1292 error. The 1292 error code indicates the error that the date value is invalid.
5. If the specified numeric value is not in the specified range, the system returns the 1264 error.
6. If the length of the specified characters exceeds the maximum length, the system returns the 1406 error.
7. If NULL is inserted into a column that stores only non-null values, the system returns the 1048 error.
8. If the name of the specified table does not exist, the system returns the 1146 error.
9. If the number of inserted column values is different from the defined number of column values, the system returns the 1136 error.
10. If the names of the partitions into which data is inserted do not exist, the system returns the 1735 error.
11. If the information about the partitions into which data is inserted is inconsistent with the partition information that is parsed by the corresponding function, the system returns the 1748 error.

18.5.11.4. UPDATE

The UPDATE statement allows you to change field values in a specified table.

Syntax

Syntax

```
UPDATE tblname
SET colname=colvalues
    [, colname=colvalues...]
[WHERE where_condition]
[ORDER BY order_list]
[LIMIT row_count];

order_list:
colname [ASC|DESC] [, colname [ASC|DESC]...]
```

The following code block describes the MySQL syntax of the UPDATE statement:

```
Single-table syntax:

UPDATE [LOW_PRIORITY] [IGNORE] table_reference
    SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...
    [WHERE where_condition]
    [ORDER BY ...]
    [LIMIT row_count]

Multiple-table syntax:

UPDATE [LOW_PRIORITY] [IGNORE] table_references
    SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...
    [WHERE where_condition]
```

Examples

1. Execute the following statement to change `hello ob` to `hello oceanbase` :

```
UPDATE test SET c2='hello oceanbase' WHERE c1=2;
```

2. Execute the following statement to view the changed information that is shown in [Update the test table](#):

```
SELECT * FROM test;
```

Update the test table

```

+-----+-----+
| c1      | c2      |
+-----+-----+
|        1 | hello alibaba |
|         2 | hello oceanbase |
+-----+-----+
2 rows in set (0.00 sec)

```

3. Update a column value in a partition.

Execute the following statements to change the value of `job_code` in the `p0` partition of the `employees` table to 1:

```

mysql> select * from employees;
+-----+-----+-----+-----+-----+-----+
| id | frame | lname | hired   | separated | job_code | store_id |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | 4     | 4     | 2000-04-04 | 2044-04-04 | 4       | 4       |
| 1 | 1     | 1     | 2000-01-01 | 2014-01-01 | 1       | 1       |
| 5 | 5     | 5     | 2000-05-05 | 2022-05-05 | 5       | 5       |
| 2 | 2     | 2     | 2000-02-02 | 2024-02-02 | 2       | 2       |
| 6 | 6     | 6     | 2000-06-06 | 2022-06-06 | 6       | 6       |
| 7 | 7     | 7     | 1999-02-02 | 2007-07-07 | 7       | 10      |
| 3 | 3     | 3     | 2000-03-03 | 2034-03-03 | 3       | 3       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.05 sec)

mysql> update employees partition (p0) set job_code=1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

```

Errors

1. If an SQL syntax error occurs, the system returns the `1064` error.
2. If the type of the specified data is inconsistent with the data type that is defined in the table, the system returns the `1366` error. For example, if the data type that is defined in a table is `INT` and the type of the specified data is `VARCHAR`, the system returns this error. The `1366` error code indicates the error that the type of the specified data is invalid.
3. If the type of the specified data cannot be recognized, the system returns the `1054` error. The `1054` error code indicates unknown column errors.
4. If the format of the specified time is invalid, the system returns the `1292` error. The `1292` error code indicates the error that the date value is invalid.
5. If the specified numeric value is not in the specified range, the system returns the `1264` error.
6. If the length of the specified characters exceeds the maximum length, the system returns the `1406` error.

7. If NULL is inserted into a column that stores only non-null values, the system returns the 1048 error.
8. If the name of the specified table does not exist, the system returns the 1146 error.
9. If an inserted value of the primary key is the same as an existing value of the primary key, the system returns the 1062 error.
10. If the names of the partitions where data is updated do not exist, the system returns the 1735 error.
11. If the information about the partitions where data is updated is inconsistent with the partition information that is parsed by the corresponding function, the system returns the 1748 error.

18.5.11.5. DELETE

The DELETE statement allows you to delete the table rows that meet the specified conditions.

Syntax

```
DELETE FROM tblname
  [WHERE where_condition]
  [ORDER BY order_list]
  [LIMIT row_count];
```

The following code block describes the MySQL syntax of the DELETE statement:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
  tbl_name[*] [, tbl_name[*]] ...
  FROM table_references
  [WHERE where_condition]
```

Or:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
  FROM tbl_name[*] [, tbl_name[*]] ...
  USING table_references
  [WHERE where_condition]
```

Examples

1. Execute the following statement to delete the row that meets the c1=2 condition. The c1 column is the primary key in the test table.

```
DELETE FROM test WHERE c1 = 2;
```

2. Execute the following statement to view the table information after the row is deleted. **Delete the row** shows the output of the statement.

```
SELECT * FROM test;
```

Delete the row

```

+-----+-----+
| c1    | c2    |
+-----+-----+
|      1 | hello alibaba |
+-----+-----+
1 row in set (0.01 sec)

```

Errors

1. If an SQL syntax error occurs, the system returns the 1064 error.
2. If the type of the specified data cannot be recognized, the system returns the 1054 error. The 1054 error code indicates unknown column errors.
3. If the name of the specified table does not exist, the system returns the 1146 error.
4. If the names of the partitions into which data is inserted do not exist, the system returns the 1735 error.

18.5.11.6. SELECT

The SELECT statement allows you to query data that is stored in tables.

Basic queries

Syntax

```

SELECT
    [ALL | DISTINCT]
    selectexpr [[AS] othername] [, selectexpr ...]
    [FROM table_references ]
    [PARTITION(partitionid [, partitionid...])]
    [WHERE where_conditions]
    [GROUP BY group_by_list]
    [HAVING search_conditions]
    [ORDER BY order_list]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
    [FOR UPDATE];

```

The following code block describes the MySQL syntax of the SELECT statement:

```

SELECT
[ALL | DISTINCT | DISTINCTROW ]
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
  [PARTITION partition_list]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[PROCEDURE procedure_name(argument_list)]
[INTO OUTFILE 'file_name'
  [CHARACTER SET charset_name]
  export_options
 | INTO DUMPFILE 'file_name'
 | INTO var_name [, var_name]]
[FOR UPDATE | LOCK IN SHARE MODE]]
    
```

SELECT clauses describes SELECT clauses.

SELECT clauses

Clause	Description
ALL DISTINCT	<p>Specifies whether the statement returns only distinct rows. A database table may contain duplicate values.</p> <ul style="list-style-type: none"> If you add DISTINCT to the statement, only distinct rows are returned in the query results. If you add ALL to the statement, all the matched rows are returned. If you do not add DISTINCT or ALL to the statement, the default setting ALL is used.
select_expr	<p>Specifies the required expressions or column names. The expressions specify the columns that you want to retrieve. If you need to specify multiple expressions or column names, separate the specified expressions or column names with commas (.). You can use an asterisk (*) to indicate that all the columns in the table are returned.</p>
AS othername	<p>Renames output fields.</p>

Clause	Description
FROM <i>table_references</i>	<ul style="list-style-type: none"> Specifies the tables from which data is read. Allows you to retrieve data from multiple tables.
WHERE <i>where_conditions</i>	<p>Specifies the filter conditions. The query results contain only the data that meets the filter conditions. This clause is optional.</p> <p><i>where_conditions</i> specifies an expression.</p>
GROUP BY <i>group_by_list</i>	Divides data into groups.
HAVING <i>search_conditions</i>	Specifies the filter conditions. HAVING clauses are similar to WHERE clauses. The difference between HAVING and WHERE clauses is that you can use aggregate functions in HAVING clauses, such as SUM and AVG.
ORDER BY <i>order_list</i> <i>order_list</i> : <i>colname</i> [ASC DESC] [, <i>colname</i> [ASC DESC]...]	<p>Displays the query results in ascending or descending order. ASC indicates the ascending order and DESC indicates the descending order.</p> <p>If you do not specify the order, the default order ASC is used.</p>
[LIMIT {[<i>offset</i> ,] <i>row_count</i> <i>row_count</i> OFFSET <i>offset</i> }]	<p>Limits the number of rows that are returned by the SELECT statement.</p> <p>You can specify one or two arguments of the numeric data type for the LIMIT clause. The arguments must be integer constants.</p> <ul style="list-style-type: none"> If you specify two arguments, the first argument specifies the offset for the first row to be returned and the second argument specifies the maximum number of rows to be returned. The initial offset for the first row is 0 instead of 1. If you specify only one argument, the argument specifies the maximum number of rows to be returned and the offset is 0.
FOR UPDATE	Applies an exclusive lock on each row of the query results. This prevents other transactions from concurrently updating the rows. This also prevents other transactions from concurrently reading the rows for which some transaction isolation levels are specified.
PARTITION(<i>partition_list</i>) Format: partition(p0,p1...)	Specifies the partition information of the specified tables.

JOIN syntax

Joins are divided into inner joins and outer joins. Outer joins are divided into left joins, right joins, and full joins. After two tables are joined, you can use the ON clause to filter the data in the tables.

In ApsaraDB for OceanBase, you can use USING clauses for joins. At least one of the join conditions must use the equal to (=) operator.

For inner joins, the query results contain only the matched rows in both tables.

For left joins, the query results contain all the rows in the table on the left of the `LEFT [OUTER] JOIN` keyword, and the matched rows in the table on the right of the keyword.

For right joins, the query results contain all the rows in the table on the right of the `[RIGHT] [OUTER] JOIN` keyword, and the matched rows in the table on the left of the keyword.

For full joins, the query results contain all the rows in both tables.

Set operations

In ApsaraDB for OceanBase, the main clauses for set operations are UNION, EXCEPT, and INTERSECT clauses.

- UNION clause

The UNION clause combines the result sets of two or more SELECT statements. To use the UNION clause, pay attention to the following considerations:

- For the UNION clause, the number of columns, the data types of the columns, and the column sequence for each of the SELECT statements must be the same.
- By default, the UNION operator returns result sets that do not contain duplicate values. If you need to obtain duplicate values, use the `UNION ALL` clause.
- The column names in each UNION result set are always the same as those in the first SELECT statement for the UNION clause.

The UNION clause combines the results of two or more SELECT statements. The JOIN clause retrieves data from two or more tables. Therefore, the UNION clause is similar to the JOIN clause. The difference between the two clauses is that the UNION clause combines the results of two or more SELECT statements and the JOIN clause joins two or more tables.

- EXCEPT clause

The EXCEPT clause returns the data that is included in the result set of the first SELECT statement but is excluded from the result set of the second SELECT statement.

- INTERSECT clause

The INTERSECT clause returns the data that is included in the result sets of both SELECT statements.

DUAL virtual table

DUAL is a virtual table. The DUAL virtual table can be considered as a special table that has one row and no columns. If you do not need to retrieve data from specific tables but need to execute the SELECT statement to retrieve some required information, DUAL helps you retrieve the required information. You can use the SELECT syntax that includes DUAL to retrieve the values of user variables or system variables.

If the SELECT statement does not include the FROM clause, `FROM DUAL` is equivalent to the FROM clause. In this scenario, the expressions in the SELECT statement can be only constant expressions.

Syntax

```
SELECT
    [ALL | DISTINCT]
    select_list
    [FROM DUAL [WHERE where_condition]]
    [LIMIT {[offset,] rowcount | rowcount OFFSET offset};
```

`SELECT... FOR UPDATE` statement

Syntax

```
SELECT ... FOR UPDATE [WAIT n| NOWAIT];
```

where:

- The `WAIT` clause specifies the number of seconds to wait. The current user must wait for the specified number of seconds before another user releases the row locks. This prevents endless waiting.
- `NOWAIT` indicates that the current user does not wait for the row locks to release.

You can execute the `SELECT ... FOR UPDATE` statement to apply an exclusive lock on each row in the query results. This prevents other transactions from concurrently updating these rows. This also prevents other transactions from concurrently reading the rows for which some transaction isolation levels are specified. To be more specific, you can use the `FOR UPDATE` clause to lock the tuples of the query results. In this scenario, the `UPDATE`, `DELETE` and `FOR UPDATE` operations cannot be performed on these tuples before the transaction is committed.

 **Notice** In ApsaraDB for OceanBase, you can run each query on only a single table.

Example:

```
SELECT * FROM a FOR UPDATE;
```

IN and OR logical operators

ApsaraDB for OceanBase supports `IN` and `OR` logical operators.

Errors

1. If an SQL syntax error occurs, the system returns the `1064` error.
2. If the type of the specified data cannot be recognized, the system returns the `1054` error. The `1054` error code indicates unknown column errors.
3. If the name of the specified table does not exist, the system returns the `1146` error.
4. If the names of the partitions from which data is retrieved do not exist, the system returns the `1735` error.
5. If a `SELECT` subquery returns multiple rows, the system returns the `1241` error. The `1241` error code indicates that the result violates the rule: The subquery results can contain only one field.

```
mysql> select * from employees where store_id=5;
+----+-----+-----+-----+-----+-----+-----+
| id | frame | lname | hired   | separated | job_code | store_id |
+----+-----+-----+-----+-----+-----+-----+
| 5 | 5   | 5   | 2000-05-05 | 2022-05-05 | 123 | 5 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select (select id from employees where store_id=5);
+-----+
| (select id from employees where store_id=5) |
+-----+
| 5 |
+-----+
1 row in set (0.00 sec)

mysql> select (select id,frame from employees where store_id=5);
ERROR 1241 (21000): Operand should contain 1 column(s)
```

6. If the referenced group functions are invalid, the system returns the `1111` error.

```
mysql> CREATE TABLE t222 (s1 INT, s2 INT, s3 INT, PRIMARY KEY(s3) );
mysql> insert into t222 values(1, 1, 1), (1, 2, 2), (1, 2, 3);
mysql> select avg(sum(s1)) from t222 group by s1;
ERROR 1111 (HY000): Invalid use of group function
```

```
mysql> select avg(a1) from (select sum(s1) as a1 from t222 group by s1);
```

18.5.12. Transaction language

A database transaction is a single logical unit of work that consists of a collection of operations.

Transaction processing ensures that SQL operations in a batch are all executed or are not executed at all. You can use transactions to maintain the data integrity of databases.

An explicit transaction is a user-defined or user-specified transaction. An explicit transaction is a transaction that starts with the `BEGIN TRANSACTION`, `BEGIN`, or `BEGIN WORK` statement and ends with the `COMMIT` or `ROLLBACK` statement. `BEGIN` and `BEGIN WORK` are supported as aliases of the `START TRANSACTION` statement.

To start a transaction, use the following statement syntax:

```
START TRANSACTION
  [WITH CONSISTENT SNAPSHOT];
BEGIN [WORK] ;
COMMIT [WORK] ;
ROLLBACK [WORK];
```

ApsaraDB for OceanBase supports only the `READ COMMITTED` isolation level.

- The `WITH CONSISTENT SNAPSHOT` clause starts a consistent read. This clause has the same effect as issuing a `START TRANSACTION` statement that is followed by a `SELECT` statement that queries an ApsaraDB for OceanBase table.

You can specify the `WITH CONSISTENT SNAPSHOT` clause in the `START TRANSACTION` statement. However, the `WITH CONSISTENT SNAPSHOT` does not take effect.

- `BEGIN` and `BEGIN WORK` are supported as aliases of `START TRANSACTION` and are used to initialize a transaction. `START TRANSACTION` is a standard SQL syntax and is the recommended way to start an ad hoc transaction. After a transaction is started, the SQL statements that follow the `START TRANSACTION` statement such as `INSERT`, `UPDATE`, and `DELETE` take effect only when the transaction is explicitly committed. However, a `REPLACE` statement is not subject to this limit.

To commit a transaction, use the following statement syntax:

```
COMMIT [WORK];
```

To roll back a transaction, use the following statement syntax:

```
ROLLBACK [WORK];
```

Example:

Perform a transaction on Table A: Find the row whose id is 3, and change the value for the name column to c, and insert a new row about sale records for product a.

Table A

id	name	num	sell_date
1	a	100	2013-06-21 10:06:43
2	b	200	2013-06-21 13:07:21
3	a	50	2013-06-21 13:08:15

- Execute the following statements in sequence to start the transaction.

```
START TRANSACTION;
UPDATE a SET name = 'c' WHERE id = 3;
INSERT INTO a VALUES (4, 'a', 30, '2013-06-21 16:09:13');
COMMIT;
```

- After the transaction is committed, execute the following statement to query data of Table A. [Table A information](#) shows the execution result.

```
SELECT * FROM a;
```

Table A information

```
+-----+-----+-----+-----+
| id   | name | num  | sell_date          |
+-----+-----+-----+-----+
|    1 | a    | 100  | 2013-06-21 10:06:43 |
|    2 | b    | 200  | 2013-06-21 13:07:21 |
|    3 | c    | 50   | 2013-06-21 13:08:15 |
|    4 | a    | 30   | 2013-06-21 16:09:13 |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Note

Before you commit a transaction, you can check whether the operations in the transaction have taken effect. For example, you can insert a `SELECT * FROM a;` statement before the `COMMIT` clause.

The session within which this transaction is executed can read the updated result. A session outside this transaction cannot read the updated result. Before the transaction is committed, your previous operations are invisible outside the transaction session. To roll back a transaction, execute the `ROLLBACK` statement to undo the `COMMIT` operation.

18.5.13. Database management language

18.5.13.1. CREATE RESOURCE UNIT

```
CREATE RESOURCE UNIT unitname
MAX_CPU [=] cpunum,
MAX_MEMORY [=] memsize,
MAX_IOPS [=] iopsnum,
MAX_DISK_SIZE [=] disksize,
MAX_SESSION_NUM [=] sessionnum,
[MIN_CPU [=] cpunum,]
[MIN_MEMORY [=] memsize,]
[MIN_IOPS [=] iopsnum] ;
```

When you create a resource unit, you must specify the values of `MAX_CPU`, `MAX_MEMORY`, `MAX_IOPS`, `MAX_DISK_SIZE`, and `MAX_SESSION_NUM`. `MIN_CPU`, `MIN_MEMORY`, and `MIN_IOPS` are optional. The default values of `MIN_CPU`, `MIN_MEMORY`, and `MIN_IOPS` are the same as the default values of `MAX_CPU`, `MAX_MEMORY`, and `MAX_IOPS`.

- The value range of `MAX_MEMORY` is [1073741824, +∞). The unit is byte. The minimum value is 1 GB.
- The value range of `MAX_IOPS` is [128, +∞).
- The value range of `MAX_DISK_SIZE` is [536870912, +∞). The unit is byte. The minimum value is 512 MB.
- The value range of `MAX_SESSION_NUM` is [64, +∞).

You can replace *memsize* or *disksize* with a value that is a combination of a number and a unit, for example, 1 GB or 100 MB. You can also specify the value in bytes, for example, 1073741824 or 104857600.

For example, the following two statements are equivalent:

```
mysql> CREATE RESOURCE UNIT unit1 max_cpu 1, max_memory '1G', max_iops 128,max_disk_size '10G', max_session_num
64, MIN_CPU=1, MIN_MEMORY='1G', MIN_IOPS=128;
Query OK, 0 rows affected (0.02 sec)
```

Equivalent to:

```
mysql> CREATE RESOURCE UNIT unit1 max_cpu 1, max_memory 1073741824, max_iops 128, max_disk_size 10737418240, m
ax_session_num 64, MIN_CPU=1, MIN_MEMORY=1073741824, MIN_IOPS=128;
Query OK, 0 rows affected (0.01 sec)
```

18.5.13.2. ALTER RESOURCE UNIT

```
ALTER RESOURCE UNIT unitname
MAX_CPU [=] cpunum,
MAX_MEMORY [=] memsize,
MAX_IOPS [=] iopsnum,
MAX_DISK_SIZE [=] disksize,
MAX_SESSION_NUM [=] sessionnum,
[MIN_CPU [=] cpunum,]
[MIN_MEMORY [=] memsize,]
[MIN_IOPS [=] iopsnum] ;
```

When you modify the properties of a resource unit, follow the same configuration rules that are described in the `CREATE RESOURCE UNIT` topic.

18.5.13.3. DROP RESOURCE UNIT

```
DROP RESOURCE UNIT unitname
```

You can execute this statement to delete a resource unit.

For example, execute the following statement to delete unit 1:

```
mysql> DROP RESOURCE UNIT unit1;  
Query OK, 0 rows affected (0.00 sec)
```

18.5.13.4. CREATE RESOURCE POOL

Syntax

```
CREATE RESOURCE POOL poolname  
UNIT [=] unitname,  
UNIT_NUM [=] unitnum,  
ZONE_LIST [=] ('zone' [, 'zone' ...]) ;
```

A resource pool contains multiple resource units. You must specify the zones to which the resource units belong.

ApsaraDB for OceanBase allows only one type of resource units in a resource pool. UNIT_NUM indicates the number of resource units in a zone. The UNIT_NUM value must be smaller than the number of OBServers in the zone.

Examples

```
mysql> CREATE RESOURCE POOL pool1 unit='unit1', unit_num=1, zone_list=('zone1');  
Query OK, 0 rows affected (0.01 sec)
```

18.5.13.5. ALTER RESOURCE POOL

Syntax

```
ALTER RESOURCE POOL poolname  
UNIT [=] unitname,  
UNIT_NUM [=] unitnum,  
ZONE [=] ('zone' [, 'zone' ...]) ;
```

You can execute this statement to modify the properties of a resource pool.

If your modification results in no resource unit in the resource pool, make sure that the resource pool is not used by a tenant when you execute the ALTER RESOURCE POOL statement. If the resource pool is in use, the system returns an error.

When you modify the properties of a resource pool by executing the ALTER RESOURCE POOL statement, you can modify only one property at a time. To modify two or more of the UNIT, UNIT_NUM, and ZONE properties, you must execute the ALTER RESOURCE POOL statement two or more times.

Examples

```
//If you modify multiple properties of a resource pool at a time, the system returns an error.
mysql> ALTER RESOURCE POOL pool1 unit='unit2', unit_num=1, zone_list=('zone1');
ERROR 1235 (0A000): alter unit_num, resource_unit, zone_list in one cmd not supported

//Modify one property at a time.
mysql> ALTER RESOURCE POOL pool1 unit='unit2';
Query OK, 0 rows affected (0.00 sec)
```

18.5.13.6. DROP RESOURCE POOL

Syntax

```
DROP RESOURCE POOL poolname;
```

You can execute this statement to delete a resource pool.

Examples

Execute the following statement to delete pool1:

```
mysql> DROP RESOURCE POOL pool1;
Query OK, 0 rows affected (0.00 sec)
```

18.5.13.7. CREATE TENANT

Syntax

```
CREATE TENANT [IF NOT EXISTS] tenantname
  [tenant_characteristic_list]

tenant_characteristic_list:
tenant_characteristic [, tenant_characteristic...]

tenant_characteristic:
COMMENT 'string'
|[CHARACTER SET | CHARSET] [=] charsetname
|COLLATE [=] collationname
|REPLICA_NUM [=] num
|ZONE_LIST [=] (zone [, zone...])
|PRIMARY_ZONE [=] zonelist
|DEFAULT TABLEGROUP [=] {NULL | tablegroup}
|RESOURCE_POOL_LIST [=](poolname [, poolname...])
|LOCALITY [=] locality
```

If the specified tenant name is already used and the `IF NOT EXISTS` option is not specified, the system returns an error.

The validity requirements for tenant names are the same as those for variable names. A tenant name must be up to 64 bytes in length and can contain only letters, digits, and underscores (_). The name must start with a letter or an underscore (_) and cannot be a keyword that is reserved for ApsaraDB for OceanBase.

Before you execute the `CREATE TENANT` statement to create a tenant, you must connect your root user to the root tenant (`root@ROOT`).

Note

You must specify `RESOURCE_POOL_LIST` when you create a tenant.

When you specify `RESOURCE_POOL_LIST` for the `CREATE TENANT` statement, only one resource pool is supported.

`DEFAULT TABLEGROUP` specifies a default table group for the tenant. If you do not specify this parameter, the value is `NULL`.

Examples

```
mysql> CREATE TENANT IF NOT EXISTS t1 charset='utf8mb4', replica_num=1, zone_list=('zone1'), primary_zone='zone1', resource_pool_list=('pool1');
Query OK, 0 rows affected (0.26 sec)
```

Errors

- If your statement has a syntax error, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax; .`
- If the specified tenant name is already used and the `IF NOT EXISTS` option is not specified, the system returns the following error: `ERROR(OB_ERR_ALREADY_EXISTS, -5025, -1, "42501", "Already exist"); .`

18.5.13.8. ALTER TENANT

Syntax

```
ALTER TENANT tenantname [SET] [tenant_options]
```

tenant_options:

```
tenant_option [ tenant_option...]
```

tenant_options:

```
COMMENT [=]'string'
```

```
{{CHARACTER SET | CHARSET} [=] charsetname
```

```
[COLLATE [=] collationname
```

```
[REPLICA_NUM [=] num
```

```
[ZONE_LIST [=] (zone [, zone...])
```

```
[PRIMARY_ZONE [=] zonelist
```

```
[RESOURCE_POOL_LIST [=](poolname [, poolname...])
```

```
[DEFAULT TABLEGROUP [=] {NULL | tablegroupname}
```

```
{{READ ONLY | READ WRITE}
```

```
| LOCALITY [=] locality
```

 **Note** You must specify `RESOURCE_POOL_LIST` when you create a tenant.

The system tenant users and the administrator of the current tenant have the permission to execute the `ALTER TENANT` statement.

`DEFAULT TABLEGROUP` specifies a default table group for the tenant. A NULL value indicates no default table group for the database.

Errors

If your statement has a syntax error, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax;` .

18.5.13.9. Lock or unlock a tenant

Syntax

```
ALTER TENANT tenantname LOCK|UNLOCK;
```

You can execute this statement to lock a tenant. After the tenant is locked, you cannot create sessions on the tenant. The existing sessions remain unchanged. You can lock a tenant if the subscription of the service is not renewed upon expiration. After the subscription is renewed, you can unlock the tenant.

Examples

- Execute the following statement to lock TENANT1:

```
ALTER TENANT TENANT1 LOCK;
```

- Execute the following statement to unlock TENANT1:

```
ALTER TENANT TENANT1 UNLOCK;
```

Errors

If your statement has a syntax error, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax;` .

18.5.13.10. DROP TENANT

Syntax

```
DROP TENANT tenantname;
```

DROP You can execute this statement to delete an ApsaraDB for OceanBase tenant.

Before you execute the `DROP TENANT` statement to delete a tenant, you must connect your root user to the root tenant (`root@ROOT`).

You can delete a tenant only if the tenant is in the Locked state. If you execute the `DROP TENANT` statement to delete an unlocked tenant, the system returns an error.

Examples

Execute the following statement to delete TENANT1:

```
DROP TENANT TENANT1;
```

Errors

If your statement has a syntax error, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax; .`

18.5.13.11. CREATE TABLEGROUP

Syntax

```
CREATE TABLEGROUP [IF NOT EXISTS] tablegroupname
```

If the specified table group name is already used and the `IF NOT EXISTS` option is not specified, the system returns an error.

A table group name must be up to 64 characters in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_) and cannot be a keyword that is reserved for ApsaraDB for OceanBase.

Only the tenant administrator can create a table group.

Examples

```
CREATE TABLEGROUP myTableGroup1;
```

18.5.13.12. DROP TABLEGROUP

Syntax

```
DROP TABLEGROUP [IF EXISTS] tablegroupname
```

Examples

```
DROP TABLEGROUP myTableGroup1;
```

Errors

- If your statement has a syntax error, the system returns the following error: `ERROR 1064 (42000): You have an error in your SQL syntax; .`
- Before you delete a table group, make sure that no object such as a table or an index is using the table group. If the table group is still in use, the system returns an error.

18.5.13.13. ALTER TABLEGROUP

Syntax

```
ALTER TABLEGROUP tablegroupname ADD [TABLE] tblname [, tblname...]
```

You can execute this statement to add multiple tables to a table group. Table names are separated with commas (,).

If you add multiple tables at a time, you can specify duplicate table names. If a table to be added already exists in the table group that is specified by `tablegroupname`, the system does not return an error.

18.5.13.14. Users and permissions

You can manage users and permissions on databases. These management operations include creating a user, deleting a user, changing a password, changing a username, locking a user, granting permissions, and revoking permissions.

ApsaraDB for OceanBase users are divided into two types: users of the system tenant and users of general tenants.

When you create a user, if the current session is created on the system tenant, the new user is created for the system tenant. If the current session is created on a general tenant, the new user is created for the general tenant.

Username are unique within a tenant. Users under different tenants can have the same name. The Username@TenantName must be globally unique in the cluster. Usernames of users of the system tenant have a predetermined prefix. You can distinguish users of the system tenant from users of general tenants based on this characteristic. The system tenant or each general tenant has a built-in root user. The root user for a system tenant is the system administrator, and the root user for a general tenant is the tenant administrator. If you purchase a general tenant, you have the permissions to use the root user and the password of the general tenant to manage resources within the tenant.

Users of a general tenant can access only objects in the general tenant. This is the same as the logic of MySQL. Users of the system tenant can access objects that belong to different tenants. Users of the system tenant cannot access user tables that are stored in general tenants. When you log on to the ApsaraDB for OceanBase system, you must specify a unique tenant name. If you are using a user of the system tenant, you can execute the `CHANGE EFFECTIVE TENANT tenantname` statement to access another tenant after you log on to the system. However, if you are using a user of a general tenant, you cannot switch the tenant.

Create users

You can execute the `CREATE USER` statement to create an ApsaraDB for OceanBase user.

After a user is created, you can use the user to connect to ApsaraDB for OceanBase.

Syntax

```
CREATE USER user_specification_list;
user_specification_list:
    user_specification [, user_specification]...;
user_specification:
    user IDENTIFIED BY 'authstring'
    user IDENTIFIED BY PASSWORD 'hashstring'
```

Notes:

- To execute the `CREATE USER` statement, you must have the global `CREATE USER` permission.
- After a user is created, a new row is added for the user to the `mysql.user` table. If the username is already used by an existing user, the system returns an error.
- You can specify the `IDENTIFIED BY` clause to set the password for your user.
- `user IDENTIFIED BY` Specify user `IDENTIFIED BY 'authstring'` to set a plaintext password. After the password is saved to the `mysql.user` table, the password is stored in ciphertext on the server.
- Specify `user IDENTIFIED BY PASSWORD 'hashstring'` to set a ciphertext password.
- If you create multiple users at a time, separate pairs of user information with commas (,).

Examples

```
Oceanbase>CREATE USER 'sqluser01' IDENTIFIED BY '123456', 'sqluser02' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.07 sec)
Oceanbase>select user from user;
+-----+
| user  |
+-----+
| root  |
| test  |
| sqluser01 |
| sqluser02 |
+-----+
4 rows in set (0.01 sec)
```

Delete users

You can execute the `DROP USER` statement to delete one or more ApsaraDB for OceanBase users.

Syntax

```
DROP USER username [, username...];
```

Notes:

- To execute the `DROP USER` statement, you must have the global `CREATE USER` permission.
- You cannot delete a user by performing a `DELETE` operation on the `mysql.user` table.
- After a user is deleted, all permissions of the user are deleted.
- If you delete multiple users at a time, separate the usernames with commas (,).

Examples

Execute the following statement to delete the user `sqluser02` :

```
DROP USER 'sqluser02';
```

Change a password

You can change the password for a user that is used to log on to the ApsaraDB for OceanBase system.

Syntax

```
SET PASSWORD [FOR user] = password_option;
password_option: {
    PASSWORD('authstring')
    ['hashstring']
}
```

Or

```
ALTER USER username IDENTIFIED BY 'password';
```

Notes:

- If you do not specify the `FOR user` clause in the statement, the system changes the password for the current user. After a user logs on to the ApsaraDB for OceanBase system, the user can change the password.
- If you specify the `FOR user` clause in the statement, or you use the second syntax, the system changes the

password for the specified user. To change the password for a specified user, you must have the global `CREATE USER` permission.

Examples

Execute the following `ALTER USER` statement to change the password of `sqluser01` to `abc123` :

```
ALTER USER sqluser01 IDENTIFIED BY 'abc123';
```

Execute the following `SET PASSWORD` statement to change the password:

```
Oceanbase>set password for test = password('abc123');
Query OK, 0 rows affected (0.03 sec)
# If you do not specify the password function, the system returns the following error:
Oceanbase>set password for test = 'abc123';
ERROR 1827 (42000): The password hash doesn't have the expected format. Check if the correct password algorithm is being used with the PASSWORD() function.
```

Change a username

You can change the username of a user that is used to log on to the ApsaraDB for OceanBase system.

Syntax

```
RENAME USER
  'oldusername' TO 'newusername'
  [, 'oldusername' TO 'newusername'...];
```

Notes:

- To execute this statement, you must have the global `CREATE USER` permission.
- If you change multiple usernames at a time, separate the pairs of username information with commas (,).
- After you change the username for a user, the permissions of the user remain unchanged.
- The username must be up to 16 bytes in length.

Examples

```
Oceanbase>select user from user;
+-----+
| user  |
+-----+
| root  |
| testall |
+-----+
2 rows in set (0.00 sec)
```

Change a username.

```
Oceanbase>rename user testall to test;
Query OK, 0 rows affected (0.03 sec)
Oceanbase>select user from user;
+-----+
| user |
+-----+
| root |
| test |
+-----+
2 rows in set (0.00 sec)
```

Lock a user

You can lock or unlock a user. Locked users are not allowed to log on to the ApsaraDB for OceanBase system.

Syntax

```
ALTER USER user [lock_option]
lock_option:{
  ACCOUNT LOCK
  | ACCOUNT UNLOCK}
```

To execute this statement, you must have the global `UPDATE USER` permission.

Examples

- Lock a user.

```
Oceanbase>alter user test account lock;
Query OK, 0 rows affected (0.04 sec)
```

- Unlock a user.

```
Oceanbase>alter user test account unlock;
Query OK, 0 rows affected (0.02 sec)
```

Grant permissions to a user

You can execute the `GRANT` statement as a system administrator to grant permissions to a user.

Syntax

```

GRANT priv_type
    ON priv_level
    TO user_specification [, user_specification]...
    [WITH with_option ...]
priv_level:
    *
    | *.*
    | db_name.*
    | db_name.tbl_name
    | tbl_name
user_specification:
    user [IDENTIFIED BY [PASSWORD] 'password']
with_option:
    GRANT OPTION
    
```

Permissions can be divided into the following levels:

- **Global permissions**
Global permissions apply to all databases. To grant global permissions, use the `GRANT ALL ON *.*` syntax.
- **Database permissions**
Database permissions apply to all objects in a specified database. To grant permissions on a specified database, use the `GRANT ALL ON db_name.*` syntax.
- **Table permissions**
Table permissions apply to all columns in a specified table. To grant permissions on a specified table, use the `GRANT ALL ON db_name.tbl_name` syntax.

Notes:

- You must grant permissions to specific users. If a specified user does not exist, the system can create the user if the SQL mode is `NO_AUTO_CREATE_USER`. You can specify this SQL mode by executing `sql_mode='no_auto_create_user'`. If you do not use the `IDENTIFIED BY` clause to specify the password, the system cannot create the user.
- To grant permissions to a user, you must have the permission that you are granting. For example, if you use user1 to grant user2 the `SELECT` permission on table t1, user1 must have the `SELECT` permission on table t1. You must also have the `GRANT OPTION` permission.
- After a permission is granted, the authorized user must relog on to the ApsaraDB for OceanBase system so that the permission can take effect.
- You can use an asterisk (*) instead of specifying a table name to grant permissions on all tables in the specified database.
- If you grant multiple permissions to a user at a time, separate the permission types with commas (,).
- If you grant permissions to multiple users at a time, separate the usernames with commas (,).
- **Description of the priv_type values** lists the priv_type values that can be specified for the `GRANT` statement.

Description of the priv_type values

Permission	Description
ALL PRIVILEGES	All permissions except <code>GRANT OPTION</code> .

Permission	Description
ALTER	The permission to execute the <code>ALTER TABLE</code> statement.
CREATE	The permission to execute the <code>CREATE TABLE</code> statement.
CREATE USER	The permission to execute the <code>CREATE USER</code> , <code>DROP USER</code> , <code>RENAME USER</code> , and <code>REVOKE ALL PRIVILEGES</code> statements.
CREATE TABLEGROUP	The global permission to execute the <code>CREATE TABLEGROUP</code> statement.
DELETE	The permission to execute the <code>DELETE</code> statement.
DROP	The permission to execute the <code>DROP</code> statement.
GRANT OPTION	The permission to execute the <code>GRANT OPTION</code> statement.
INSERT	The permission to execute the <code>INSERT</code> statement.
SELECT	The permission to execute the <code>SELECT</code> statement.
UPDATE	The permission to execute the <code>UPDATE</code> statement.
SUPER	The permission to execute the <code>SET GLOBAL</code> statement to modify global system parameters.
SHOW DATABASES	The global permission to execute the <code>SHOW DATABASES</code> statement to display all databases.
INDEX	The permission to execute the <code>CREATE INDEX</code> and <code>DROP INDEX</code> statements.
CREATE VIEW	The permission to create or delete a view.
SHOW VIEW	The permission to execute the <code>SHOW CREATE VIEW</code> statement.

 **Note** You cannot grant users the permission to execute the `CHANGE EFFECTIVE TENANT` statement. However, all users of the system tenant have this permission.

Revoke permissions

You can execute the `REVOKE` statement as a system administrator to revoke the specified permissions from users.

Syntax

```
REVOKE priv_type
      ON database.tblname
      FROM 'user';
```

Notes:

- To revoke permissions from a user, you must have the permission that you are revoking. For example, if you use user1 to revoke the SELECT permission on table t1 from user2, user1 must have the SELECT permission on table t1. You must also have the GRANT OPTION permission.
- To revoke the ALL PRIVILEGES and GRANT OPTION permissions, you must have the global GRANT OPTION permission or the UPDATE and DELETE permissions on the permission list.
- Revocations do not have cascading effects. Assume that user1 grants permissions to user2. When the permissions are revoked from user1, the permissions are not revoked from user2.
- You can use an asterisk (*) instead of specifying a table name to revoke permissions on all tables in the specified database.
- If you revoke multiple permissions from a user at a time, separate the permission types with commas (,).
- If you revoke permissions from multiple users at a time, separate the usernames with commas (,).
- [Description of the priv_type values](#) lists the priv_type values that can be specified for the REVOKE statement.

Examples

Revoke all permissions from obsqluser01 .

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'obsqluser01';
```

View permissions

You can execute the SHOW GRANTS statement as a system administrator to view permissions that have been granted to a user.

Syntax

```
SHOW GRANTS [FOR username];
```

Notes:

- If you do not specify the username, this statement returns the permissions that have been granted to the current user. You can always view the permissions of the user that you are using.
- To view the permissions of a non-current user, you must have the SELECT permission on the mysql.user table.

Examples

```
Oceanbase>show grants for test;
+-----+
| Grants for test      |
+-----+
| GRANT USAGE ON *. * TO 'test'|
+-----+
1 row in set (0.01 sec)
```

18.5.13.15. Modify system variables

System variables such as autocommit and tx_isolation affect SQL functionality and are stored in the __all_sys_variable table.

ApsaraDB for OceanBase maintains two types of variables:

- Global variables

Global variables affect overall operations on the ApsaraDB for OceanBase system. When the ApsaraDB for OceanBase system starts, the system initializes all global variables to the default values. To modify global variables, you must have the SUPER permission.

- Session variables

Session variables affect operations on the clients that are connected to the ApsaraDB for OceanBase system. When a client connects to the ApsaraDB for OceanBase system, the system initializes the session variables of the client to the current values of the corresponding global variables. To set session variables, you do not need special permissions. You can modify only the session variables of your client. You cannot modify the session variables of other clients.

 **Note** Modifications to global variables do not affect the session variables of clients that are connected to the ApsaraDB for OceanBase system. This is true even if the `SET GLOBAL` statement is executed on the connected clients.

Syntax

- Set a global variable.

```
SET GLOBAL system_var_name = expr;
```

Or

```
SET @@GLOBAL.system_var_name = expr;
```

- Set a session variable.

```
SET [SESSION | @@SESSION. | LOCAL | LOCAL. | @@]system_var_name =expr;
```

- Display the values of system variables.

If the GLOBAL modifier is specified, the SHOW VARIABLES statement displays the values of global system variables. If the SESSION modifier is specified, this statement displays the values of system variables that are specific to the current session. If no modifier is specified, this statement displays the values of system variables that are specific to the current session.

```
SHOW [GLOBAL | SESSION]
  VARIABLES
  [LIKE 'system_var_name' | WHERE expr];
```

You can execute a `SELECT @@var_name` statement to search for a variable. If you do not specify the GLOBAL modifier, the system returns the value of a session-specific variable. If you specify the GLOBAL modifier, the system returns the value of a global variable.

Configure the READ ONLY property of a tenant

```
SET GLOBAL READ_ONLY={ON | OFF};
```

Or

```
SET @@GLOBAL.READ_ONLY={ON | OFF};
```

Examples

- Execute the following statement to change the value of the ob_trx_timeout session variable:

```
SET @@SESSION.ob_tx_timeout = 900000;
```

- Execute the following statement to query the value of the ob_trx_timeout variable:

```
Oceanbase>show variables like 'ob_trx_timeout';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| ob_trx_timeout | 100000000 |
+-----+-----+
1 row in set (0.02 sec)
```

- Execute the following SELECT statement to query the value of the wait_timeout system variable:

```
SELECT @@wait_timeout;
+-----+
| @@wait_timeout |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

18.5.13.16. Modify user variables

A user variable in one statement stores a user-defined value that you can refer to later in another statement. This helps you pass values between statements. User variables are session-specific. A user variable that is defined by one client cannot be queried or used by another client. When a client exits, all variables that are specific to the client session are automatically released.

Syntax

```
SET @var_name = expr;
```

- User variables are expressed as @var_name. The variable name var_name consists of alphanumeric characters, decimal points (.), underscores (_), and dollar signs (\$) from the current character set.
- The expression of a variable can be an integer, real number, string, or NULL value.
- If you set multiple user variables at a time, separate the variable definitions with commas (,).

Examples

- Execute the following SET statements to set user variables:

```
Oceanbase>SET @a=2, @b=3;
Query OK, 0 rows affected (0.01 sec)

Oceanbase>SET @c = @a + @b;
Query OK, 0 rows affected (0.00 sec)
```

- Execute the following statement to query user variables:

```
Oceanbase>SELECT @a, @b, @c;
+-----+-----+-----+
| @a | @b | @c |
+-----+-----+-----+
| 2 | 3 | 5 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

- Execute the following SELECT statement to set user variables:

```
Oceanbase> SELECT @a:=2, @b:=3, @c:=@a+@b;
+-----+-----+-----+
| @a:=2 | @b:=3 | @c:=@a+@b |
+-----+-----+-----+
| 2 | 3 | 5 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

18.5.13.17. ALTER SYSTEM database management statements

18.5.13.17.1. Overview

You can execute ALTER SYSTEM statements to send commands to the ApsaraDB for OceanBase system to perform specified operations.

18.5.13.17.2. System-level management statements

Bootstrap the system

```
ALTER SYSTEM BOOTSTRAP
[REGION [=] 'region'] ZONE [=] 'zone' SERVER [=] 'ip:port'
[, [REGION [=] 'region'] ZONE [=] 'zone' SERVER [=] 'ip:port' ...] ;
```

To bootstrap the system, you must specify the root servers. Use commas (,) to separate multiple root servers.

REGION specifies a region. You must specify a region if your ApsaraDB for OceanBase service is deployed across zones in multiple regions.

A cluster can have only one root server to act as the leader to provide services.

Example:

```
ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '10.218.248.178:55410';
```

Separate multiple root servers with commas (,).

```
ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '172.24.65.24:55410', ZONE 'zone2' SERVER '172.24.65.114:55410';
```

Update the system version

```
ALTER SYSTEM BEGIN UPGRADE # Start the system update.  
ALTER SYSTEM END UPGRADE # End the system update.
```

Notes:

1. When the administrator executes `ALTER SYSTEM BEGIN UPGRADE` to update the system version, the leader root server performs a series of checks. For example, it checks the servers for version consistency and modifies the `enable_upgrade_mode` parameter.
2. The leader root server maintains the minimum server version of the cluster. The leader root server analyzes server versions in the `__all_server` table and saves the minimum version in the `min_observer_version` parameter.

 **Note** ApsaraDB for OceanBase supports 3-bit version numbers, such as V1.2.3. In a version number, the first digit represents `observer_major_version`, the second digit represents `observer_minor_version`, and the third digit represents `observer_patch_version`.

3. When a system update begins, the leader root server checks whether all servers in the cluster are of the same version and whether the version is the same as the minimum version.

When the `ALTER SYSTEM END UPGRADE` statement is executed to end the update, the leader root server checks whether all servers are updated and changes the `min_observer_version` value.

Manage the schema

To refresh the schema, execute the following statement:

```
ALTER SYSTEM REFRESH SCHEMA {SERVER='ip:port'} ZONE='zone';
```

When the system executes Data Definition Language (DDL) operations, the leader root server notifies all ODBServers to refresh the schema.

If an exception occurs to an ODBServer, the ODBServer is disconnected from the leader root server. You must manually refresh the schema. You can refresh the schema for an ODBServer or a cluster.

Examples:

- Refresh the schema for an ODBServer.

```
ALTER SYSTEM REFRESH SCHEMA SERVER='172.24.65.24:55410';
```

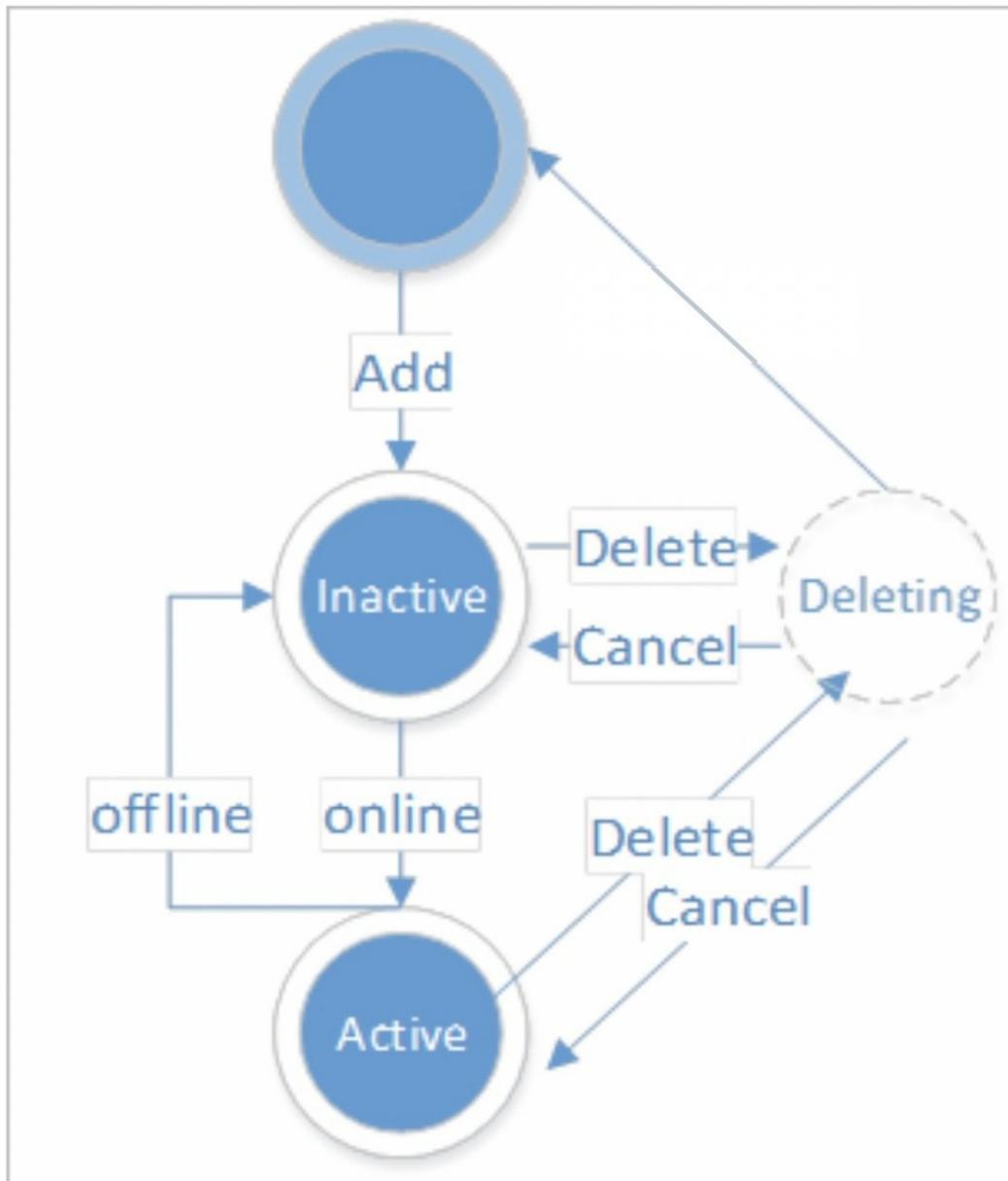
- Refresh the schema for all ODBServers that are deployed in a zone.

```
ALTER SYSTEM REFRESH SCHEMA ZONE='zone1';
```

Manage servers

Server status shows the status of a server when different operations are performed.

Server status

**Notes:**

1. You can execute the `ALTER SYSTEM ADD SERVER` or `ALTER SYSTEM DELETE SERVER` statement to add a server to or delete a server from the server list. Only servers on the list can provide services.
2. When the `ALTER SYSTEM DELETE SERVER` statement is executed, the ApsaraDB for OceanBase system selects a new replica leader and replicates replicas.
3. The `DELETE` operation is time-consuming. You can execute the `ALTER SYSTEM CANCEL DELETE SERVER` statement to abort this operation.

To add a server, execute the following statement:

```
ALTER SYSTEM ADD SERVER 'ip:port' [, 'ip:port'...] [ZONE='zone'];
```

If a zone is specified, the system checks whether the server that you want to add is deployed in the specified zone.

Example:

```
ALTER SYSTEM ADD SERVER '172.24.65.113:55410' ZONE 'zone1';
```

To delete and undelete a server, execute the following statements:

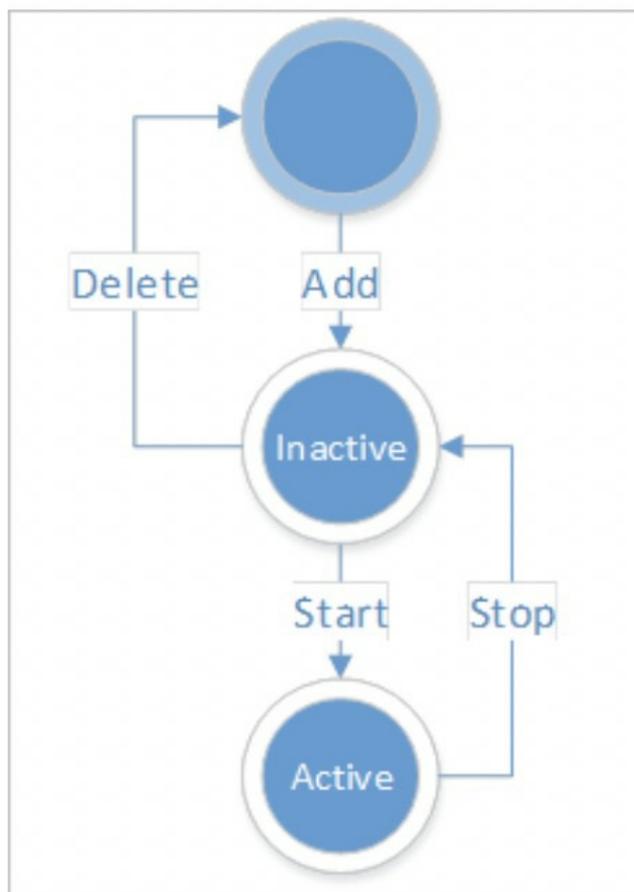
```
ALTER SYSTEM DELETE SERVER 'ip:port' [, 'ip:port!...'] [ZONE[='zone']]
```

```
ALTER SYSTEM CANCEL DELETE SERVER 'ip:port' [, 'ip:port!...'] [ZONE[='zone'];
```

Manage zones

Zone status shows the status of a zone when different operations are performed.

Zone status



- Add a zone.

```
ALTER SYSTEM ADD ZONE 'zone' [REGION 'region'];
```

- Delete a zone.

```
ALTER SYSTEM DELETE ZONE 'zone';
```

- Activate or deactivate a zone.

```
ALTER SYSTEM {START|STOP} ZONE 'zone';
```

Modify the region property for a zone

You can specify the region property for a zone in ApsaraDB for OceanBase V1.3 and later versions.

Syntax:

```
ALTER SYSTEM {ALTER|MODIFY|CHANGE} ZONE 'zone_name' [set] alter_zone_option
alter_zone_option: region [=] 'region_name'
```

Manage sessions

If you are a user of a general tenant, execute the following statement to close a session:

```
KILL [GLOBAL|LOCAL] [CONNECTION] 'sessionid';
```

If you are a system administrator, execute the following statement to close a session:

```
ALTER SYSTEM KILL [GLOBAL|LOCAL] SESSION 'sessionid';
```

Manage partitions

- Select a new replica leader.

```
ALTER SYSTEM
SWITCH REPLICA LEADER | FOLLOWER
(PARTITION_ID='partidx%partcount@tableid' SERVER='ip:port' | SERVER='ip:port' | ZONE='zone');
```

- Delete a replica.

```
ALTER SYSTEM
DROP REPLICA
PARTITION_ID = 'partidx%partcount@tableid'
SERVER = 'ip:port'
[CREATE_TIMESTAMP = ctimestamp]
[ ZONE='zone'];
```

To delete a replica on the specified OBServer, you must specify the following parameters: PARTITION_ID, SERVER, and CREATE_TIMESTAMP.

- Migrate or replicate a replica from one OBServer to another OBServer.

```
ALTER SYSTEM
{MOVE|COPY} REPLICA
PARTITION_ID 'part_idx%part_count@table_id'
SOURCE='ip:port'
DESTINATION='ip:port' ;
```

To migrate or replicate a replica, you must specify the source OBServer, the destination OBServer, and the partition ID.

- Cancel a replica migration or replication task.

```
ALTER SYSTEM
CANCEL [PARTITION MIGRATION] TASK 'task_id';
```

To cancel a migration task, you must specify the task ID. You can query task IDs from the `__all_virtual_sys_task_status` table.

- Report replicas.

```
ALTER SYSTEM
  REPORT REPLICA
  {SERVER = 'ip:port' | ZONE='zone'};
```

This statement requires an OBCServer or all OBCServers in a zone to report replicas.

- Recycle replicas that are not used.

```
ALTER SYSTEM RECYCLE REPLICA {SERVER = 'ip:port' | ZONE='zone'};
```

- Modify the properties of a replica.

```
ALTER SYSTEM
  {alter|change|modify} REPLICA
  PARTITION_ID 'part_idx%part_count@table_id'
  SERVER = 'ip:port'
  [set] change_actions;
change_actions: REPLICA_TYPE = 'replica_type'
```

This statement modifies the type of a specified replica.

The valid values of replica type are FULL, READONLY, and LOGONLY. You can set the value of `replica_type` to the full name or the abbreviation of a valid replica type, such as F, R, or L. The value is not case-sensitive.

Run a system job

To run a system job, execute the following statement:

```
ALTER SYSTEM RUN JOB 'job_name' {SERVER='ip:port' | ZONE='zone'}
```

If the specified job name contains a special character, enclose the job name in single quotation marks ('). In other cases, single quotation marks are optional.

The system supports the following jobs:

- `check_partition_table`

You can run this job to check partitioned tables on an OBCServer.

- `root_inspection`

You can run this job to trigger the leader root server to check the following information:

- Whether the hard-coded schema in the system table is the same as the schema in the internal table.
- Whether the hard-coded system variables are the same as the system variables in the internal table.
- Whether the hard-coded `zone_info` and `sys_stat` are the same as those in the internal table.

Perform daily major freeze operations

You can perform the following daily major freeze operations:

- Initiate a daily major freeze request.

```
ALTER SYSTEM MAJOR FREEZE
```

- Activate a manually triggered major freeze operation.

```
ALTER SYSTEM SET ENABLE_MANUAL_MERGE='True'
```

- Deactivate a manually triggered major freeze operation.

```
ALTER SYSTEM SET ENABLE_MANUAL_MERGE='False'
```

- Start the daily major freeze operation.

```
ALTER SYSTEM START MERGE ZONE='zone';
```

- Suspend the daily major freeze operation.

```
ALTER SYSTEM SUSPEND MERGE [ZONE='zone']
```

- Resume the daily major freeze operation.

```
ALTER SYSTEM RESUME MERGE [ZONE='zone']
```

```
ALTER SYSTEM CLEAR ROOTTABLE [TENANT='tenantname'];
```

Manage memory

You can perform the following memory management operations:

- Initiate a minor freeze request.

```
ALTER SYSTEM MINOR FREEZE
  [{TENANT [=] ('tt1' [, 'tt2'...]) | PARTITION_ID [=] 'partidx%partcount@tableid'}]
  [{SERVER [=] ('ip:port' [, 'ip:port'...])}] ;
```

- Clear a plan cache.

```
ALTER SYSTEM FLUSH PLAN CACHE
```

- Clear KV caches.

```
1. Clear the schema cache of the system tenant.
ALTER SYSTEM FLUSH KVCACHE TENANT='sys' CACHE='schema_cache';

2. Clear the KV caches of the system tenant.
ALTER SYSTEM FLUSH KVCACHE TENANT='sys';

3. Clear the KV caches of all tenants.
ALTER SYSTEM FLUSH KVCACHE;
```

Configure parameters

- Modify system parameters.

```
ALTER SYSTEM [SET] param_name [=] expr
  [COMMENT [=]'text']
  [SCOPE [=] conf_scope]
  {SERVER [=] 'ip:port' | ZONE [=] 'zone'};
```

For more information about how to modify system parameters, see [Clause description](#).

Clause description

Clause	Description
<i>param_name = expr</i>	The name of a system parameter.

Clause	Description
COMMENT ' <i>text</i> '	Optional. This clause is used to add a comment about the modification. We recommend that you add a comment.
SCOPE = <i>conf_scope</i>	<p>The effective range of the modification. The following three values are supported:</p> <ul style="list-style-type: none"> ◦ MEMORY: Only the parameter value in the memory is modified. The modified parameter takes effect immediately after the modification and becomes invalid after the server is restarted. However, no parameter supports this option. ◦ SPFILE: Only the parameter value in the configuration table is modified. The modified parameter takes effect after the server is restarted. ◦ BOTH: The parameter values in the configuration table and in the memory are modified. The modified parameter takes effect immediately after the modification and remains valid after the server is restarted. <p> Note The default value is BOTH. If you specify BOTH or MEMORY for a parameter that cannot take effect immediately after the modification, the system returns an error.</p>
SERVER_TYPE = <i>server_type</i>	The server type. Valid values: ROOTSERVER, UPDATESERVER, CHUNKSERVER, and MERGESERVER.
ZONE = ' <i>zone_name</i> '	The name of the zone, which indicates that the modified parameter applies to the specified type of servers in the specified cluster. If you do not specify the zone name, the modified parameter applies to the specified type of servers in all clusters.
SERVER = ' <i>ip:port</i> '	The IP address and port number of the server, which indicates that only the parameter of a specified server is modified.

If you modify multiple system parameters in one request, separate the parameters with commas (,).

- Query system parameters.

```
SHOW PARAMETERS [LIKE 'pattern' | WHERE expr];
```

Manage time zone information

```
ALTER SYSTEM REFRESH TIME_ZONE_INFO
```

ApsaraDB for OceanBase uses `mysql_tzinfo_to_sql` to generate an SQL script based on the time zone information in the operating system. The system then adds the time zone information into the following four system tables: `__all_time_zone`, `__all_time_zone_name`, `__all_time_zone_transition`, and `__all_time_zone_transition_type`.

The `ALTER SYSTEM REFRESH TIME_ZONE_INFO` statement notifies all servers in the cluster to update the local time zone information based on the related system tables.

Reset the valid flag for a disk

```
ALTER SYSTEM SET DISK VALID SERVER [=] 'ip:port'
```

ApsaraDB for OceanBase automatically checks whether a disk has failed. If a possible disk failure is detected, the system sets a flag to migrate the leader from the current OBServer to another OBServer.

This flag is valid for all partitions on the OBServer.

When the database administrator finds that the disk restores to a normal status, the administrator can execute this statement to reset the flag.

18.5.13.17.3. Tenant-level management statements

Modify tenant-level parameters

```
ALTER SYSTEM SET param_name = expr
  [COMMENT 'text']
  [SCOPE = conf_scope]
  [TENANT = 'tenantname']
```

Enable or disable the migration feature or the replication feature

- Enable or disable the migration feature.

```
ALTER SYSTEM SET ENABLE_REBALANCE = {true|false} [TENANT='tenantname'];
```

- Enable or disable the replication feature.

```
ALTER SYSTEM SET ENABLE_REREPLICATION= {true|false} [TENANT='tenantname'];
```

18.5.14. READ ONLY

18.5.14.1. Overview

ApsaraDB for OceanBase allows you to set the READ ONLY property at the tenant, database, or table level.

General syntax

The following rules apply:

1. To define the READ ONLY property, you can execute the following ALTER statement, which is consistent with Oracle in syntax.

```
ALTER {TENANT | DATABASE | TABLE }
 {tenant_name | database_name | table_name}
 {READ ONLY | READ WRITE}
```

2. To be compatible with MySQL instances, you can execute the `SET @@GLOBAL.READ_ONLY=ON|OFF` statement to set the READ ONLY property for a tenant.
3. After you configure the READ ONLY property at a specific level, the specified objects can be in one of the following states: READ ONLY, READ WRITE, and PARTIALLY READ ONLY. PARTIALLY READ ONLY is an intermediate status, which indicates that the specified objects are in the READ ONLY state on some OBServers.

Notes

When you set the READ ONLY property, follow these rules:

1. The READ ONLY property can apply to the following levels in descending order: tenant level, database level, or table level. The READ ONLY property at a higher level takes precedence over that at a lower level.
For example, if a tenant is in the READ ONLY state, the databases and tables within the tenant are in the READ ONLY state regardless of whether the READ ONLY property is set.
2. To be compatible with MySQL, you must have the SUPER permission to perform read and write operations on the objects that are in the READ ONLY state.

18.5.14.2. Tenant-level READ ONLY property

Configure the READ ONLY property

Use one of the following two methods based on your needs:

- In a system tenant, log on to the system by using a user that has the SUPER permission, and execute the following statement to set the READ ONLY property for all tenants:

```
ALTER TENANT tenant_name {READ ONLY | READ WRITE};
```

- In a general tenant, log on to the system by using a user that has the SUPER permission, and execute one of the following statements to set the READ ONLY property for the current tenant:

```
SET GLOBAL READ_ONLY={ON | OFF};  
Or  
SET @@GLOBAL.READ_ONLY={ON | OFF};
```

Query the READ ONLY property

Use one of the following three methods to query the READ ONLY property of the current tenant:

- Execute the `SHOW TENANT STATUS` statement.

```
SHOW TENANT STATUS;
```

Example:

```
OceanBase (admin@test)> show tenant status;  
+-----+-----+  
| Tenant | State |  
+-----+-----+  
| sys   | read only |  
+-----+-----+  
1 row in set (0.01 sec)
```

- Query the internal table `oceanbase.__tenant_virtual_tenant_status`.

```
OceanBase (admin@test)> select * from oceanbase.__tenant_virtual_tenant_status;  
+-----+-----+-----+-----+  
| tenant | host   | port | read_only |  
+-----+-----+-----+-----+  
| sys   | 172.24.65.24 | 55410 | 1 |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

You can query the status of the current tenant on all OBServers based on the `oceanbase.__tenant_virtual_tenant_status` table.

- Query the `read_only` system variable.

```
show global variables like 'read_only'
```

Or

```
select @@global.read_only
```

18.5.14.3. Database-level READ ONLY property

Configure the READ ONLY property

```
ALTER DATABASE database_name {READ ONLY | READ WRITE};
```

Query the READ ONLY property

Use one of the following two methods to query the READ ONLY property of the databases under the current tenant:

- Execute the `SHOW DATABASES STATUS` statement to query the READ ONLY property of the specified databases.

The system returns the status information for each database. In the output, one column shows whether each database is in the READ ONLY state. Each database supports three states: READ ONLY, READ WRITE, and PARTIALLY READ ONLY. READ ONLY indicates that this database is only readable. READ WRITE indicates that this database is readable and writable. PARTIALLY READ ONLY indicates an intermediate status where the database is in the READ WRITE state on some OBServers and in the READ ONLY state on the other OBServers.

You can include a LIKE or WHERE clause in this statement. The clause is used to specify the databases for which you want to query the status.

- Query the internal table `oceanbase.__tenant_virtual_database_status`.

```
OceanBase (admin@test)> select * from oceanbase.__tenant_virtual_database_status;
+-----+-----+-----+-----+
| db   | host   | port | read_only |
+-----+-----+-----+-----+
| mysql | 172.24.65.24 | 55410 | 0 |
| test  | 172.24.65.24 | 55410 | 0 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

You can query the status of all databases under the current tenant on all OBServers based on the `oceanbase.__tenant_virtual_database_status` table.

18.5.14.4. Table-level READ ONLY property

Configure the READ ONLY property

```
ALTER TABLE table_name [SET] {READ ONLY | READ WRITE}
```

Query the READ ONLY property

Use one of the following two methods to query the READ ONLY property of tables:

- Execute the `SHOW TABLE STATUS` statement.

```
SHOW TABLE STATUS
```

- Query the internal table `oceanbase.__tenant_virtual_table_status`.

You can query the status of all tables in all databases under the current tenant on all OBServers based on the `oceanbase.__tenant_virtual_table_status` table.

18.5.15. Other SQL statements

18.5.15.1. SHOW statements

You can execute SHOW statements to query information such as the status of databases, tables, columns, or servers in the ApsaraDB for OceanBase system.

SHOW CHARACTER SET

```
SHOW CHARACTER SET
[LIKE 'pattern' | WHERE expr]
```

The `SHOW CHARACTER SET` statement returns information about all available character sets.

This statement can contain a LIKE or WHERE clause.

The clause is used to specify the names of character sets or columns that you want to match.

```
mysql> SHOW CHARACTER SET WHERE charset='utf8mb4';
+-----+-----+-----+-----+
| Charset | Description | Default collation | Maxlen |
+-----+-----+-----+-----+
| utf8mb4 | UTF-8 Unicode | utf8mb4_general_ci | 4 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SHOW CHARACTER SET like 'utf8%';
+-----+-----+-----+-----+
| Charset | Description | Default collation | Maxlen |
+-----+-----+-----+-----+
| utf8 | UTF-8 Unicode | utf8_general_ci | 3 |
| utf8mb4 | UTF-8 Unicode | utf8mb4_general_ci | 4 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

SHOW COLLATION

```
SHOW COLLATION
[LIKE 'pattern' | WHERE expr]
```

The `SHOW COLLATION` statement returns all available collations.

This statement can contain a LIKE or WHERE clause.

The clause is used to specify the names of character sets or columns that you want to match.

```
mysql> SHOW COLLATION LIKE 'utf8mb4_bin%';
+-----+-----+-----+-----+-----+
| Collation | Charset | Id | Default | Compiled | Sortlen |
+-----+-----+-----+-----+-----+
| utf8mb4_bin | utf8mb4 | 46 | | Yes | 1 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> SHOW COLLATION WHERE charset='utf8mb4';
+-----+-----+-----+-----+-----+
| Collation      | Charset | Id | Default | Compiled | Sortlen |
+-----+-----+-----+-----+-----+
| utf8mb4_general_ci | utf8mb4 | 45 | Yes | Yes | 1 |
| utf8mb4_bin      | utf8mb4 | 46 | | Yes | 1 |
+-----+-----+-----+-----+-----+
```

SHOW COLUMNS

```
SHOW [FULL] COLUMNS {FROM | IN} tblname [{FROM | IN} dbname]
[LIKE 'pattern' | WHERE expr]
```

The `SHOW COLUMNS` statement returns information about the columns of a specified table.

You can also execute this statement to return column information for a specified view.

`SHOW FIELDS` is equivalent to `SHOW COLUMNS`.

- If the `FULL` keyword is specified, the output contains the permissions that you have and the comments for each column.
- The `dbname.tblname` syntax is an alternative to `tblname FROM dbname`. These two statements are equivalent.

Example:

```
mysql> SHOW COLUMNS FROM mytable FROM mydb;

mysql> SHOW COLUMNS FROM mydb.mytable;
```

```
mysql> SHOW COLUMNS IN users IN test;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra  |
+-----+-----+-----+-----+-----+
| name  | varchar(30) | YES  |     | NULL    |       |
| class | varchar(30) | YES  |     | NULL    |       |
| hometown | varchar(30) | YES  |     | NULL    |       |
| id    | int(11)  | NO   | PRI | NULL    | auto_increment |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> SHOW FULL COLUMNS IN users IN test;
+-----+-----+-----+-----+-----+-----+-----+
| Field | Type  | Collation | Null | Key | Default | Extra  | Privileges          | Comment |
+-----+-----+-----+-----+-----+-----+-----+
| name  | varchar(30) | utf8_general_ci | YES  |     | NULL    |       | select,insert,update,references |
| class | varchar(30) | utf8_general_ci | YES  |     | NULL    |       | select,insert,update,references |
| hometown | varchar(30) | utf8_general_ci | YES  |     | NULL    |       | select,insert,update,references |
| id    | int(11)  | NULL      | NO   | PRI | NULL    | auto_increment | select,insert,update,references |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

SHOW CREATE DATABASE

```
SHOW CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] dbname
```

The SHOW CREATE DATABASE statement returns the CREATE DATABASE statement that creates a specified database. You can also execute the SHOW CREATE SCHEMA statement to accomplish the same effect.

SHOW DATABASES STATUS

```
SHOW DATABASES STATUS [LIKE 'pattern' | WHERE expr]
```

The SHOW DATABASES STATUS statement returns the status of databases. In the output, one column shows whether a database is read only. A database supports the following states: READ ONLY, READ WRITE, and PARTIALLY READ ONLY. READ ONLY indicates that the database is only readable. READ WRITE indicates that the database is readable and writable. PARTIALLY READ ONLY indicates an intermediate status where the database is in the READ WRITE state on some OBServers and in the READ ONLY state on the other OBServers.

This statement can contain a LIKE or WHERE clause. The clause is used to specify the databases that you want to match.

SHOW CREATE TABLE

```
SHOW CREATE TABLE tblName
```

The SHOW CREATE TABLE statement returns the CREATE TABLE statement that creates a specified table.

You can also execute this statement to return the CREATE TABLE statement that creates a specified view.

SHOW CREATE VIEW

```
SHOW CREATE VIEW viewname
```

The `SHOW CREATE VIEW` statement returns the `CREATE VIEW` statement that creates a specified view.

SHOW DATABASES

```
SHOW {DATABASES | SCHEMAS}
    [LIKE 'pattern' | WHERE expr]
```

The `SHOW DATABASES` statement returns information about ApsaraDB for OceanBase databases.

You can query information about only databases on which you have permissions unless you have the global `SHOW DATABASES` permission.

`SHOW SCHEMAS` is equivalent to `SHOW DATABASES`.

This statement can contain a `LIKE` or `WHERE` clause.

The clause is used to specify the databases that you want to match.

SHOW ERRORS

```
SHOW ERRORS [LIMIT [offset,] row_count]
SHOW COUNT(*) ERRORS
```

The `SHOW ERRORS` statement is similar to the `SHOW WARNINGS` statement in the effect. The difference is that the `SHOW ERRORS` statement returns only errors, whereas the `SHOW WARNINGS` statement also returns warnings and notes.

The `SHOW COUNT(*) ERRORS` statement returns the number of errors.

SHOW WARNINGS

```
SHOW WARNINGS [LIMIT [offset,] row_count]
SHOW COUNT(*) WARNINGS
```

`SHOW WARNINGS [LIMIT [offset,]]` The `SHOW WARNINGS` statement returns the errors, warnings, and notes that were generated during the execution of the previous statement. The `LIMIT` clause specifies the number of rows that can be displayed.

The `SHOW COUNT(*) WARNINGS` statement returns the total number of errors, warnings, and notes.

SHOW GRANTS

```
SHOW GRANTS [FOR user]
```

The `SHOW GRANTS` statement returns information about the permissions that have been granted to an ApsaraDB for OceanBase user.

If you do not specify the username, this statement returns the permissions that have been granted to the current user.

SHOW INDEX

```
SHOW {INDEX | INDEXES | KEYS}
      {FROM | IN} tblname
      [{FROM | IN} dbname]
      [WHERE expr]
```

The `SHOW INDEX` statement returns information about the indexes of a table.

The `dbname.tblname` syntax is an alternative to `tblname FROM dbname`.

The following two statements are equivalent:

```
mysql> SHOW INDEX FROM mytable FROM mydb;
```

```
mysql> SHOW INDEX FROM mydb.mytable;
```

SHOW PRIVILEGES

```
SHOW PRIVILEGES
```

The `SHOW PRIVILEGES` statement returns the system permissions that are supported by ApsaraDB for OceanBase.

SHOW PROCESSLIST

```
SHOW [FULL] PROCESSLIST
```

The `SHOW PROCESSLIST` statement returns the information about running threads.

If you have the `SUPER` permission, you can view all threads. If you do not have the `SUPER` permission, you can view only your own threads.

If the `FULL` keyword is not specified, this statement returns only the first 100 characters of each query.

SHOW STATUS

```
SHOW [GLOBAL | SESSION] STATUS
      [LIKE 'pattern' | WHERE expr]
```

The `SHOW STATUS` statement returns information about server status.

If you specify the `GLOBAL` modifier, this statement returns the status values for all connections to the ApsaraDB for OceanBase system. If you specify the `SESSION` modifier, this statement returns status values for the current connection. If neither of the two modifiers is specified, the default is `SESSION`.

`LOCAL` is equivalent to `SESSION`.

Note

Some status variables have only global values.

For these variables, this statement returns the same result regardless of whether you use the `GLOBAL` or `SESSION` modifier.

SHOW TABLE STATUS

```
SHOW TABLE STATUS [{FROM | IN} dbname]
[LIKE 'pattern' | WHERE expr]
```

The `SHOW TABLE STATUS` statement is similar to the `SHOW TABLES` statement. However, the `SHOW TABLE STATUS` statement returns detailed information about each non-temporary table.

You can also execute this statement to return information about views.

SHOW TABLES

```
SHOW [FULL] TABLES [{FROM | IN} dbname]
[LIKE 'pattern' | WHERE expr]
```

The `SHOW TABLES` statement lists the non-temporary tables in a specified database.

This statement also lists the views in the database.

The `FULL` keyword is optional. If this keyword is specified, the `SHOW FULL TABLES` statement returns the second output column.

For a table, the value of the second column is `BASE TABLE`. For a view, the value of the second column is `VIEW`. If you do not have permissions on a table, this table is not displayed in the output from the `SHOW TABLES` statement.

SHOW VARIABLES

```
SHOW [GLOBAL | SESSION] VARIABLES
[LIKE 'pattern' | WHERE expr]
```

The `SHOW VARIABLES` statement returns the values of system variables that are supported by ApsaraDB for OceanBase.

If you specify the `GLOBAL` modifier, this statement returns the values that are used for new connections to the ApsaraDB for OceanBase system. If you specify the `SESSION` modifier, this statement returns the values that are in effect for the current connection. If neither of the two modifiers is specified, the default is `SESSION`.

`LOCAL` is equivalent to `SESSION`.

If you specify a `LIKE` or `WHERE` clause, this statement returns only variables that match the specified condition pattern.

SHOW PARAMETERS

```
SHOW PARAMETERS
[LIKE 'pattern' | WHERE expr]
```

The `SHOW PARAMETERS` statement returns the value of each parameter.

SHOW CREATE TENANT

```
SHOW CREATE TENANT tenantname
```

The `SHOW CREATE TENANT` statement returns the information about a specified tenant.

```
SHOW CREATE TENANT test;
+-----+-----+
| Tenant | Create Tenant |
+-----+-----+
| test | CREATE TENANT test
charset='utf8mb4', replica_num=1, zone_list=('zone1'), primary_zone='zone1', resource_pool_list=('p1'); |
+-----+-----+
1 row in set (0.00 sec)
```

SHOW TENANT

```
SHOW TENANT
```

The SHOW TENANT statement returns the information about the current tenant.

```
SHOW TENANT
+-----+
| CURRENT TENANT NAME |
+-----+
| test|
+-----+
1 row in set (0.00 sec)
```

SHOW TABLEGROUPS statement

```
SHOW TABLEGROUPS
[LIKE 'pattern' | WHERE expr]
```

The SHOW TABLEGROUPS statement returns the information about all table groups that are created in the current tenant.

This statement can contain a LIKE or WHERE clause.

The clause is used to specify the table groups you want to match. Then, you can query the tables that are contained in each specified table group.

18.5.15.2. KILL statement

```
KILL [GLOBAL | LOCAL] [CONNECTION | QUERY] 'sessionid'
```

Each connection to the ApsaraDB for OceanBase system runs in an independent thread. To query the running threads, execute the SHOW PROCESSLIST; statement. To kill a thread, execute the KILL sessionid statement.

When you execute the KILL statement, follow these rules:

- The KILL CONNECTION statement closes the connection that is associated with the specified thread. This is the same as a KILL statement that does not contain a modifier.
- The KILL QUERY statement terminates the statement that is being executed over the connection. The connection state remains unchanged.

If you have the PROCESS permission, you can view all threads.

If you have the SUPER permission, you can kill all threads and statements. If you do not have the PROCESS or SUPER permission, you can view or kill only your own threads and statements.

18.5.15.3. DESCRIBE statement

```
{DESCRIBE | DESC | EXPLAIN} tblname [colname | wild];
```

This statement is equivalent to the `SHOW COLUMNS FROM` statement.

Example:

```
mysql> DESCRIBE city;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| Id    | int(11) | NO  | PRI | NULL    | auto_increment |
| Name  | char(35) | NO  |    |         |              |
| Country | char(3) | NO  | UNI |         |              |
| District | char(20) | YES | MUL |         |              |
| Population | int(11) | NO  |    | 0      |              |
+-----+-----+-----+-----+-----+-----+
```

Notes:

- The Key field indicates whether the column is indexed.

Valid values:

- PRI: indicates that the column is a primary key or is one of the columns in a multiple-column primary key of the table.
- UNI: indicates that the column is part of a unique index.
- MUL: indicates that a given value can appear in the column multiple times.

If a unique index is a composite unique index that consists of multiple columns, the value MUL is displayed in the unique index. Although the combination of columns is unique, each column may contain a given value that appears multiple times. Note that in a composite index, only the leftmost column of the index can be included in the Key field.

- The Default field indicates whether a default value is assigned to the column.
- The Extra field lists additional information about the given column.

In this example, the Extra field indicates that the AUTO_INCREMENT keyword is used when the Id column is created.

18.5.15.4. USE statement

Syntax

```
USE dbname
```

`USE The USE dbname` statement informs the client to use a specified database as the default database for subsequent statements. This database acts as the default database until the session ends or another USE statement is issued.

Examples

```
mysql> USE db1;
mysql> SELECT COUNT(*) FROM mytable; # selects from db1.mytable
mysql> USE db2;
mysql> SELECT COUNT(*) FROM mytable; # selects from db2.mytable
```

If you do not execute the USE statement to specify the current database, the following error is returned:

```
root@(none) 04:59:12>create table t1 (a int);
ERROR 1046 (3D000): No database selected
root@(none) 04:59:24>select * from t1;
ERROR 1046 (3D000): No database selected
```

If you execute the USE statement to specify a database as the current database, you can still access tables in other databases.

The following example shows how to access the author table from the db1 database and the editor table from the db2 database:

```
mysql> USE db1;
mysql> SELECT author_name,editor_name FROM author,db2.editor
-> WHERE author.editor_id = db2.editor.editor_id;
```

Errors

If the database name that follows the USE keyword is not found, the system returns the following error:

```
ERROR 1049 (42000): Unknown database 'test1' .
```

18.5.15.5. Hints

Syntax

Hints are a special type of comments for SQL statements. The general syntax for comments is `/* ... */`. This is the same as the syntax for C language comments. Based on the general comment syntax, the hint syntax adds a plus sign (+) that follows the `/*` comment opening sequence and is displayed as `/*+ ... */`.

If the server does not recognize the hint in your SQL statement, it ignores the hint and does not return an error. Therefore, when an SQL statement is distributed to different databases, the hints in this statement do not cause syntax errors.

Hints only affect the internal optimization logic of the database server. They do not affect the semantics of SQL statements.

ApsaraDB for OceanBase is compatible with Oracle in hint syntax, except for some unique hints that are specific to ApsaraDB for OceanBase databases.

Hints that are supported by ApsaraDB for OceanBase have the following characteristics:

- Some hints do not contain parameters, for example, `/*+ KAKA */`.
- Some hints contain parameters, for example, `/*+ HAHA(param) */`.
- Multiple hints can be included in one hint comment and are separated with commas (,). For example, `/*+ KAKA A, HAHA(param) */`.
- A hint in the SELECT statement must follow the SELECT keyword and precede other keywords. For example, `SELECT /*+ KAKA */ ...`.
- A hint in the UPDATE statement or the DELETE statement must follow the UPDATE or DELETE keyword.

Specify a valid subquery

In data manipulation language (DML) statements, QB_NAME indicates the name of a query block, which uniquely identifies the query block. The query block name can be system-generated or user-specified.

If you do not specify QB_NAME in the hint, the system assigns SEL\$1, SEL\$2, UPD\$1, and DEL\$1 to query blocks in sequence from left to right. The system also uses the same sequence to resolve these query block names.

You can find each table based on QB_NAME, or specify the actions on a specific query block.

You can specify QB_NAME in TBL_NAME to find the table. The first QB_NAME in the hint allows you to find the query block to which the hint applies.

For example, you can specify the primary key and the index for outer queries and subqueries in the following scenario:

```
OceanBase (root@test)> create table t1(c1 int, c2 int, key t_c1(c1));
select /*+INDEX(t@SEL$1 PRIMARY) INDEX(@SEL$2 t@SEL$2 t_c1)*/ *
from t ,
      (select * from t where c2 = 1) ta
where t.c1 = 1;

Query Plan: =====
|ID|OPERATOR          |NAME |EST. ROWS|COST |
-----
|0|NESTED-LOOP INNER JOIN CARTESIAN|    |1      |16166|
|1|TABLE SCAN         |t   |1      |1397 |
|2|TABLE SCAN         |t(t_c1)|1    |14743|
=====

Outputs & filters:
-----
0 - output([t.c1], [t.c2], [t.c1], [t.c2]), filter(nil),
   conds(nil), nl_params_(nil)
1 - output([t.c1], [t.c2]), filter([t.c1 = 1]),
   access([t.c1], [t.c2]), partitions(p0)
2 - output([t.c2], [t.c1]), filter([t.c2 = 1]),
   access([t.c2], [t.c1]), partitions(p0)
```

The following table lists the hints that are supported by ApsaraDB for OceanBase.

Hints

hint	Parameter	Applicable statement	Description
------	-----------	----------------------	-------------

hint	Parameter	Applicable statement	Description
READ_CONSISTENCY	WEAK, STRONG, or FROZEN	SELECT	<p>Indicates whether to perform strongly consistent reads or weakly consistent reads. If the value is not specified in the SQL statement, the default value depends on the value of the system variable <code>ob_read_consistency</code>.</p> <p>FROZEN indicates that the system reads the data from the latest frozen point. The frozen version is automatically selected by the system.</p>
FROZEN_VERSION	Frozen version	SELECT	Reads only the data of the specified version.
READ_ZONE	FOLLOWER or LEADER	SELECT	<p>Specifies the cluster to which the request is sent.</p> <p><code>READ_ZONE(FOLLOWER)</code> indicates that the system preferentially sends the request to a secondary cluster regardless of whether a secondary cluster exists.</p> <p>This hint only controls the routing policy of an ApsaraDB for OceanBase client on SQL statements. It is not associated with <code>READ_CONSISTENCY</code>.</p>
QUERY_TIMEOUT	Time-out period in microseconds	All DML statements	Defines the timeout period for SQL statements when they are executed on the server. After a statement times out, the server terminates the execution and returns a timeout error code.
USE_NL	USE_NL, which specifies the name of the right physical table	SELECT	Performs a NESTED LOOP JOIN to join two tables. For more information, see the usage of Oracle statements.

hint	Parameter	Applicable statement	Description
USE_MERGE	Table name	SELECT	In a MERGE JOIN, the system first sorts each table you want to join based on the column to be joined. Then, the system extracts data from the associated sorted tables to another sorted table to match data. MERGE JOINS require multiple sorts and are resource-consuming.
LEADING	Table name	SELECT	In a multi-table join query, this hint specifies the driving table and the table to which the optimizer firstly accesses.
ORDERED	-	SELECT	The system selects the driving table based on the sequence of the tables that follow the FROM keyword.
USE_PLAN_CACHE	DEFAULT / NONE	All DML statements	Defines an execution plan to be used.
INDEX	INDEX, which is in the <TableName IndexName> format	All DML statements	<p>Similar to the INDEX hint in Oracle, this hint specifies that the data from the specified table must be queried based on the index name. If the index does not exist or is unavailable, the system does not return an error.</p> <p>Example:</p> <pre>SELECT /*+ INDEX(t1 i1) , INDEX(t2 i2)*/ from t1, t2 WHERE t1.c1=t2.c1;</pre> <pre>DELETE /*+ INDEX(t1 i1) */ from t1 WHERE t1.c1= 1;</pre>

hint	Parameter	Applicable statement	Description
PARALLEL	PARALLEL(N)	SELECT	<p>Specifies the maximum number of SQL statements that can be concurrently executed.</p> <p>Only SELECT statements that meet one of the following conditions can be concurrently executed:</p> <ul style="list-style-type: none"> • SELECT statements that contain an aggregate function • SELECT statements that contain a GROUP BY clause • SELECT statements that contain an ORDER BY clause • SELECT statements that do not contain a LIMIT clause <p>Example: <code>select /*+ parallel(5) */ count(*) from t1;</code></p>

hint	Parameter	Applicable statement	Description
TOPK	TOPK(param1 param2)	SELECT	<p>Obtains an approximate value.</p> <p>On-Line Analytical Processing (OLAP) queries that retrieve large amounts of data are time-consuming. In some cases where accuracy is not the major concern, you can reduce the query time by setting a lower accuracy rate.</p> <pre>select /*+ topk(90 1000) */ sum(c2), c1 from t1 group by c1 order by sum(c2) limit 10</pre> <p>In the TOPK(param1 param2) syntax, note that no comma exists between the two parameters. The first parameter is an integer that ranges from 0 to 100. The value of the first parameter indicates the accuracy rate of results. For example, 100 indicates full accuracy, and 90 indicates 90% accuracy. A higher value of the first parameter indicates that the result is more accurate, but the query process consumes a longer length of time.</p> <p>The second parameter is a positive integer that starts from 1. This value is related to the execution of the SQL statement. This approximation algorithm takes effect for only queries that contain the GROUP BY, ORDER BY, and LIMIT clauses.</p>

hint	Parameter	Applicable statement	Description
LOG_LEVEL	<ul style="list-style-type: none"> • ERROR • USER_ERROR • WARN • INFO • TRACE • DEBUG 	All DML statements	<p>Defines the log level for a statement.</p> <p>Example:</p> <pre>select /*+log_level('debug')*/ * from t;</pre> <pre>select /*+log_level('sql.*:debug, common.*:info')*/ * from t;</pre>
USE_LATE_MATERIALIZATION NO_USE_LATE_MATERIALIZATION	N/A	N/A	<p>USE_LATE_MATERIALIZATION: generates a late materialization plan for a query.</p> <p>NO_USE_LATE_MATERIALIZATION: generates a materialization plan for a query. This plan is not a late materialization plan.</p> <p>ApsaraDB for OceanBase V1.0 provides the late materialization feature. The system determines whether to automatically rewrite an index scan in the form of <code>self join</code> based on the cost. You can use this feature if you need to rewrite a large number of SQL statements in the form of <code>self join</code>.</p>
USE_HASH NO_USE_HASH	USE_HASH(relation1 [comma] relation2) NO_USE_HASH(relation1 [comma] relation2)	SELECT	Specifies whether to use the <code>hash join</code> algorithm to join tables.

18.5.15.6. HELP statements

Syntax

- Query the help information for all SQL queries that are supported by ApsaraDB for OceanBase.

```
HELP
```

- Query the help information for the syntax that is used at the SQL server.

```
HELP contents
```

- Query the help information for a specified SQL query.

```
HELP searchstring
```

Examples

- Statement: `mysql> help;`

Output:

```
For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Enterprise support, training, or other products, visit:
  https://shop.mysql.com/

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?      (\?) Synonym for `help'.
clear  (\c) Clear the current input statement.
connect (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit   (\e) Edit command with $EDITOR.
ego    (\G) Send command to mysql server, display result vertically.
exit   (\q) Exit mysql. Same as quit.
go     (\g) Send command to mysql server.
help   (\h) Display this help.
nopager (\n) Disable pager, print to stdout.
notee  (\t) Don't write into outfile.
pager  (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print  (\p) Print current command.
prompt (\R) Change your mysql prompt.
quit   (\q) Quit mysql.
rehash (\#) Rebuild completion hash.
source (\.) Execute an SQL script file. Takes a file name as an argument.
status (\s) Get status information from the server.
system (\!) Execute a system shell command.
tee    (\T) Set outfile [to_outfile]. Append everything into given outfile.
use    (\u) Use another database. Takes database name as argument.
charset (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.

For server side help, type 'help contents'
ERROR 1046 (3D000): No database selected
root@(none) 04:59:24>select * from t1;
```

- Statement: `mysql> help contents;`

Output:

```
You asked for help about help category: "Contents"
For more information, type 'help <item>', where <item> is one of the following
categories:
  Data Types
  Functions
  Operator
  Escape character
  Data Definition
  Data Manipulation
  Transaction Statements
  Prepared Statements
  Compound Statements
  Administration
  Utility
```

- **Statement:** `mysql> help use`

Output:

```
Name: 'USE'
Description:
```

18.5.16. OUTLINE

18.5.16.1. Overview

You can use the outline feature to specify throttling rules or execution plans for SQL statements.

When you test statements that include hints, you must enable the `-c` option on the MySQL client. If the `-c` option is disabled, the statements do not take effect.

18.5.16.2. CREATE OUTLINE

You can execute the `CREATE OUTLINE` statement to create an outline. Based on the outline, the optimizer of ApsaraDB for OceanBase generates a throttling rule or an execution plan for the specified SQL statements.

Syntax

```
CREATE [OR REPLACE] OUTLINE outline_name ON stmt [ TO target_stmt ];
```

where:

- The `stmt` parameter specifies a data manipulation language (DML) statement that contains a hint.
- The hint in `stmt` determines whether to implement the throttling or plan stability feature.
- `TO target_stmt` specifies the SQL statements on which you want to implement the throttling or plan stability feature.
- `OR REPLACE` indicates that you can replace an existing execution plan or a throttling rule.

 **Note** A throttling rule and an execution plan can replace each other.

Plan stability

The plan stability feature is used to generate the specified execution plans for SQL statements. After you create an outline to enable the plan stability feature, the optimizer of ApsaraDB for OceanBase generates an execution plan based on the specified hint for the specified SQL statements.

Notes:

- If the specified hint is not `MAX_CONCURRENT(NUM)`, the `CREATE OUTLINE` statement enables the plan stability feature instead of the throttling feature.
- If you create an outline to enable the throttling feature, you can specify question marks (`?`). However, if you create an outline to enable the plan stability feature, you cannot specify question marks (`?`). When the `CREATE OUTLINE` statement is executed, the portions that can be represented by parameters are converted to question marks (`?`). If you specify question marks (`?`) in this scenario, the system returns the following error: `invalid outline .`
- One `outline_name` is associated with only one execution plan.
- If you do not specify a hint in `stmt` for the `CREATE OUTLINE` statement, the statement creates an outline to enable the plan stability feature. In this scenario, the optimizer of ApsaraDB for OceanBase generates the execution plan based on the outline.

Example:

```
OceanBase (root@oceanbase)> create outline ol_1 on select /*+index(t1 c1)*/ from t1 where c1 =1;
```

In this example, the `ol_1` outline is created. For the statements that meet the `SELECT * FROM t1 WHERE c1 =?` condition, execution plans are generated based on `/*+ BEGIN_OUTLINE_DATA INDEX(@"SEL$1" "oceanbase.t1"@"SEL$1" "c1") END_OUTLINE_DATA */`.

Throttling

If you specify the `MAX_CONCURRENT(NUM)` hint in the `CREATE OUTLINE` statement, the statement applies a throttling rule on the specified SQL statements. The throttling rule controls the maximum number of SQL statements that can be concurrently executed in an OBCServer.

Notes:

- Similar to the plan stability feature, the throttling feature is implemented based on the specified hint. To use the throttling feature, specify the `MAX_CONCURRENT(NUM)` hint. In the hint, `NUM` specifies the maximum number of SQL statements that can be concurrently executed.

If the number of concurrent SQL statements exceeds the upper limit, the server returns the following error:

```
SQL
reach max concurrent num .
```

- You can execute the `CREATE OUTLINE` statement to enable the throttling or plan stability feature. To ensure the correctness and clarity of semantics, do not specify the throttling hint together with other types of hints in one `CREATE OUTLINE` statement.
- When you create throttling rules, you can use question marks (`?`) to distinguish variable parameters and constant parameters.

 **Note** If you execute the `CREATE OUTLINE` statement to enable the plan stability feature, you cannot specify question marks (`?`). In this scenario, you must use parameters to represent constants.

- One `outline_name` can be associated with multiple throttling rules that have the same signature.
- Each throttling rule specifies the maximum number of concurrent statements. If an SQL statement matches multiple throttling rules, the system uses the throttling rule that specifies the smallest value among the upper limits.

Example:

```
OceanBase (root@oceanbase)> create outline ol_1 on select /*+max_concurrent(1)*/ * from t2 where c1 = 1 and c2 = ? ;
```

In this example, the ol_1 outline is created. For the SQL statements that meet the `SELECT * FROM t2 WHERE c1 = 1 and c2 = ?` condition, only one of the statements is executed on a single OBServer.

`c2 = ?` indicates that the question mark can be replaced with a constant value.

For example, throttling is implemented on the following SQL statements:

```
select * from t2 where c1 = 1 and c2 = 1;
select * from t2 where c1 = 1 and c2 = 2;
select * from t2 where c1 = 1 and c2 = "2";
select * from t2 where c1 = 1 and c2 = true;
```

`c1=1` indicates that throttling is implemented on only SQL statements that contain `c1=1`. For example, throttling is not implemented on the following SQL statements:

```
select * from t2 where c1 = 2 and c2 = 1;
select * from t2 where c1 = "2" and c2 = 2;
select * from t2 where c1 = false and c2 = "2";
```

18.5.16.3. ALTER OUTLINE

Syntax

```
ALTER OUTLINE outline_name ADD stmt [ TO target_stmt ]
```

You can use `TO target_stmt` in this statement and `TO target_stmt` in the `CREATE OUTLINE` statement in the same way. You can specify `TO target_stmt` if you need to implement the throttling or plan stability feature on the SQL statements that contain the corresponding hints.

Examples

- Execute the ALTER OUTLINE statement to add a throttling rule.

```
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 = 1 and c2 = ? ;
OceanBase (root@oceanbase)> alter outline ol_1 add select /*+max_concurrent(1)*/ * from t1 where c1 = ? and c2 = 1;
```

- Execute the ALTER OUTLINE statement to add an execution plan.

```
OceanBase (root@oceanbase)> create outline ol_2 on select /*+max_concurrent(1)*/ * from t1,t2 where t1.c1 = 1;
OceanBase (root@oceanbase)> alter outline ol_2 add select /*+use_nl(t2)*/ * from t1,t2 where t1.c1 = 1;
```

Notes

- One outline_name is associated with only one execution plan. If you execute the `CREATE OUTLINE` statement to specify an execution plan, you cannot add another execution plan by executing the ALTER OUTLINE statement.
- When you execute the `ALTER OUTLINE` statement, you can specify only a throttling rule or an execution plan

for each execution. The similar rule applies to the `CREATE OUTLINE` statement.

- To ensure that the `ALTER OUTLINE` statement is valid, the value of the `outline_name` parameter must match the signature of the outline.

18.5.16.4. DROP OUTLINE

You can execute the `DROP OUTLINE` statement to drop an outline for an ApsaraDB for OceanBase database.

Syntax

```
DROP OUTLINE outline_name;
```

Examples

```
DROP OUTLINE ol_1;
```

Errors

- If the specified outline does not exist, the ApsaraDB for Oceanbase database returns the following error: `ERROR HY000: Outline 'XXX' doesn't exist` .
- If no database is selected for the current session, the ApsaraDB for Oceanbase database returns the following error: `ERROR 3D000: No database selected` .

18.5.16.5. Others

18.5.16.5.1. Considerations

Considerations on tenants

All outline-related features such as throttling and plan stability take effect within a tenant. You cannot use the system tenant to create, modify, or drop outlines for other tenants. To create an outline for a tenant, you must execute the `CREATE OUTLINE` statement within this tenant.

Considerations on databases

When you create an outline, the namespace of `stmt` is parsed based on the database of the current session. The throttling or plan stability feature takes effect only if the specified SQL statements are executed in this database.

The following scenarios are two examples:

- Assume that you execute the `CREATE OUTLINE ol_1 ON SELECT /*+use_nl(t1)*/ * FROM t2,t1` statement in the `db1` database. When you execute the `SELECT * FROM t2` statement in database `db2`, the query result is not affected by the `ol_1` outline.
- If an SQL statement is executed and no database is specified, you cannot create an outline for the SQL statement.

Considerations on MySQL clients

Assume that you execute the `CREATE OUTLINE` statement on a MySQL client. The MySQL client adds a space between the hint and the first `SELECT` expression in the statement if the two elements are not separated with characters. The test results show that this issue does not occur if Java and Python are used to establish connections.

The following code shows an example:

```
OceanBase (root@oceanbase)> create outline ol_1 on select /*+full(t1)*/ * from t1;
OceanBase (root@oceanbase)> select sql_text from __all_outline;
+-----+
| sql_text          |
+-----+
| select /*+full(t1)*/ * from t1 |
+-----+
1 row in set (0.60 sec)
```

Concurrency

The concurrency settings limit the number of SQL statements that can be concurrently executed on a single ODBServer based on physical execution plans. Therefore, the maximum number of the specified SQL statements that can be concurrently executed for a cluster is greater than that specified in the hint.

Assume that you specify a throttling rule that allows only one of the statements that meet the `SELECT * FROM t2 WHERE c1 = ?` condition to be executed on a single ODBServer. These statements are affected by the following factors:

The throttling feature is implemented on a per-ODBServer basis. If the OBProxy sends `SELECT * FROM t2 WHERE c1 = 1` to ODBServer 1 and ODBServer 2, two SQL statements are concurrently executed in the cluster.

ApsaraDB for OceanBase implements throttling by matching SQL statements to physical execution plans based on the signatures of the SQL statements. Therefore, the database separately controls the maximum number of concurrent SQL statements for each physical execution plan in the plan cache.

Signature

After you create an outline, a record is generated in the system table `__all_outline`. The signature column of the record indicates the unique identifier of the outline and is used in multiple scenarios.

A signature is generated based on the SQL statement and the corresponding `not_params`. In the SQL statement, the hint is excluded and constants are replaced with question marks (?).

 **Note** In the latest code, the parameter value of `not_param` is backfilled into the SQL statement. Therefore, the signature is generated based on only the SQL statement.

When a data manipulation language (DML) statement is executed, a signature is generated. The system checks whether an outline that has the same signature already exists.

When you create or add throttling rules or outlines, all throttling rules or outlines that are subordinate to the same outline name must have the same signature.

When you execute the `CREATE OUTLINE` statement, the system returns an `already exist` error if the outline name or the signature already exists.

When you create or replace an outline and use it to replace an existing outline, the operation succeeds only if the outline name and the signature are matched. If only one of the two attributes is matched, the system returns an `already exist` error.

When you modify an outline, the system checks whether the signature of the `stmt` that is specified in the `ALTER OUTLINE` statement matches the signature of the outline that is specified by `outline_name`.

18.5.16.5.2. Related system tables

Table name	Property	Description
__all_outline	System table	This table stores meta information about outlines.
__all_outline_history	System table	This table stores meta information about outlines.
__tenant_virtual_outline	Virtual table	This table shows the outline information for the current tenant. You can use this table to migrate outlines.
gv\$outline	View	This table is created based on the __tenant_virtual_outline table.
dba_outlines	View	This table is created based on the __all_outline table.
__tenant_virtual_concurrent_limit_sql	Virtual table	This table shows the throttling rules of the current tenant.
gv\$concurrent_limit_sql	View	This table is created based on the __tenant_virtual_concurrent_limit_sql table.

18.5.17. Hierarchical queries

You can use hierarchical queries to view hierarchical data that is organized based on hierarchy levels.

Hierarchical data is organized based on hierarchy levels. The hierarchy levels are frequent occurrences in real-life events. The following examples are used to explain hierarchy levels:

- Relationships between leaders and members in an organization
- Relationships between superior and subordinate departments in an enterprise
- Relationships between web pages that are connected by hyperlinks

ApsaraDB for OceanBase uses the `CONNECT BY` clause to perform hierarchical queries. ApsaraDB for OceanBase also provides relevant virtual columns and functions for hierarchical queries.

Syntax

```
SELECT select_list
FROM table_expression
[ WHERE ... ]
[ START WITH start_expression ]
CONNECT BY [NOCYCLE] { PRIOR child_expr = parent_expr | parent_expr = PRIOR child_expr }
[ ORDER SIBLINGS BY ...]
[ GROUP BY ... ]
[ HAVING ... ]
[ ORDER BY ... ]
```

Execution process

The key to using a hierarchical query is to understand the execution process. The following steps show the process to run a hierarchical query:

1. The ApsaraDB for OceanBase system performs the specified `SCAN` or `JOIN` operation that follows the `FROM` keyword.

2. The ApsaraDB for OceanBase system generates hierarchical relationships based on the `START WITH` and `CONNECT BY` clauses.
3. The ApsaraDB for OceanBase system executes the remaining clauses in the hierarchical query by following the same process of running general queries. The clauses include the `WHERE`, `GROUP`, and `ORDER BY` clauses.

The following steps show the detailed process to generate hierarchical relationships:

1. The ApsaraDB for OceanBase system queries the root rows based on the expression in the `START WITH` clause.
2. The ApsaraDB for OceanBase system generates the child rows of each root row based on the expression in the `CONNECT BY` clause.
3. The ApsaraDB for OceanBase system uses the generated child rows as new root rows to further generate child rows. The system repeats this step until no new row is generated.

18.5.18. Materialized views

Materialized views are a special type of views that store or materialize query results to accelerate specific queries. You can also use materialized views to implement read/write splitting.

Syntax

Create a materialized view

```
CREATE MATERIALIZED VIEW view_name AS subquery
```

Drop a materialized view

```
DROP MATERIALIZED VIEW [IF EXISTS] view_name
```

Limits

Only materialized views that join two tables can be created. The following limits apply to materialized views:

- Only user-created physical tables can be joined. Views, virtual tables, system tables, and other materialized views cannot be joined.
- A join condition must contain the primary key columns of the right table.
- You must specify `R{all_server}` for the right table. To simplify the procedure, specify the `ALL_SERVER` type for your cluster. This ensures that each server in the zone of your cluster has a read-only replica of the right table.
- Aliases cannot be specified for tables.
- The `PROJECTION`, `JOIN`, and `ORDER BY` operations are supported. You must include the `ORDER BY` clause in the statement.
- The `LIMIT` clause is not supported.
- `AGGREGATE` operations are not supported.
- Set operations such as `UNION`, `UNION ALL`, and `INTERSECT` are not supported.
- The `ROWNUM` clause is not supported.
- You can specify only original columns of the left table in the `ORDER BY` clause. You cannot specify expressions or generated columns in the clause.
- You can specify only original columns of the left table and the right table in the `PROJECTION` clause. You cannot specify expressions or generated columns in the clause.
- Each join condition must use the equal to (`=`) operator that is applied on an original column of the left table and an original column of the right table. You cannot specify expressions or generated columns in the join condition. You can use the `AND` operator to specify multiple join conditions. The `OR` operator is not supported. The specified original column in each join condition must uniquely identify each row of the right table. The primary key or unique key can uniquely identify each row. However, you can use only the primary key of the

right table when you specify the join condition.

- Nested subqueries are not supported.
- You cannot specify columns that have the same name in the PROJECTION clause.

The following limits apply to data definition language (DDL) operations on a materialized view:

- A materialized view takes effect only after a daily major freeze operation is performed.
- When you modify the attributes of a column in the left table or the right table, follow the existing modification rules if the column does not affect the materialized view.
- If the column affects the materialized view, modify the attributes in a compatible manner. For example, you can change the data type of a column from int to bigint, or from `varchar(10)` to `varchar(20)`.
- To delete a column that is used by a materialized view, you must drop the materialized view first.
- To truncate the left table or the right table that is used by the materialized view, you must drop the materialized view first.
- To drop a right table that is used by a materialized view, you must drop the materialized view first.
- When you drop a left table, the materialized view that uses the left table is also dropped.
- A materialized view cannot be truncated.
- After you drop a left table that is used by a materialized view, the table and the view cannot be recovered from the recycle bin.

Examples

```
create table collect_info (user_id int, item_id int, info_collect_time timestamp, primary key(user_id, item_id));

create table collect_item(item_id int primary key, item_url varchar(5000), item_price int) locality = 'FULL{1},READONLY{ALL_SERVER}@zone1';

create view info_item_view as select user_id, collect_info.item_id, info_collect_time, item_url from collect_info join collect_item on collect_info.item_id = collect_item.item_id order by user_id, info_collect_time; create materialized view info_item_mv as select user_id, collect_info.item_id, info_collect_time, item_url from collect_info join collect_item on collect_info.item_id = collect_item.item_id order by user_id, info_collect_time;
```

To query data from `info_item_view`, you can read the data that is stored in the materialized view `info_item_mv` and do not need to perform the JOIN operation.

The materialized view can also accelerate the following query:

```
select user_id, collect_info.item_id, info_collect_time, item_url from collect_info join collect_item on collect_info.item_id = collect_item.item_id where user_id = 1 order by user_id, info_collect_time;
```

This statement is rewritten to the following statement at the SQL layer:

```
select * from info_item_view where user_id = 1
```

You can execute the following EXPLAIN statement to identify whether a query is performed based on the materialized view:

```
mysql> explain select user_id, collect_info.item_id, info_collect_time, item_url from collect_info join collect_item on collect_info.item_id = collect_item.item_id where user_id = 1;
===== |ID|OPERATOR |NAME |EST. ROWS|COST| ----- |0 |TABLE SCAN|(info_item_mv)|1 |30 |===== Outputs & filters:
----- 0 - output([info_item_mv.user_id], [info_item_mv.item_id], [info_item_mv.info_collect_time], [info_item_mv.item_url]), filter([info_item_mv.item_id = info_item_mv.item_id]), access([info_item_mv.item_id], [info_item_mv.item_id], [info_item_mv.user_id], [info_item_mv.info_collect_time], [info_item_mv.item_url]), partitions(p0)
```

18.5.19. SQL modes

You can perform operations on an ApsaraDB for OceanBase server in different SQL modes for different clients. You can specify the SQL mode of the server for each application based on your business needs.

An SQL mode determines the SQL syntax that ApsaraDB for OceanBase supports and the data validation checks that ApsaraDB for OceanBase performs. This allows you to use ApsaraDB for OceanBase in different environments.

You can set the SQL mode by specifying the `--sql-mode="modes"` option. To enable the strict mode, specify either or both of the `STRICT_TRANS_TABLES` and `STRICT_ALL_TABLES` modes.

ApsaraDB for OceanBase supports the following SQL modes:

- **HIGH_NOT_PRECEDENCE**

By default, the NOT operator has a lower precedence than other operators. For example, the expression `NOT a BETWEEN b AND c` is parsed as `NOT (a BETWEEN b AND c)`. If you need the expression to be parsed as `(NOT a) BETWEEN b AND c`, enable the HIGH_NOT_PRECEDENCE SQL mode. This SQL mode improves the precedence of the NOT operator.

- **ONLY_FULL_GROUP_BY**

This SQL mode rejects queries for nonaggregated columns that are not specified in the `GROUP BY` clause.

- **PAD_CHAR_TO_FULL_LENGTH**

By default, spaces are truncated from values that are retrieved from CHAR columns. If the PAD_CHAR_TO_FULL_LENGTH mode is enabled, truncation does not occur. This mode does not apply if you retrieve values from VARCHAR columns.

Example:

```

mysql> CREATE TABLE t1 (c1 CHAR(10));
Query OK, 0 rows affected (0.09 sec)
mysql> INSERT INTO t1 (c1) VALUES('xy');
Query OK, 1 row affected (0.00 sec)
mysql> SET sql_mode = ""; //The default SQL mode
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT c1, CHAR_LENGTH(c1) FROM t1;
+-----+-----+
| c1 | CHAR_LENGTH(c1) |
+-----+-----+
| xy |          2 |
+-----+-----+
1 row in set (0.00 sec)
mysql> SET sql_mode = 'PAD_CHAR_TO_FULL_LENGTH'; //Change the SQL mode.
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT c1, CHAR_LENGTH(c1) FROM t1;
+-----+-----+
| c1 | CHAR_LENGTH(c1) |
+-----+-----+
| xy |          10 |
+-----+-----+
1 row in set (0.02 sec)

```

- **PIPES_AS_CONCAT**

This SQL mode treats `||` as a string concatenation operator (+) rather than as a synonym for the OR operator. It functions the same as `CONCAT()`.

- **STRICT_ALL_TABLES**

This SQL mode enables the strict SQL mode for all storage engines. Invalid values are rejected.

- **STRICT_TRANS_TABLES**

This SQL mode enables the strict SQL mode for transactional storage engines. In some cases, this SQL mode may also enable the strict mode for non-transactional storage engines.

The strict mode controls how ApsaraDB for OceanBase handles invalid or missing values in statements. A value can be invalid for several reasons. For example, a value is of an invalid data type for the column, or a value is out of a specified range. If a new row to be inserted does not contain a value for a column that has no explicit Default clause in its definition, the value is missing.

For transactional tables, if either `STRICT_ALL_TABLES` or `STRICT_TRANS_TABLES` is enabled, ApsaraDB for OceanBase returns an error if invalid or missing values are found in a statement. The statement is aborted and rolled back.

For non-transactional tables, if a bad value occurs in the first row to be inserted or updated, ApsaraDB for OceanBase handles in the same way under either of the two strict modes. The statement is aborted and the table remains unchanged. If the statement inserts or modifies multiple rows and a bad value occurs in the second or later row, the processing method of ApsaraDB for OceanBase depends on the strict mode that is enabled:

- If `STRICT_ALL_TABLES` is enabled, ApsaraDB for OceanBase returns an error and ignores the rest of the rows. However, a partial update is complete because the earlier rows are inserted or updated. This may not meet your requirements. To avoid this issue, we recommend that you use single-row statements. If you abort a single-row statement, your original table remains unchanged.

- If `STRICT_TRANS_TABLES` is enabled, ApsaraDB for OceanBase converts an invalid value to the closest valid value for the column and inserts the adjusted value. If a value is missing, ApsaraDB for OceanBase inserts the implicit default value in the column. In all cases, ApsaraDB for OceanBase generates a warning rather than an error and continues to execute the statement.

In strict mode, invalid dates such as `2004-04-31` are not allowed. Dates that have zero parts, such as `2004-04-00`, or zero dates are also rejected as invalid dates.

If you do not enable the `STRICT_TRANS_TABLES` or `STRICT_ALL_TABLES` mode, ApsaraDB for OceanBase adjusts the invalid or missing values, inserts the adjusted values, and then generates warnings.

- `NO_AUTO_VALUE_ON_ZERO`

`NO_AUTO_VALUE_ON_ZERO` affects how you handle `AUTO_INCREMENT` columns. In most cases, you can insert NULL or 0 into a column to generate the next sequence number for the column. However, 0 is disabled in `NO_AUTO_VALUE_ON_ZERO` mode. You can insert only NULL to generate the next sequence number.

18.5.20. System views

18.5.20.1. Overview

ApsaraDB for OceanBase provides a complete set of system views for you to check the running status of the system. The name of each system view starts with `v$` or `gv$`. `v$` indicates a view for a single OBServer and `gv$` indicates a view for all OBServers.

18.5.20.2. v\$statname

Shows definitions of all statistical events.

View definition

```
view_definition =
"select tenant_id as CON_ID,
stat_id as STAT_ID,
`statistic#` as `STATISTIC#`,
name as NAME,
display_name as DISPLAY_NAME,
class as CLASS
from oceanbase.__tenant_virtual_statname"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
STAT_ID	bigint(20)	The ID of the statistical event.
STATISTICS#	bigint(20)	The number that represents the statistical event.
NAME	varchar(64)	The name of the statistical event.
DISPLAY_NAME	varchar(64)	The alias of the statistical event.

Name	Data type	Description
CLASS	bigint(20)	The alias of the class of the statistical event.

18.5.20.3. v\$event_name

Shows definitions of all wait events.

View definition

```
view_definition =
"select tenant_id as CON_ID,
event_id as EVENT_ID,
`event#` as `EVENT#`,
name as NAME,
display_name as DISPLAY_NAME,
parameter1 as PARAMETER1,
parameter2 as PARAMETER2,
parameter3 as PARAMETER3,
wait_class_id as WAIT_CLASS_ID,
`wait_class#` as `WAIT_CLASS#`,
wait_class as WAIT_CLASS
from oceanbase.__tenant_virtual_event_name"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT#	bigint(20)	The number that represents the wait event.
NAME	varchar(64)	The name of the wait event.
DISPLAY_NAME	varchar(64)	The alias of the wait event.
PARAMETER1	varchar(64)	The first parameter of the wait event.
PARAMETER2	varchar(64)	The second parameter of the wait event.
PARAMETER3	varchar(64)	The third parameter of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CALSS#	bigint(20)	The number that represents the class of the wait event.

Name	Data type	Description
WAIT_CLASS	varchar(64)	The name of the class of the wait event.

18.5.20.4. v\$session_event

Shows the wait events that occur in each session.

View definition

```
view_definition =
"select SID,CON_ID,EVENT_ID,EVENT,WAIT_CLASS_ID,
`WAIT_CLASS#`,WAIT_CLASS,TOTAL_WAITS,TOTAL_TIMEOUTS,TIME_WAITED,
MAX_WAIT,AVERAGE_WAIT,TIME_WAITED_MICRO
from oceanbase.gv$session_event
where SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.
CON_ID	bigint(20)	The ID of the tenant.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT	varchar(64)	The description of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
TOTAL_WAITS	bigint(20)	The total number of waits for the event.
TOTAL_TIMEOUTS	bigint(20)	The total number of wait time-outs for the event.
TIME_WAITED	double	The total amount of time that was spent on waiting for the event. Unit: 10 ms.
MAX_WAIT	double	The maximum amount of time that was spent on waiting for the event. Unit: 10 ms.
AVERAGE_WAIT	double	The average amount of time that was spent on waiting for the event. Unit: 10 ms.

Name	Data type	Description
TIME_WAITED_MICRO	bigint(20)	The total amount of time that was spent on waiting for the event. Unit: microseconds.

18.5.20.5. v\$session_wait

Shows details about the current wait event for each session. For each wait event, ApsaraDB for OceanBase provides three parameters to record the corresponding values.

View definition

```
view_definition =
"select SID,CON_ID,EVENT,P1TEXT,P1,
P2TEXT,P2,P3TEXT,P3,WAIT_CLASS_ID,
`WAIT_CLASS#`,WAIT_CLASS,STATE,WAIT_TIME_MICRO,TIME_REMAINING_MICRO,
TIME_SINCE_LAST_WAIT_MICRO
from oceanbase.gv$session_wait
where SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.
CON_ID	bigint(20)	The ID of the tenant.
EVENT	varchar(64)	The description of the wait event.
P1TEXT	varchar(64)	The name of the first parameter of the wait event.
P1	bigint(20) unsigned	The value of the first parameter of the wait event.
P2TEXT	varchar(64)	The name of the second parameter of the wait event.
P2	bigint(20) unsigned	The value of the second parameter of the wait event.
P3TEXT	varchar(64)	The name of the third parameter of the wait event.
P3	bigint(20) unsigned	The value of the third parameter of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.

Name	Data type	Description
STATE	varchar(19)	The state of the wait event.
WAIT_TIME_MICRO	bigint(20)	The amount of time that was spent on waiting for the event. Unit: microseconds.
TIME_REMAINING_MICRO	bigint(20)	The amount of time that is remaining for the current wait before it times out. Unit: microseconds.
TIME_SINCE_LAST_WAIT_MICRO	bigint(20)	The amount of time that elapsed between the current wait event and the previous wait event. Unit: microseconds.

18.5.20.6. v\$session_wait_history

Shows details about the last 10 wait events for each session.

View definition

```
view_definition =
"SELECT SID,CON_ID,`SEQ#`,`EVENT#`,`EVENT`,
P1TEXT,P1,P2TEXT,P2,P3TEXT,
P3,WAIT_TIME_MICRO,TIME_SINCE_LAST_WAIT_MICRO,WAIT_TIME
FROM oceanbase.gv$session_wait_history
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

For more information about the fields, see the Fields section in [v\\$session_wait](#).

18.5.20.7. v\$sesstat

Shows summary information about statistical events based on sessions.

View definition

```
view_definition =
"select SID,CON_ID,`STATISTIC#`,`VALUE` from oceanbase.gv$sesstat"
where SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.
CON_ID	bigint(20)	The ID of the tenant.
STATISTICS#	bigint(20)	The number that represents the statistical event.

Name	Data type	Description
VALUE	bigint(20)	The total number of occurrences of the statistical event.

18.5.20.8. v\$sysstat

Shows summary information about statistical events based on tenants.

View definition

```
view_definition =
"SELECT CON_ID,`STATISTIC#`,VALUE,STAT_ID,NAME,CLASS from oceanbase.gv$sysstat
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
STATISTICS#	bigint(20)	The number that represents the statistical event.
VALUE	bigint(20)	The result value of the statistical event.
STAT_ID	bigint(20)	The ID of the statistical event.
NAME	varchar(64)	The name of the statistical event.
CLASS	bigint(20)	The class of the statistical event.

18.5.20.9. v\$system_event

Shows statistics about wait events based on tenants.

View definition

```
view_definition =
"SELECT CON_ID,EVENT_ID,EVENT,WAIT_CLASS_ID,`WAIT_CLASS#`,WAIT_CLASS,TOTAL_WAITS,TOTAL_TIMEOUTS,TIME_W
AITED,AVERAGE_WAIT, TIME_WAITED_MICRO
FROM oceanbase.gv$system_event
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT	varchar(64)	The description of the wait event.

Name	Data type	Description
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
TOTAL_WAITS	bigint(20)	The total number of waits for the event.
TOTAL_TIMEOUTS	bigint(20)	The total number of wait time-outs for the event.
TOTAL_WAITED	double	The total amount of time that was spent on waiting for the event. Unit: 10 ms.
AVERAGE_WAIT	double	The average amount of time that was spent on waiting for the event. Unit: 10 ms.
TOTAL_WAITED_MICRO	bigint(20)	The total amount of time that was spent on waiting for the event. Unit: microseconds.

18.5.20.10. v\$memory

Shows memory statistics based on tenants.

View definition

```
view_definition =
"SELECT TENANT_ID, CONTEXT, COUNT, USED, ALLOC_COUNT, FREE_COUNT
FROM oceanbase.gv$memory
WHERE IP=HOST_IP() AND PORT=RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
CONTEXT	varchar(256)	The name of the mod to which the memory belongs.
COUNT	bigint(20)	The number of memory units that are used in the mod, which is the difference between the number of allocated units and the number of available units.
USED	bigint(20)	The memory size that is used in the mod.
ALLOC_COUNT	bigint(20)	The total number of memory units that are allocated to the mod.

Name	Data type	Description
FREE_COUNT	bigint(20)	The total number of available memory units in the mod.

18.5.20.11. v\$memstore

Shows MemStore statistics based on tenants.

View definition

```
view_definition =
"SELECT TENANT_ID, ACTIVE, TOTAL, `FREEZE_TRIGGER`, `MEM_LIMIT`
FROM oceanbase.gv$memstore
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
ACTIVE	bigint(20)	The active MemStore size of the tenant.
TOTAL	bigint(20)	The total MemStore size of the tenant.
FREEZE_TRIGGER	bigint(20)	The MemStore size that triggers a major freeze action.
MEM_LIMIT	bigint(20)	The maximum MemStore size of the tenant.

18.5.20.12. v\$memstore_info

Shows MemStore statistics based on tenants. The statistics include the details about all partitions.

View definition

```
view_definition =
"SELECT TENANT_ID, IP, PORT, TABLE_ID, PARTITION_ID, VERSION, IS_ACTIVE, USED, HASH_ITEMS, BTREE_ITEMS
FROM oceanbase.gv$memstore_info
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.

Name	Data type	Description
TABLE_ID	bigint(20)	The ID of the table.
PARTITION_ID	bigint(20)	The ID of the partition.
VERSION	varchar(128)	The version number.
IS_ACTIVE	bigint(20)	Indicates whether the MemStore is active.
USED	bigint(20)	The memory size that is used by the MemStore.
HASH_ITEMS	bigint(20)	The hash indexes that are contained in the MemStore.
BTREE_ITEMS	bigint(20)	The B-tree indexes that are contained in the MemStore.

18.5.20.13. v\$plan_cache_stat

Shows the basic status of a plan cache.

View definition

```
view_definition=
"SELECT tenant_id,svr_ip,svr_port,sql_num,mem_used,access_count,hit_count,hit_rate,plan_num,mem_limit,hash_bucket,stmtkey_num
FROM oceanbase.gv$plan_cache_stat
WHERE svr_ip=HOST_IP() AND svr_port=RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
SQL_NUM	bigint(20)	The number of SQL statements for which execution plans are stored in the plan cache for future use.
MEM_USED	bigint(20)	The memory size that is used by the plan cache.
ACCESS_COUNT	bigint(20)	The number of accesses to the plan cache.
HIT_COUNT	bigint(20)	The number of plan cache hits.
HIT_RATE	bigint(20)	The plan cache hit ratio.
PLAN_NUM	bigint(20)	The number of execution plans in the plan cache.

Name	Data type	Description
MEM_LIMIT	bigint(20)	The maximum memory size that can be used by the plan cache.
HASH_BUCKET	bigint(20)	The number of buckets that are contained in the <code>hash map</code> of the plan cache.
STMTKEY_NUM	bigint(20)	The number of <code>stmt_keys</code> in the plan cache.

18.5.20.14. v\$plan_cache_plan_stat

Shows the status of all execution plans in a plan cache.

View definition

view_definition=

```
"SELECT tenant_id, svr_ip, svr_port, plan_id, sql_id, type, statement, plan_hash, first_load_time, schema_version, merged_version, last_active_time, avg_exe_usec, slowest_exe_time, slowest_exe_usec, slow_count, hit_count, executions, disk_reads, direct_writes, buffer_gets, application_wait_time, concurrency_wait_time, user_io_wait_time, rows_processed, elapsed_time, cpu_time, large_queries, delayed_large_queries, outline_version, outline_id
FROM oceanbase.gv$plan_cache_plan_stat
WHERE svr_ip=HOST_IP() AND svr_port=RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address of the server.
SVR_PORT	bigint(20)	The port number of the server.
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The ID of the SQL statement.
TYPE	bigint(20)	The type of the execution plan.
STATEMENT	varchar(4096)	The text of the SQL statement.
PLAN_HASH	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time when the execution plan was loaded for the first time.
SCHEMA_VERSION	bigint(20)	The version number of the schema.
MERGED_VERSION	bigint(20)	The merge version number that corresponds to the cached execution plan.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.

Name	Data type	Description
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	timestamp(6)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of hits.
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.
CPU_TIME	bigint(20) unsigned	The CPU time.

18.5.20.15. v\$sqlplan_cache_plan_explain

Shows basic information about all execution plans in a plan cache.

View definition

```
view_definition=
"SELECT *
FROM oceanbase.gv$sqlplan_cache_plan_explain
WHERE IP =HOST_IP() AND PORT = RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
OPERATOR	varchar(128)	The name of the operator.
NAME	varchar(256)	The name of the table.
ROWS	bigint(20)	The estimated number of result rows.
COST	bigint(20)	The estimated cost.
PROPERTY	varchar(256)	The information about the operator.

18.5.20.16. v\$sql_audit

Shows an SQL audit table that records the source and execution status of each SQL statement.

View definition

```
view_definition='SELECT * FROM oceanbase.gv$sql_audit WHERE svr_ip=HOST_IP() AND svr_port=RPC_PORT()'
```

Fields

Name	Data type	Description
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
REQUEST_ID	bigint(20)	The ID of the request.
SQL_EXEC_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
TRACE_ID	varchar(128)	The trace ID of the SQL statement.
SID	bigint(20) unsigned	The ID of the session.
CLIENT_IP	varchar(32)	The IP address of the client that sent the request.
CLIENT_PORT	bigint(20)	The port number of the client that sent the request.

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant that sent the request.
TENANT_NAME	varchar(64)	The name of the tenant that sent the request.
USER_ID	bigint(20)	The ID of the user that sent the request.
USER_NAME	varchar(64)	The name of the user that sent the request.
DB_ID	bigint(20) unsigned	The ID of the database where the SQL statement was executed.
DB_NAME	varchar(128)	The name of the database where the SQL statement was executed.
SQL_ID	varchar(32)	The ID of the SQL statement.
QUERY_SQL	varchar(65536)	The text of the SQL statement.
AFFECTED_ROWS	bigint(20)	The number of affected rows.
RETURN_ROWS	bigint(20)	The number of returned rows.
RET_CODE	bigint(20)	The return code of the execution result.
EVENT	varchar(64)	The name of the event for which the amount of waited time is the longest.
P1TEXT	varchar(64)	The name of the first parameter of the wait event.
P1	bigint(20) unsigned	The value of the first parameter of the wait event.
P2TEXT	varchar(64)	The name of the second parameter of the wait event.
P2	bigint(20) unsigned	The value of the second parameter of the wait event.
P3TEXT	varchar(64)	The name of the third parameter of the wait event.
P3	bigint(20) unsigned	The value of the third parameter of the wait event.
LEVEL	bigint(20)	The level of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.

Name	Data type	Description
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
STATE	varchar(19)	The state of the wait event.
WAIT_TIME_MICRO	bigint(20)	The amount of time that was spent on waiting for the event. Unit: microseconds.
TOTAL_WAIT_TIME_MICRO	bigint(20)	The total amount of time for all wait events during the execution. Unit: microseconds.
TOTAL_WAITS	bigint(20)	The total number of waits during the execution.
RPC_COUNT	bigint(20)	The number of RPC requests that were sent.
PLAN_TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
IS_INNER_SQL	tinyint(4)	Indicates whether the request is an internal SQL query.
IS_EXECUTOR_RPC	tinyint(4)	Indicates whether the request is an RPC request.
IS_HIT_PLAN	tinyint(4)	Indicates whether the plan cache is hit.
REQUEST_TIME	bigint(20)	The time when the execution started.
ELAPSED_TIME	bigint(20)	The amount of time that was spent on processing the request after the request was received.
NET_TIME	bigint(20)	The amount of time that elapsed from the time when the RPC request was sent to the time when the RPC request was received.
NET_WAIT_TIME	bigint(20)	The amount of time that elapsed from the time when the request was received to the time when the request was added to the queue.
QUEUE_TIME	bigint(20)	The amount of time for which the request waited in the queue.
DECODE_TIME	bigint(20)	The amount of time that was spent on decoding the request after the request exited the queue.
GET_PLAN_TIME	bigint(20)	The amount of time that elapsed from the time when the SQL statement was processed to the time when an execution plan was generated.

Name	Data type	Description
EXECUTE_TIME	bigint(20)	The amount of time that was spent on executing the execution plan.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
SCHEDULE_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all schedule events.
ROW_CACHE_HIT	bigint(20)	The number of row cache hits.
BLOOM_FILTER_CACHE_HIT	bigint(20)	The number of Bloom filter cache hits.
BLOCK_CACHE_HIT	bigint(20)	The number of block cache hits.
BLOCK_INDEX_CACHE_HIT	bigint(20)	The number of block index cache hits.
DISK_READS	bigint(20)	The number of physical reads from disks.
RETRY_CNT	bigint(20)	The number of times the SQL statement was re-executed from the redo log.
TABLE_SCAN	tinyint(4)	Indicates a table scan.
CONSISTENCY_LEVEL	bigint(20)	The consistency level of the SQL statement.
MEMSTORE_READ_ROW_COUNT	bigint(20)	The number of rows in the SQL result set that was returned from the MemStore.
SSSTORE_READ_ROW_COUNT	bigint(20)	The number of rows in the SQL result set that was returned from the disk.
REQUEST_MEMOMERY_USED	bigint(20)	The memory size used by the SQL statement.

18.5.20.17. v\$latch

Shows latch information.

View definition

```
view_definition =
"SELECT *
FROM oceanbase.gv$latch
WHERE SVR_IP=host_ip() and SVR_PORT=rpc_port()"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
ADDR	varchar(256)	The value is displayed as NULL.
LATCH#	bigint(20)	The number that represents the class of the latch.
LEVEL#	bigint(20)	The number that represents the level of the latch.
NAME	varchar(256)	The name of the latch.
HASH	bigint(20)	The value is displayed as 0.
GETS	bigint(20)	The number of successful low_lock operations.
MISSES	bigint(20)	The number of waits after the low_lock operations are performed more than the yield operations.
SLEEPS	bigint(20)	The total number of yield operations.
IMMEDIATE_GETS	bigint(20)	The number of successful try_lock operations.
IMMEDIATE_MISSES	bigint(20)	The number of failed try_lock operations.
SPIN_GETS	bigint(20)	The total number of spin operations.
WAIT_TIME	bigint(20)	The amount of time that was spent on waiting for the latch.

18.5.20.18. v\$oobrpc_outgoing

Shows statistics about RPC requests that were sent from an OBServer for different tenants based on the RPC packet codes.

View definition

```
view_definition =
"SELECT *
FROM gv$obrpc_outgoing
WHERE IP=HOST_IP() AND PORT=RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PCODE	bigint(20)	rpc packet code
PCODE_NAME	varchar(256)	The name that corresponds to the RPC packet code.
COUNT	bigint(20)	The number of times that the RPC packet code was used.
TOTAL_TIME	bigint(20)	The total amount of time that was spent.
TOTAL_SIZE	bigint(20)	The total amount of data that was sent.
FAILURE	bigint(20)	The number of RPC requests that have failed sending.
TIMEOUT	bigint(20)	The number of sending time-outs.
SYNC	bigint(20)	The number of RPC requests that were sent synchronously.
ASYNC	bigint(20)	The number of RPC requests that were sent asynchronously.
LAST_TIMESTAMP	timestamp(6)	The time when the statistics was last updated.

18.5.20.19. v\$obrpc_incoming

Shows statistics about RPC requests that were received by an OBServer for different tenants based on the RPC packet codes.

View definition

```
view_definition =
"SELECT *
FROM gv$obrpc_incoming
WHERE IP=HOST_IP() AND PORT=RPC_PORT()"
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PCODE	bigint(20)	rpc packet code
PCODE_NAME	varchar(256)	The name that corresponds to the RPC packet code.
COUNT	bigint(20)	The number of times that the RPC packet code was used.
TOTAL_SIZE	bigint(20)	The total amount of data that was received.
NET_TIME	bigint(20)	The network time.
WAIT_TIME	bigint(20)	The amount of time that elapsed from the time when the request was received to the time when the request was added to the queue.
QUEUE_TIME	bigint(20)	The amount of time for which the request waited in the queue.
PROCESS_TIME	bigint(20)	The amount of time that was spent on processing the request.
LAST_TIMESTAMP	timestamp(6)	The last update time.

18.5.20.20. v\$sql

Shows hot-updated SQL statistics for all execution plans. Each row in the table records the statistics of a single execution plan. The statistics contain summary information about multiple executions of the plan.

View definition

```
view_definition=
"SELECT *
FROM oceanbase.gv$sql
WHERE svr_ip=HOST_IP() AND svr_port=RPC_PORT()"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The identifier of the SQL statement.

Name	Data type	Description
TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
SQL_TEXT	varchar(4096)	The text of the SQL statement.
PLAN_HASH_VALUE	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time of the first execution.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	timestamp(6)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of plan cache hits.
PLAN_SIZE	bigint(20)	
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.

Name	Data type	Description
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.
CPU_TIME	bigint(20) unsigned	The CPU time.

18.5.20.21. v\$sql_monitor

Shows statistics about slow SQL queries based on execution plans. Each slow SQL query has a statistical record that can be used to trace the execution plan.

View definition

```
view_definition=
"SELECT tenant_id as CON_ID,\
  request_id as SQL_EXEC_ID,\
  job_id as JOB_ID,\
  task_id as TASK_ID,\
  svr_ip as SVR_IP,\
  svr_port as SVR_PORT,\
  sql_exec_start as SQL_EXEC_START, \
  plan_id as PLAN_ID,\
  scheduler_ip as SCHEDULER_IP, \
  scheduler_port as SCHEDULER_PORT, \
  monitor_info as MONITOR_INFO,\
  extend_info as EXTEND_INFO FROM oceanbase.__all_virtual_sql_monitor \
WHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) \
and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1) "
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SQL_EXEC_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
JOB_ID	bigint(20)	The job ID. The executor uses this ID to identify a segment in the physical execution plan. Job IDs are globally incremental on a single OBServer.
TASK_ID	bigint(20)	The task ID, which uniquely identifies an execution of a segment that corresponds to a job ID in a distributed execution plan.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.

Name	Data type	Description
SQL_EXEC_START	timestamp(6)	The time when the execution started.
PLAN_ID	bigint(20)	The plan ID, which uniquely identifies an execution plan in the plan cache on a single OBServer. Plan IDs are incremental values that are managed by the plan cache module. When a new execution plan is added to the plan cache, a plan ID is assigned to the new execution plan.
SCHEDULER_IP	varchar(32)	The IP address of the OBServer that is scheduled to execute the SQL statement.
SCHEDULER_PORT	bigint(20)	The port number of the OBServer that is scheduled to execute the SQL statement.
MONITOR_INFO	varchar(65535)	The relevant information such as the event for which the amount of waited time is the longest and the reception time.
EXTEND_INFO	varchar(65535)	The extended information such as all trace information that was generated in the process of executing the SQL statement.

18.5.20.22. v\$sql_plan_monitor

Shows statistics about slow SQL queries based on execution plans. Each slow SQL query has a statistical record that can be used to trace the execution plan.

View definition

```
view_definition='SELECT * from oceanbase.gv$sql_plan_monitor WHERE svr_ip=HOST_IP() AND svr_port=RPC_Port()'
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SQL_EXEC_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
JOB_ID	bigint(20)	The job ID. The executor uses this ID to identify a segment in the physical execution plan. Job IDs are globally incremental on a single OBServer.

Name	Data type	Description
TASK_ID	bigint(20)	The task ID, which uniquely identifies an execution of a segment that corresponds to a job ID in a distributed execution plan.
OPERATION_ID	bigint(20)	The operator ID, which uniquely identifies an operator in a physical execution plan. Operator IDs are generated based on the postorder traversal of the physical execution plan tree.
SVR_IP	varchar(32)	The IP address of the OBServer.
SVR_PORT	bigint(20)	The port number of the OBServer.
SQL_EXEC_START	timestamp(6)	The time when the execution of the SQL statement started.
PLAN_ID	bigint(20)	The plan ID, which uniquely identifies an execution plan in the plan cache on a single OBServer. Plan IDs are incremental values that are managed by the plan cache module. When a new execution plan is added to the plan cache, a plan ID is assigned to the new execution plan.
SCHEDULER_IP	varchar(32)	The IP address of the OBServer that is scheduled to execute the SQL statement.
SCHEDULER_PORT	bigint(20)	The port number of the OBServer that is scheduled to execute the SQL statement.
PLAN_OPERATION	varchar(32)	The operators that are used in the execution plan.
MONITOR_INFO	varchar(65535)	The execution time statistics for each step in the execution plan.
EXTEND_INFO	varchar(65535)	The extended information.

18.5.20.23. gv\$plan_cache_stat

Shows the plan cache status in a cluster. A cluster consists of multiple OBServers where multiple OBSERVER instances are deployed.

View definition

```

view_definition='
"SELECT tenant_id,
    svr_ip,
    svr_port,
    sql_num,
    mem_used,
    access_count,
    hit_count,
    hit_rate,
    plan_num,
    mem_limit,
    hash_bucket,
    stmtkey_num
FROM oceanbase.__all_virtual_plan_cache_stat
WHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1) "

```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
SQL_NUM	bigint(20)	The number of SQL statements for which the execution plans are stored in all plan caches in the cluster.
MEM_USED	bigint(20)	The memory size used by all plan caches in the cluster.
ACCESS_COUNT	bigint(20)	The number of accesses to all plan caches in the cluster.
HIT_COUNT	bigint(20)	The number of plan cache hits of the cluster.
HIT_RATE	bigint(20)	The plan cache hit ratio of the cluster.
PLAN_NUM	bigint(20)	The number of execution plans that are stored in the plan caches in the cluster.
MEM_LIMIT	bigint(20)	The maximum memory size that can be used by the plan caches in the cluster.
HASH_BUCKET	bigint(20)	The number of buckets that are contained in the hash maps of all plan caches in the cluster.
STMTKEY_NUM	bigint(20)	The number of stmt_keys in all plan caches in the cluster.

18.5.20.24. gv\$sqlplan_cache_plan_stat

Shows the status of each execution plan in the plan caches of each OBCServer instance on all OBCServers in a cluster.

View definition

```
view_definition=
"SELECT tenant_id,
  svr_ip,
  svr_port,
  plan_id,
  sql_id,
  type,
  statement,
  plan_hash,
  first_load_time,
  schema_version,
  merged_version,
  last_active_time,
  avg_exe_usec,
  slowest_exe_time,
  slowest_exe_usec,
  slow_count,
  hit_count,
  executions,
  disk_reads,
  direct_writes,
  buffer_gets,
  application_wait_time,
  concurrency_wait_time,
  user_io_wait_time,
  rows_processed,
  elapsed_time,
  cpu_time,
  large_querys,
  delayed_large_querys,
  outline_version,
  outline_id
FROM
  oceanbase.__all_virtual_plan_stat
WHERE
  is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1)'
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.

Name	Data type	Description
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The identifier of the SQL statement.
TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
STATEMENT	varchar(4096)	The text of the SQL statement.
PLAN_HASH	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time when the execution plan was loaded for the first time.
SCHEMA_VERSION	bigint(20)	The version number of the schema.
MERGED_VERSION	bigint(20)	The merge version that corresponds to the execution plan in the plan cache.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	bigint(20)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of plan cache hits.
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_READS	bigint(20)	The number of direct reads.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.

Name	Data type	Description
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.
CPU_TIME	bigint(20) unsigned	The CPU time.
LARGE_QUERYYS	bigint(20)	
DELAYED_LARGE_QUERYYS	bigint(20)	
OUTLINE_ID	bigint(20)	

18.5.20.25. gv\$session_event

Shows information about wait events based on sessions for all OBServers in a cluster.

View definition

```
view_definition = "select session_id as SID, \
    tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    event_id as EVENT_ID, \
    event as EVENT, \
    wait_class_id as WAIT_CLASS_ID, \
    `wait_class#` as `WAIT_CLASS#`, \
    wait_class as WAIT_CLASS, \
    total_waits as TOTAL_WAITS, \
    total_timeouts as TOTAL_TIMEOUTS, \
    time_waited as TIME_WAITED, \
    max_wait as MAX_WAIT, \
    average_wait as AVERAGE_WAIT, \
    time_waited_micro as TIME_WAITED_MICRO \
from oceanbase.__all_virtual_session_event where \
is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT	varchar(64)	The description of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
TOTAL_WAITS	bigint(20)	The total number of waits for the event.
TOTAL_TIMEOUTS	bigint(20)	The total number of wait time-outs for the event.
TIME_WAITED	double	The total amount of time that was spent on waiting for the event. Unit: 10 ms.
MAX_WAIT	double	The maximum amount of time that was spent on waiting for the event. Unit: 10 ms.
AVERAGE_WAIT	double	The average amount of time that was spent on waiting for the event. Unit: 10 ms.
TIME_WAITED_MICRO	bigint(20)	The total amount of time that was spent on waiting for the event. Unit: microseconds.

18.5.20.26. gv\$sqlsession_wait

Shows details about all wait events based on sessions for all OBServers in a cluster.

View definition

```
view_definition = "select session_id as SID, \
    tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    event_id as EVENT_ID, \
    event as EVENT, \
    wait_class_id as WAIT_CLASS_ID, \
    `wait_class#` as `WAIT_CLASS#`, \
    wait_class as WAIT_CLASS, \
    total_waits as TOTAL_WAITS, \
    total_timeouts as TOTAL_TIMEOUTS, \
    time_waited as TIME_WAITED, \
    max_wait as MAX_WAIT, \
    average_wait as AVERAGE_WAIT, \
    time_waited_micro as TIME_WAITED_MICRO \
from oceanbase.__all_virtual_session_event where \
is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT	varchar(64)	The description of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
TOTAL_WAITS	bigint(20)	The total number of waits for the event.
TOTAL_TIMEOUTS	bigint(20)	The total number of wait time-outs for the event.
TIME_WAITED	double	The total amount of time that was spent on waiting for the event. Unit: 10 ms.
MAX_WAIT	double	The maximum amount of time that was spent on waiting for the event. Unit: 10 ms.
AVERAGE_WAIT	double	The average amount of time that was spent on waiting for the event.

Name	Data type	Description
TIME_WAITED_MICRO	bigint(20)	The total amount of time that was spent on waiting for the event. Unit: microseconds.

18.5.20.27. gv\$session_wait_history

Shows details about all wait events based on sessions for all OBServers in a cluster. This view shows more detailed history information than the gv\$session_wait view.

View definition

```
view_definition = "select session_id as SID, \
    tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    `seq#` as `SEQ#`, \
    `event#` as `EVENT#`, \
    event as EVENT, \
    p1text as P1TEXT, \
    p1 as P1, \
    p2text as P2TEXT, \
    p2 as P2, \
    p3text as P3TEXT, \
    p3 as P3, \
    wait_time_micro as WAIT_TIME_MICRO, \
    time_since_last_wait_micro as TIME_SINCE_LAST_WAIT_MICRO, \
    wait_time as WAIT_TIME \
from oceanbase.__all_virtual_session_wait_history where \
is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
SEQ#	bigint(20)	
EVENT#	bigint(20)	The number that represents the wait event.
EVENT	varchar(64)	The description of the wait event.
P1TEXT	varchar(64)	The name of the first parameter of the wait event.
P1	bigint(20) unsigned	The value of the first parameter of the wait event.

Name	Data type	Description
P2TEXT	varchar(64)	The name of the second parameter of the wait event.
P2	bigint(20) unsigned	The value of the second parameter of the wait event.
P3TEXT	varchar(64)	The name of the third parameter of the wait event.
P3	bigint(20) unsigned	The value of the third parameter of the wait event.
WAIT_TIME_MICRO	bigint(20)	The amount of time that was spent on waiting for the event. Unit: microseconds.
TIME_SINCE_LAST_WAIT_MICRO	bigint(20)	The amount of time that elapsed between the current wait event and the previous wait event. Unit: microseconds.
WAIT_TIME	double	

18.5.20.28. gv\$system_event

Shows information about wait events based on tenants for all OBServers in a cluster.

View definition

```
view_definition = "select tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    event_id as EVENT_ID, \
    event as EVENT, \
    wait_class_id as WAIT_CLASS_ID, \
    `wait_class#` as `WAIT_CLASS#`, \
    wait_class as WAIT_CLASS, \
    total_waits as TOTAL_WAITS, \
    total_timeouts as TOTAL_TIMEOUTS, \
    time_waited as TIME_WAITED, \
    average_wait as AVERAGE_WAIT, \
    time_waited_micro as TIME_WAITED_MICRO \
from oceanbase.__all_virtual_system_event where \
is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.

Name	Data type	Description
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
EVENT_ID	bigint(20)	The ID of the wait event.
EVENT	varchar(64)	The description of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
TOTAL_WAITS	bigint(20)	The total number of waits for the event.
TOTAL_TIMEOUTS	bigint(20)	The total number of wait time-outs for the event.
TIME_WAITED	double	The total amount of time that was spent on waiting for the event. Unit: 10 ms.
MAX_WAIT	double	The maximum amount of time that was spent on waiting for the event. Unit: 10 ms.
AVERAGE_WAIT	double	The average amount of time that was spent on waiting for the event. Unit: 10 ms.
TIME_WAITED_MICRO	bigint(20)	The total amount of time that was spent on waiting for the event. Unit: microseconds.

18.5.20.29. gv\$sesstat

Shows information about statistical events based on sessions for all OBServers in a cluster.

View definition

```
view_definition = "select session_id as SID, \
    tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    `statistic#` as `STATISTIC#`, \
    value as VALUE \
from oceanbase.__all_virtual_sesstat \
where is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and can_visible = true and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
SID	bigint(20)	The ID of the session.
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
STATISTIC#	bigint(20)	The number that represents the statistical event.
VALUE	bigint(20)	

18.5.20.30. gv\$sysstat

Shows information about statistical events based on tenants for all OBServers in a cluster.

View definition

```
view_definition = "select tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    `statistic#` as `STATISTIC#`, \
    value as VALUE, \
    stat_id as STAT_ID, \
    name as NAME, \
    class as CLASS \
from oceanbase.__all_virtual_sysstat \
where is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and can_visible = true and \
(tenant_id = effective_tenant_id() or effective_tenant_id() = 1)"
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
STATISTIC#	bigint(20)	The number that represents the statistical event.
VALUE	bigint(20)	
STAT_ID	bigint(20)	The ID of the statistical event.
NAME	varchar(64)	The name of the statistical event.
CLASS	bigint(20)	The alias of the class of the statistical event.

18.5.20.31. gv\$sql_audit

Shows an SQL audit table for all OBServers in a cluster.

View definition

```
view_definition = 'select \  
    svr_ip as SVR_IP, \  
    svr_port as SVR_PORT, \  
    request_id as REQUEST_ID, \  
    execution_id as SQL_EXEC_ID, \  
    trace_id as TRACE_ID, \  
    client_ip as CLIENT_IP, \  
    client_port as CLIENT_PORT, \  
    tenant_id as TENANT_ID, \  
    tenant_name as TENANT_NAME, \  
    user_id as USER_ID, \  
    user_name as USER_NAME, \  
    sql_id as SQL_ID, \  
    query_sql as QUERY_SQL, \  
    affected_rows as AFFECTED_ROWS, \  
    return_rows as RETURN_ROWS, \  
    ret_code as RET_CODE, \  
    event as EVENT, \  
    p1text as P1TEXT, \  
    p1 as P1, \  
    p2text as P2TEXT, \  
    p2 as P2, \  
    p3text as P3TEXT, \  
    p3 as P3, \  
    level as LEVEL, \  
    wait_class_id as WAIT_CLASS_ID, \  
    `wait_class#` as `WAIT_CLASS#`, \  
    wait_class as WAIT_CLASS, \  
    state as STATE, \  
    wait_time_micro as WAIT_TIME_MICRO, \  
    total_wait_time_micro as TOTAL_WAIT_TIME_MICRO, \  
    total_waits as TOTAL_WAITS, \  
    rpc_count as RPC_COUNT, \  
    plan_type as PLAN_TYPE, \  
    is_inner_sql as IS_INNER_SQL, \  
    is_executor_rpc as IS_EXECUTOR_RPC, \  
    is_hit_plan as IS_HIT_PLAN, \  
    request_time as REQUEST_TIME, \  
    elapsed_time as ELAPSED_TIME, \  
    net_time as NET_TIME, \  
    net_wait_time as NET_WAIT_TIME, \  
    queue_time as QUEUE_TIME, \  
    ...'
```

```

queue_time as QUEUE_TIME, \
decode_time as DECODE_TIME, \
get_plan_time as GET_PLAN_TIME, \
execute_time as EXECUTE_TIME, \
application_wait_time as APPLICATION_WAIT_TIME, \
concurrency_wait_time as CONCURRENCY_WAIT_TIME, \
user_io_wait_time as USER_IO_WAIT_TIME, \
schedule_time as SCHEDULE_TIME, \
disk_reads as DISK_READS \
from oceanbase.__all_virtual_sql_audit \
where is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and (tenant_id = effective_tenant_id() or
effective_tenant_id() = 1)'

```

Fields

Name	Data type	Description
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
REQUEST_ID	bigint(20)	The ID of the request.
EXECUTION_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
TRACE_ID	varchar(128)	The trace ID of the SQL statement.
CLIENT_IP	varchar(32)	The IP address of the client that sent the request.
CLIENT_PORT	bigint(20)	The port number of the client that sent the request.
TENANT_ID	bigint(20)	The ID of the tenant that sent the request.
TENANT_NAME	varchar(64)	The name of the tenant that sent the request.
USER_ID	bigint(20)	The ID of the user that sent the request.
USER_NAME	varchar(64)	The name of the user that sent the request.
SQL_ID	varchar(32)	The identifier of the SQL statement.
QUERY_SQL	varchar(65536)	The text of the SQL statement.
PLAN_ID	bigint(20)	The ID of the execution plan.
AFFECTED_ROWS	bigint(20)	The number of affected rows.
RETURN_ROWS	bigint(20)	The number of returned rows.
RET_CODE	bigint(20)	The return code of the execution result.

Name	Data type	Description
EVENT	varchar(64)	The description of the wait event.
P1TEXT	varchar(64)	The name of the first parameter of the wait event.
P1	bigint(20) unsigned	The value of the first parameter of the wait event.
P2TEXT	varchar(64)	The name of the second parameter of the wait event.
P2	bigint(20) unsigned	The value of the second parameter of the wait event.
P3TEXT	varchar(64)	The name of the third parameter of the wait event.
P3	bigint(20) unsigned	The value of the third parameter of the wait event.
LEVEL	bigint(20)	The level of the wait event.
WAIT_CLASS_ID	bigint(20)	The ID of the class of the wait event.
WAIT_CLASS#	bigint(20)	The number that represents the class of the wait event.
WAIT_CLASS	varchar(64)	The name of the class of the wait event.
STATE	varchar(19)	The state of the wait event.
WAIT_TIME_MICRO	bigint(20)	The amount of time that was spent on waiting for the event. Unit: microseconds.
TOTAL_WAIT_TIME_MICRO	bigint(20)	The total amount of time for all wait events during the execution. Unit: microseconds.
TOTAL_WAITS	bigint(20)	The total number of waits during the execution.
RPC_COUNT	bigint(20)	The number of RPC requests that were sent.
PLAN_TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
IS_INNER_SQL	tinyint(4)	Indicates whether the request is an internal SQL query.
IS_EXECUTOR_RPC	tinyint(4)	Indicates whether the request is an RPC request.
IS_HIT_PLAN	tinyint(4)	Indicates whether the plan cache is hit.
REQUEST_TIME	bigint(20)	The time when the execution started.
ELAPSED_TIME	bigint(20)	The amount of time that was spent on processing the request after the request was received.

Name	Data type	Description
NET_TIME	bigint(20)	The amount of time that elapsed from the time when the RPC request was sent to the time when the RPC request was received.
NET_WAIT_TIME	bigint(20)	The amount of time that elapsed from the time when the request was received to the time when the request was added to the queue.
QUEUE_TIME	bigint(20)	The amount of time for which the request waited in the queue.
DECODE_TIME	bigint(20)	The amount of time that was spent on decoding the request after the request exited the queue.
GET_PLAN_TIME	bigint(20)	The amount of time that elapsed from the time when the SQL statement was processed to the time when an execution plan was generated.
EXECUTE_TIME	bigint(20)	The amount of time that was spent on executing the execution plan.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
SCHEDULE_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all schedule events.
DISK_READS	bigint(20)	The number of physical reads from disks.

18.5.20.32. gv\$latch

Shows latch information for all OBServers in a cluster.

View definition

```

view_definition = 'select tenant_id as CON_ID, \
    svr_ip as SVR_IP, \
    svr_port as SVR_PORT, \
    addr as ADDR, \
    latch_id as `LATCH#`, \
    level as `LEVEL#`, \
    name as NAME, \
    hash as HASH, \
    gets as GETS, \
    misses as MISSES, \
    sleeps as SLEEPS, \
    immediate_gets as IMMEDIATE_GETS, \
    immediate_misses as IMMEDIATE_MISSES, \
    spin_gets as SPIN_GETS, \
    wait_time as WAIT_TIME from oceanbase.__all_virtual_latch where \
    is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and \
    (tenant_id = effective_tenant_id() or effective_tenant_id() = 1)'

```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
ADDR	varchar(256)	The value is displayed as NULL.
LATCH_ID#	bigint(20)	The number that represents the class of the latch.
LEVEL#	bigint(20)	The number that represents the level of the latch.
NAME	varchar(256)	The name of the latch.
HASH	bigint(20)	The value is displayed as 0.
GETS	bigint(20)	The number of successful low_lock operations.
MISSES	bigint(20)	The number of waits after the low_lock operations are performed more than the yield operations.
SLEEPS	bigint(20)	The total number of yield operations.
IMMEDIATE_GETS	bigint(20)	The number of successful try_lock operations.
IMMEDIATE_MISSES	bigint(20)	The number of failed try_lock operations.

Name	Data type	Description
SPIN_GETS	bigint(20)	The total number of spin operations.
WAIT_TIME	bigint(20)	The amount of time that was spent on waiting for the latch.

18.5.20.33. gv\$memory

Shows memory statistics based on tenants for all OBServers in a cluster.

View definition

```
view_definition = ' SELECT \
  tenant_id as TENANT_ID, \
  svr_ip AS IP, \
  svr_port AS PORT, \
  mod_id as MOD_ID, \
  mod_type as MOD_TYPE, \
  mod_name AS CONTEXT, \
  count as COUNT, \
  zone as ZONE, \
  used as USED, \
  alloc_count as ALLOC_COUNT, \
  free_count as FREE_COUNT \
FROM \
  oceanbase.__all_virtual_memory_info \
WHERE \
  ( \
    effective_tenant_id()=1 \
    OR \
    tenant_id=effective_tenant_id() \
  ) \
AND \
  mod_type='user' \
AND \
  is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) \'
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
MOD_ID	bigint(20)	The ID of the mod.
MOD_TYPE	varchar(256)	The type of the mod.

Name	Data type	Description
CONTEXT	varchar(256)	The name of the mod to which the memory belongs.
COUNT	bigint(20)	The number of memory units that are used in the mod, which is the difference between the number of allocated units and the number of available units.
ZONE	varchar(256)	
USED	bigint(20)	The memory size that is used in the mod.
ALLOC_COUNT	bigint(20)	The total number of memory units that are allocated to the mod.
FREE_COUNT	bigint(20)	The total number of available memory units in the mod.

18.5.20.34. gv\$memstore

Shows MemStore statistics based on tenants for all OBServers in a cluster.

View definition

```
view_definition = ' SELECT
    TENANT_ID,
    SVR_IP AS IP,
    SVR_PORT AS PORT,
    ACTIVE_MEMSTORE_USED AS ACTIVE,
    TOTAL_MEMSTORE_USED AS TOTAL,
    MAJOR_FREEZE_TRIGGER AS `FREEZE_TRIGGER`,
    MEMSTORE_LIMIT AS `MEM_LIMIT`
FROM
    oceanbase.__all_virtual_tenant_memstore_info
WHERE
    (EFFECTIVE_TENANT_ID)=1
    OR
    TENANT_ID=EFFECTIVE_TENANT_ID()
    AND
    is_serving_tenant(svr_ip, svr_port, effective_tenant_id())
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.

Name	Data type	Description
ACTIVE	bigint(20)	The active MemStore size of the cluster.
TOTAL	bigint(20)	The total MemStore size of the cluster.
FREEZE_TRIGGER	bigint(20)	The MemStore size that triggers a major freeze action.
MEM_LIMIT	bigint(20)	The maximum memory size that can be used in the cluster.

18.5.20.35. gv\$memstore_info

Shows MemStore statistics based on tenants for all OBServers in a cluster. The statistics include the details about all partitions.

View definition

```
view_definition=
SELECT
    TENANT_ID,
    SVR_IP AS IP,
    SVR_PORT AS PORT,
    table_id AS TABLE_ID,
    partition_idx AS PARTITION_ID,
    VERSION,
    IS_ACTIVE,
    MEM_USED as USED,
    hash_item_count as HASH_ITEMS,
    btree_item_count as BTREE_ITEMS
FROM
    oceanbase.__all_virtual_memstore_info
WHERE
    (EFFECTIVE_TENANT_ID)=1
    OR
    TENANT_ID=EFFECTIVE_TENANT_ID()
AND
    is_serving_tenant(svr_ip, svr_port, effective_tenant_id())
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
TABLE_ID	bigint(20)	The ID of the table.

Name	Data type	Description
PARTITION_ID	bigint(20)	The ID of the partition.
PARTITION_CNT	bigint(20)	The number of partitions.
VERSION	varchar(128)	The version number.
IS_ACTIVE	bigint(20)	Indicates whether the MemStore is active.
USED	bigint(20)	The memory size that is used by the MemStore.
HASH_ITEMS	bigint(20)	The hash indexes that are contained in the MemStore.
BTREE_ITEMS	bigint(20)	The B-tree indexes that are contained in the MemStore.

18.5.20.36. gv\$sqlplan_cache_plan_explain

View definition

```
view_definition='SELECT TENANT_ID,\n      SVR_IP as IP, \n      SVR_PORT as PORT, \n      PLAN_ID, \n      OPERATOR, \n      NAME,\n      ROWS,\n      COST,\n      PROPERTY \nFROM oceanbase.__all_virtual_plan_cache_plan_explain \nWHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and (EFFECTIVE_TENANT_ID)=1 or TENANT_ID=EFFECTIVE_TENANT_ID()'
```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
OPERATOR	varchar(128)	The name of the operator.
NAME	varchar(256)	The name of the table.
ROWS	bigint(20)	The estimated number of result rows.

Name	Data type	Description
COST	bigint(20)	The estimated cost.
PROPERTY	varchar(256)	-

18.5.20.37. gv\$obrpc_outgoing

Shows statistics about RPC requests that were sent from all OBServers in a cluster.

View definition

```

view_definition = ""
SELECT
  TENANT_ID,
  SVR_IP AS IP,
  SVR_PORT AS PORT,
  PCODE,
  PCODE_NAME,
  COUNT,
  TOTAL_TIME,
  TOTAL_SIZE,
  FAILURE,
  TIMEOUT,
  SYNC,
  ASYNC,
  LAST_TIMESTAMP
FROM
  oceanbase.__all_virtual_obrpc_stat
WHERE
  is_serving_tenant(svr_ip, svr_port, effective_tenant_id())
AND
  (EFFECTIVE_TENANT_ID)=1
OR
  TENANT_ID=EFFECTIVE_TENANT_ID()

```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PCODE	bigint(20)	rpc packet code
PCODE_NAME	varchar(256)	The name that corresponds to the RPC packet code.

Name	Data type	Description
COUNT	bigint(20)	The total number of times that the RPC packet code was used.
TOTAL_TIME	bigint(20)	The total amount of time that was spent.
TOTAL_SIZE	bigint(20)	The total amount of data that was sent.
FAILURE	bigint(20)	The number of RPC requests that have failed sending.
TIMEOUT	bigint(20)	The number of sending time-outs.
SYNC	bigint(20)	The number of RPC requests that were sent synchronously.
ASYN	bigint(20)	The number of RPC requests that were sent asynchronously.
LAST_TIMESTAMP	timestamp(6)	The time when the statistics was last updated.

18.5.20.38. gv\$obrpc_incoming

Shows statistics about RPC requests that were received by all OBServers in a cluster.

View definition

```

view_definition = ""
SELECT
  TENANT_ID,
  SVR_IP AS IP,
  SVR_PORT AS PORT,
  PCODE,
  PCODE_NAME,
  ICOUNT AS COUNT,
  ISIZE AS TOTAL_SIZE,
  NET_TIME,
  WAIT_TIME,
  QUEUE_TIME,
  PROCESS_TIME,
  ILAST_TIMESTAMP AS LAST_TIMESTAMP
FROM
  oceanbase.__all_virtual_obrpc_stat
WHERE
  EFFECTIVE_TENANT_ID()=1
OR
  TENANT_ID=EFFECTIVE_TENANT_ID()

```

Fields

Name	Data type	Description
TENANT_ID	bigint(20)	The ID of the tenant.
IP	varchar(32)	The IP address.
PORT	bigint(20)	The port number.
PCODE	bigint(20)	rpc packet code
PCODE_NAME	varchar(256)	The name that corresponds to the RPC packet code.
COUNT	bigint(20)	The number of times that the RPC packet code was used.
TOTAL_SIZE	bigint(20)	The total amount of data that was received.
NET_TIME	bigint(20)	The network time.
WAIT_TIME	bigint(20)	The amount of time that elapsed from the time when the request was received to the time when the request was added to the queue.
QUEUE_TIME	bigint(20)	The amount of time for which the request waited in the queue.
PROCESS_TIME	bigint(20)	The amount of time that was spent on processing the request.
LAST_TIMESTAMP	timestamp(6)	The last update time.

18.5.20.39. gv\$sql

Shows hot-updated SQL statistics for all execution plans of all OBServers in a cluster. Each row in the table records the statistics of a single execution plan. The statistics contain summary information about multiple executions of the plan.

View definition

```
view_definition='SELECT tenant_id AS CON_ID, \
  svr_ip AS SVR_IP, \
  svr_port AS SVR_PORT, \
  plan_id AS PLAN_ID, \
  sql_id AS SQL_ID, \
  type AS TYPE, \
  statement AS SQL_TEXT, \
  plan_hash AS PLAN_HASH_VALUE, \
  first_load_time AS FIRST_LOAD_TIME, \
  last_active_time AS LAST_ACTIVE_TIME, \
  avg_exe_usec AS AVG_EXE_USEC, \
  slowest_exe_time AS SLOWEST_EXE_TIME, \
  slowest_exe_usec as SLOWEST_EXE_USEC, \
  slow_count as SLOW_COUNT, \
  hit_count as HIT_COUNT, \
  executions as EXECUTIONS, \
  disk_reads as DISK_READS, \
  direct_writes as DIRECT_WRITES, \
  buffer_gets as BUFFER_GETS, \
  application_wait_time as APPLICATION_WAIT_TIME, \
  concurrency_wait_time as CONCURRENCY_WAIT_TIME, \
  user_io_wait_time as USER_IO_WAIT_TIME, \
  rows_processed as ROWS_PROCESSED, \
  elapsed_time as ELAPSED_TIME, \
  cpu_time as CPU_TIME \
FROM oceanbase.__all_virtual_plan_stat \
WHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1)'
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The identifier of the SQL statement.
TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
SQL_TEXT	varchar(4096)	The text of the SQL statement.

Name	Data type	Description
PLAN_HASH_VALUE	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time of the first execution.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	timestamp(6)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of plan cache hits.
PLAN_SIZE	bigint(20)	
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.
CPU_TIME	bigint(20) unsigned	The CPU time.
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.

Name	Data type	Description
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The identifier of the SQL statement.
TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
SQL_TEXT	varchar(4096)	The text of the SQL statement.
PLAN_HASH_VALUE	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time of the first execution.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	timestamp(6)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of plan cache hits.
PLAN_SIZE	bigint(20)	
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.

Name	Data type	Description
CPU_TIME	bigint(20) unsigned	The CPU time.
CON_ID	bigint(20)	The ID of the tenant.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
PLAN_ID	bigint(20)	The ID of the execution plan.
SQL_ID	varchar(32)	The identifier of the SQL statement.
TYPE	bigint(20)	The type of the execution plan. Valid values: local, remote, and distribute.
SQL_TEXT	varchar(4096)	The text of the SQL statement.
PLAN_HASH_VALUE	bigint(20) unsigned	The hash value of the execution plan.
FIRST_LOAD_TIME	timestamp(6)	The time of the first execution.
LAST_ACTIVE_TIME	timestamp(6)	The time of the last execution.
AVG_EXE_USEC	bigint(20)	The average amount of execution time.
SLOWEST_EXE_TIME	timestamp(6)	The time when the slowest execution started.
SLOWEST_EXE_USEC	bigint(20)	The amount of time that was spent on the slowest execution.
SLOW_COUNT	bigint(20)	The number of slow queries.
HIT_COUNT	bigint(20)	The number of plan cache hits.
PLAN_SIZE	bigint(20)	
EXECUTIONS	bigint(20)	The number of executions.
DISK_READS	bigint(20)	The number of physical reads from disks.
DIRECT_WRITES	bigint(20)	The number of physical writes.
BUFFER_GETS	bigint(20)	The number of logical reads.
APPLICATION_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all application events.
CONCURRENCY_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all concurrency events.
USER_IO_WAIT_TIME	bigint(20) unsigned	The amount of time that was spent on waiting for all user I/O events.

Name	Data type	Description
ROWS_PROCESSED	bigint(20)	The number of rows to be processed by the SQL statement. If a SELECT statement is executed, the value indicates the number of rows that are returned. If a DELETE, UPDATED, or INSERT statement is executed, the value indicates the number of rows that are affected.
ELAPSED_TIME	bigint(20) unsigned	The amount of time that was spent on processing the request after the request was received.
CPU_TIME	bigint(20) unsigned	The CPU time.

18.5.20.40. gv\$sql_monitor

Shows statistics about slow SQL queries based on execution plans for all OBServers in a cluster. Each slow SQL query has a statistical record that can be used to trace the execution plan.

View definition

```
view_definition='SELECT tenant_id as CON_ID,\
  request_id as SQL_EXEC_ID,\
  job_id as JOB_ID,\
  task_id as TASK_ID,\
  svr_ip as SVR_IP,\
  svr_port as SVR_PORT,\
  sql_exec_start as SQL_EXEC_START, \
  plan_id as PLAN_ID,\
  scheduler_ip as SCHEDULER_IP, \
  scheduler_port as SCHEDULER_PORT, \
  monitor_info as MONITOR_INFO,\
  extend_info as EXTEND_INFO FROM oceanbase.__all_virtual_sql_monitor \
WHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) \
and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1)';
```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SQL_EXEC_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
JOB_ID	bigint(20)	The job ID. The executor uses this ID to identify a segment in the physical execution plan. Job IDs are globally incremental on a single OBServer.

Name	Data type	Description
TASK_ID	bigint(20)	The task ID, which uniquely identifies an execution of a segment that corresponds to a job ID in a distributed execution plan.
SVR_IP	varchar(32)	The IP address.
SVR_PORT	bigint(20)	The port number.
SQL_EXEC_START	timestamp(6)	The time when the execution started.
PLAN_ID	bigint(20)	The plan ID, which uniquely identifies an execution plan in the plan cache on a single OBServer. Plan IDs are incremental values that are managed by the plan cache module. When a new execution plan is added to the plan cache, a plan ID is assigned to the new execution plan.
SCHEDULER_IP	varchar(32)	The IP address of the OBServer that is scheduled to execute the SQL statement.
SCHEDULER_PORT	bigint(20)	The port number of the OBServer that is scheduled to execute the SQL statement.
MONITOR_INFO	varchar(65535)	The relevant information such as the event for which the amount of waited time is the longest and the reception time.
EXTEND_INFO	varchar(65535)	The extended information such as all trace information that was generated in the process of executing the SQL statement.

18.5.20.41. gv\$sql_plan_monitor

Shows statistics about slow SQL queries based on execution plans for all OBServers in a cluster. Each slow SQL query has a statistical record that can be used to trace the execution plan.

View definition

```

view_definition='SELECT tenant_id as CON_ID, \
  request_id as SQL_EXEC_ID, \
  job_id as JOB_ID, \
  task_id as TASK_ID, \
  operation_id as OPERATION_ID, \
  svr_ip as SVR_IP, \
  svr_port as SVR_PORT, \
  sql_exec_start as SQL_EXEC_START, \
  plan_id as PLAN_ID, \
  scheduler_ip as SCHEDULER_IP, \
  scheduler_port as SCHEDULER_PORT, \
  operation as PLAN_OPERATION, \
  monitor_info as MONITOR_INFO, \
  extend_info as EXTEND_INFO FROM oceanbase.__all_virtual_sql_plan_monitor \
  WHERE is_serving_tenant(svr_ip, svr_port, effective_tenant_id()) \
  and (tenant_id = effective_tenant_id() or effective_tenant_id() = 1)'

```

Fields

Name	Data type	Description
CON_ID	bigint(20)	The ID of the tenant.
SQL_EXEC_ID	bigint(20)	The unique ID of the SQL statement. This ID is the MD5 hash value of the parameterized string of the SQL statement.
JOB_ID	bigint(20)	The job ID. The executor uses this ID to identify a segment in the physical execution plan. Job IDs are globally incremental on a single OBServer.
TASK_ID	bigint(20)	The task ID, which uniquely identifies an execution of a segment that corresponds to a job ID in a distributed execution plan.
OPERATION_ID	bigint(20)	The operator ID, which uniquely identifies an operator in a physical execution plan. Operator IDs are generated based on the postorder traversal of the physical execution plan tree.
SVR_IP	varchar(32)	The IP address of the OBServer.
SVR_PORT	bigint(20)	The port number of the OBServer.
SQL_EXEC_START	timestamp(6)	The time when the execution of the SQL statement started.
PLAN_ID	bigint(20)	The plan ID, which uniquely identifies an execution plan in the plan cache on a single OBServer. Plan IDs are incremental values that are managed by the plan cache module. When a new execution plan is added to the plan cache, a plan ID is assigned to the new execution plan.

Name	Data type	Description
SCHEDULER_IP	varchar(32)	The IP address of the OBServer that is scheduled to execute the SQL statement.
SCHEDULER_PORT	bigint(20)	The port number of the OBServer that is scheduled to execute the SQL statement.
PLAN_OPERATION	varchar(32)	The operators that are used in the execution plan.
MONITOR_INFO	varchar(65535)	The execution time statistics for each step in the execution plan.
EXTEND_INFO	varchar(65535)	The extended information.

18.5.21. Information Schema

18.5.21.1. Overview

ApsaraDB for OceanBase supports the INFORMATION_SCHEMA database of MySQL. The stored data in the INFORMATION_SCHEMA database depends on the data that is stored in ApsaraDB for OceanBase. If ApsaraDB for OceanBase does not support a field in the database, the field is retained but the field stores empty values.

The INFORMATION_SCHEMA database stores database metadata.

Metadata is data that provides information about other data, such as database names, table names, data types of columns, and access permissions.

The INFORMATION_SCHEMA database stores the information about all the other databases that are maintained by your server.

The INFORMATION_SCHEMA database contains multiple read-only tables. These read-only tables are views instead of base tables. Therefore, you cannot view the files that are associated with these read-only tables.

Each user can access only specific rows in these read-only tables. The specific rows include the objects on which the user has appropriate read permissions.

Advantages of executing SELECT statements to access metadata

`SELECT ... FROM INFORMATION_SCHEMA` statements are an alternative to SHOW statements. `SELECT...FROM INFORMATION_SCHEMA` statements provide a consistent method for you to access the information that is returned by the supported SHOW statements. The examples of the supported SHOW statements include SHOW DATABASES and SHOW TABLES statements.

`SELECT...FROM INFORMATION_SCHEMA` statements offer the following advantages over SHOW statements:

- `SELECT...FROM INFORMATION_SCHEMA` statements comply with Codd's rules. All the read and write operations are performed on tables.
- You need only to familiarize yourself with the syntax of SELECT statements. You do not need to familiarize yourself with other statements.

If you are familiar with the syntax of SELECT statements, you need only to know the object names.

- You do not need to concern yourself with the issues of adding keywords.
- `SELECT...FROM INFORMATION_SCHEMA` statements can return millions of results. This helps you meet various metadata requirements of applications.

Permissions

To query metadata, you can execute SHOW statements or `SELECT...FROM INFORMATION_SCHEMA` statements. The permissions that are required to execute the two types of statements are the same.

You must have permissions on objects to view the information about the objects.

18.5.21.2. INFORMATION_SCHEMA tables

Table	Description
INFORMATION_SCHEMA.CHARACTER_SETS	Provides the information about available character sets.
INFORMATION_SCHEMA.COLLATIONS	Provides the information about collations for character sets.
INFORMATION_SCHEMA.COLLATION_CHARACTER_SET_APPLICABILITY	Specifies the character set that is used for the collation.
INFORMATION_SCHEMA.COLUMNS	Provides the information about table columns.
INFORMATION_SCHEMA.COLUMN_PRIVILEGES	Provides the information about column permissions.
INFORMATION_SCHEMA.ENGINES	Provides the information about storage engines.
INFORMATION_SCHEMA.EVENTS	Provides the information about events.
INFORMATION_SCHEMA.FILES	Provides the information about tablespace data.
INFORMATION_SCHEMA.GLOBAL_STATUS	Provides the information about the status of the global server variables.
INFORMATION_SCHEMA.GLOBAL_VARIABLES	Provides the information about global server variables.
INFORMATION_SCHEMA.KEY_COLUMN_USAGE	Describes key columns that have constraints.
INFORMATION_SCHEMA.OPTIMIZER_TRACE	Provides the information that is produced by the optimizer tracing feature.
INFORMATION_SCHEMA.PARAMETERS	Provides the information about function and method parameters.
INFORMATION_SCHEMA.PARTITIONS	Provides the information about table partitions.
INFORMATION_SCHEMA.PLUGINS	Provides the information about plug-ins.
INFORMATION_SCHEMA.PROCESSLIST	Provides the information about running threads.
INFORMATION_SCHEMA.PROFILING	Provides the information about statement profiling.
INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS	Provides the information about foreign key constraints.
INFORMATION_SCHEMA.ROUTINES	Provides the information about stored subprograms: stored procedures and stored functions.
INFORMATION_SCHEMA.SCHEMATA	Provides the information about databases.
INFORMATION_SCHEMA.SCHEMA_PRIVILEGES	Provides the information about database permissions.
INFORMATION_SCHEMA.SESSION_STATUS	Provides the information about the status of local variables.
INFORMATION_SCHEMA.SESSION_VARIABLES	Provides the information about local variables.
INFORMATION_SCHEMA.STATISTICS	Provides the information about table indexes.
INFORMATION_SCHEMA.TABLES	Provides the information about database tables.

Table	Description
INFORMATION_SCHEMA.TABLESPACES	Provides the information about tablespaces.
INFORMATION_SCHEMA.TABLE_CONSTRAINTS	Describes the tables that have constraints.
INFORMATION_SCHEMA.TABLE_PRIVILEGES	Provides the information about table permissions.
INFORMATION_SCHEMA.TRIGGERS	Provides the information about triggers.
INFORMATION_SCHEMA.USER_PRIVILEGES	Provides the information about user permissions.
INFORMATION_SCHEMA.USER_RECYCLEBIN	-
INFORMATION_SCHEMA.VIEWS	Provides the information about views that are stored in databases.

This topic does not provide details about tables and columns in the INFORMATION_SCHEMA database. These details are provided in other topics. For each column of the tables in the INFORMATION_SCHEMA database, the following three types of information are provided:

- **Standard name:** specifies the name of the column in an INFORMATION_SCHEMA table. This column name is the standard name that is used in SQL statements.
- **SHOW name:** specifies the equivalent field name in the latest SHOW statement.
- **Remarks:** provides additional information about the column.

18.5.21.3. INFORMATION_SCHEMA.SCHEMATA table

This table provides the information about databases.

Standard name	SHOW name	Remarks
CATALOG_NAME	-	def
SCHEMA_NAME	Database	The name of the database.
DEFAULT_CHARACTER_SET_NAME	-	The default character set.
DEFAULT_COLLATION_NAME	-	The default collation name.
SQL_PATH	-	NULL

The following statements are equivalent:

```
SELECT SCHEMA_NAME AS `Database`
  FROM INFORMATION_SCHEMA.SCHEMATA
 [WHERE SCHEMA_NAME LIKE 'wild']

SHOW DATABASES
 [LIKE 'wild']
```

18.5.21.4. INFORMATION_SCHEMA.TABLES table

This table provides the information about database tables.

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLE_SCHEMA	Table_...	The name of the schema or database where the table is stored.
TABLE_NAME	Table_...	The name of the table.
TABLE_TYPE		The type of the table. The BASE TABLE value indicates a table, the VIEW value indicates a view, and the TEMPORARY value indicates a temporary table.
ENGINE	Engine	The storage engine.
VERSION	Version	The version number.
ROW_FORMAT	Row_format	-
TABLE_ROWS	Rows	-
AVG_ROW_LENGTH	Avg_row_length	The average row size.
DATA_LENGTH	Data_length	The size of the data file. Unit: bytes.
MAX_DATA_LENGTH	Max_data_length	The maximum size of the data file. Unit: bytes.
INDEX_LENGTH	Index_length	The size of the index file. Unit: bytes.
DATA_FREE	Data_free	-
AUTO_INCREMENT	Auto_increment	Specifies whether the values are auto-incremented.
CREATE_TIME	Create_time	The time when the table was created.
UPDATE_TIME	Update_time	The time when the table was last updated.
CHECK_TIME	Check_time	The time when the table was last checked.
TABLE_COLLATION	Collation	The table collation.
CHECKSUM	Checksum	The checksum value.
CREATE_OPTIONS	Create_options	The options that are used in the CREATE TABLE statement.
TABLE_COMMENT	Comment	The description.

Notes:

- In the outputs of SHOW statements, the values of the TABLE_SCHEMA and TABLE_NAME columns are included in the values of a single field.
- The valid values in the TABLE_TYPE column include BASE TABLE and VIEW. The BASE TABLE value indicates a table and the VIEW value indicates a view. If you are using a temporary table, set TABLE_TYPE to TEMPORARY: TABLE_TYPE = TEMPORARY .
- If the table is stored in the INFORMATION_SCHEMA database, the value in the TABLE_ROWS column is NULL.

- The outputs of query statements do not include the information about the default character set of the table. The TABLE_COLLATION column is disabled because the collation name starts with the character set name.

The following statements are equivalent:

```
SELECT table_name FROM INFORMATION_SCHEMA.TABLES
  [WHERE table_schema = 'db_name']
  [WHERE|AND table_name LIKE 'wild']

SHOW TABLES
[FROM db_name]
[LIKE 'wild']
```

18.5.21.5. INFORMATION_SCHEMA.COLUMNS table

This table provides the information about table columns.

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	The name of the database.
TABLE_NAME	-	The name of the table.
COLUMN_NAME	Field	The name of the column.
ORDINAL_POSITION	-	The position of the column. For more information, see the "Notes" section in this topic.
COLUMN_DEFAULT	Default	The default value for the column.
IS_NULLABLE	Null	Specifies whether NULL values can be stored in the column.
DATA_TYPE	Type	The data type of the column.
CHARACTER_MAXIMUM_LENGTH	Type	The maximum length of each string. The length is measured in characters.
CHARACTER_OCTET_LENGTH	-	The maximum length of each string. The length is measured in bytes. For more information, see the "Notes" section in this topic.
NUMERIC_PRECISION	Type	The numeric precision.
NUMERIC_SCALE	Type	The range of the numeric values.
DATETIME_PRECISION	Type	The date and time granularity.
CHARACTER_SET_NAME	-	The name of the character set.
COLLATION_NAME	Collation	The collation name.
COLUMN_TYPE	Type	The data type of the column.

Standard name	SHOW name	Remarks
COLUMN_KEY	Key	The column key.
EXTRA	Extra	The additional information about the column.
PRIVILEGES	Privileges	The permissions.
COLUMN_COMMENT	Comment	The description of the column.

Notes:

- In the outputs of SHOW statements, the Type column lists the data types of different columns that are stored in the COLUMNS table.
- ORDINAL_POSITION is required in some scenarios. For example, you may need to execute the ORDER BY ORDINAL_POSITION statement to sort data based on the ordinal positions of columns. Unlike SHOW statements, SELECT statements returns data that is not automatically sorted.
- If single-byte character sets are used, the CHARACTER_OCTET_LENGTH value is the same as the CHARACTER_MAXIMUM_LENGTH value. If multi-byte character sets are used, the values are different.
- You can obtain the CHARACTER_SET_NAME column value based on the COLLATION_NAME column value. For example, if you execute a SHOW FULL COLUMNS FROM `tbl_name` statement and the latin1_swedish_ci value is displayed in the COLLATION_NAME column, the name of the character set is latin1. The portion that is located before the first underscore (_) specifies the name of the character set.

The following statements are equivalent:

```
SELECT COLUMN_NAME, DATA_TYPE, IS_NULLABLE, COLUMN_KEY, COLUMN_DEFAULT, EXTRA
FROM INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'tbl_name'
[AND table_schema = 'db_name']
[AND column_name LIKE 'wild']

SHOW COLUMNS
FROM tbl_name
[FROM db_name]
[LIKE wild]
```

18.5.21.6. INFORMATION_SCHEMA.STATISTICS table

This table provides the information about table indexes.

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	The name of the database.
TABLE_NAME	Table	The name of the table.
NON_UNIQUE	Non_unique	-

Standard name	SHOW name	Remarks
INDEX_SCHEMA	-	The name of the database.
INDEX_NAME	Key_name	The name of the index.
SEQ_IN_INDEX	Seq_in_index	-
COLUMN_NAME	Column_name	The name of the column.
COLLATION	Collation	-
CARDINALITY	Cardinality	-
SUB_PART	Sub_part	-
PACKED	Packed	-
NULLABLE	Null	-
INDEX_TYPE	Index_type	-
COMMENT	Comment	-
INDEX_COMMENT	-	-
IS_VISIBLE	-	-

The following statements are equivalent:

```
SELECT * FROM INFORMATION_SCHEMA.STATISTICS
  WHERE table_name = 'tbl_name'
  [AND table_schema = 'db_name']
```

```
SHOW INDEX
  FROM tbl_name
  [FROM db_name]
```

18.5.21.7. INFORMATION_SCHEMA.USER_PRIVILEGES table

This table provides the information about user permissions.

Table information

Standard name	SHOW name	Remarks
GRANTEE	-	The name of the account to which the permission is granted. For example, the value can be user'@'host.
TABLE_CATALOG	-	def
PRIVILEGE_TYPE	-	The permission type.
IS_GRANTABLE	-	Specifies whether the user has the permission to execute the GRANT OPTION statement.

18.5.21.8. INFORMATION_SCHEMA.SCHEMA_PRIVILEGES table

This table provides the information about database permissions.

Table information

Standard name	SHOW name	Remarks
GRANTEE	-	The name of the account to which the permission is granted. For example, the value can be user'@'host.
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	-
PRIVILEGE_TYPE	-	The permission type.
IS_GRANTABLE	-	Specifies whether the user has the permission to execute the GRANT OPTION statement. If the user has the permission, the value is YES. If the user does not have the permission, the value is NO.

18.5.21.9. INFORMATION_SCHEMA.TABLE_PRIVILEGES table

This table provides the information about table permissions.

Standard name	SHOW name	Remarks
GRANTEE	-	The name of the account to which the permission is granted. For example, the value can be user'@'host.
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	The name of the database where the table is stored.
TABLE_NAME	-	The name of the table.
PRIVILEGE_TYPE	-	The permission type.
IS_GRANTABLE	-	Specifies whether the user has the permission to execute the GRANT OPTION statement. If the user has the permission, the value is YES. If the user does not have the permission, the value is NO.

The following statements are not equivalent:

```
SELECT ... FROM INFORMATION_SCHEMA.TABLE_PRIVILEGES
```

```
SHOW GRANTS ...
```

The `PRIVILEGE_TYPE` column stores only one of the following values: `SELECT`, `INSERT`, `UPDATE`, `REFERENCES`, `ALTER`, `INDEX`, `DROP`, and `CREATE VIEW`.

18.5.21.10. INFORMATION_SCHEMA.CHARACTER_SETS table

This table provides the information about available character sets.

Table information

Standard name	SHOW name	Remarks
<code>CHARACTER_SET_NAME</code>	Charset	The name of the character set.
<code>DEFAULT_COLLATE_NAME</code>	Default collation	The default collation for the character set.
<code>DESCRIPTION</code>	Description	The description of the character set.
<code>MAXLEN</code>	Maxlen	The maximum length.

 **Note** The `DESCRIPTION` and `MAXLEN` columns are not standard columns. These columns are added to align with the following columns that are displayed in the output of the `SHOW CHARACTER SET` statement: `Description` and `Maxlen`.

The following statements are equivalent:

```
SELECT * FROM INFORMATION_SCHEMA.CHARACTER_SETS
  [WHERE name LIKE 'wild']

SHOW CHARACTER SET
  [LIKE 'wild']
```

18.5.21.11. INFORMATION_SCHEMA.COLLATIONS table

This table provides the information about collations for character sets.

Table information

Standard name	SHOW name	Remarks
<code>COLLATION_NAME</code>	Collation	The collation name.
<code>CHARACTER_SET_NAME</code>	Charset	The name of the character set.
<code>ID</code>	Id	The collation ID.
<code>IS_DEFAULT</code>	Default	Specifies whether the collation is the default collation for the character set.
<code>IS_COMPILED</code>	Compiled	Specifies whether the character set is compiled.
<code>SORTLEN</code>	Sortlen	The number of bytes in the strings when the strings are sorted in the memory. The strings are expressed based on the character set.

Note The CHARACTER_SET_NAME, ID, IS_DEFAULT, IS_COMPILED, and SORTLEN columns are not standard columns. These columns are added to align with the following columns that are displayed in the output of the `SHOW COLLATION` statement: Charset, Id, Default, Compiled, and Sortlen.

The following statements are equivalent:

```
SELECT COLLATION_NAME FROM INFORMATION_SCHEMA.COLLATIONS
[WHERE collation_name LIKE 'wild']

SHOW COLLATION
[LIKE 'wild']
```

18.5.21.12.

INFORMATION_SCHEMA.COLLATION_CHARACTER_SET_APPLICABILITY table

This table specifies the character set that is used for the collation. This table consists of two columns. The two columns correspond to the first two columns that are displayed in the output of the `SHOW COLLATION` statement.

Table information

Standard name	SHOW name	Remarks
COLLATION_NAME	Collation	The collation name.
CHARACTER_SET_NAME	Charset	The name of the character set.

18.5.21.13. INFORMATION_SCHEMA.TABLE_CONSTRAINTS table

This table provides the information about tables that have constraints.

Table information

Standard name	SHOW name	Remarks
CONSTRAINT_CATALOG	-	def
CONSTRAINT_SCHEMA	-	The name of the schema or database to which the constraint belongs.
CONSTRAINT_NAME	-	The name of the constraint.
TABLE_SCHEMA	-	The name of the schema or database where the table is stored.
TABLE_NAME	-	The name of the table.
CONSTRAINT_TYPE	-	The type of the constraint.

Notes:

- The valid values in the `CONSTRAINT_TYPE` column include `UNIQUE`, `PRIMARY KEY`, and `FOREIGN KEY`.
- If the value of the `Non_unique` field is 0, the `UNIQUE` and `PRIMARY KEY` information is the same as the

Key_name column values in the output of the SHOW INDEX statement.

- The CONSTRAINT_TYPE column stores only one of the following values: UNIQUE, PRIMARY KEY, FOREIGN KEY, and CHECK. The data type of the CONSTRAINT_TYPE column is CHAR instead of ENUM.

The CHECK value is unavailable before the CHECK constraint is supported by ApsaraDB for OceanBase.

18.5.21.14. INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS

table

This table provides the information about foreign key constraint.

Standard name	SHOW name	Remarks
CONSTRAINT_CATALOG	-	def
CONSTRAINT_SCHEMA	-	The name of the schema or database to which the foreign key constraint belongs.
CONSTRAINT_NAME	-	The name of the foreign key constraint.
UNIQUE_CONSTRAINT_CATALOG	-	def
UNIQUE_CONSTRAINT_SCHEMA	-	The name of the database. The database contains the unique constraint or the primary key that the foreign key constraint references.
UNIQUE_CONSTRAINT_NAME	-	The name of the unique constraint or the primary key that the foreign key constraint references.
MATCH_OPTION	-	The match rule of the foreign key constraint. The value in the column can be FULL, PARTIAL, or NONE.
UPDATE_RULE	-	The update rule of the foreign key constraint. The value in the column can be CASCADE, SET NULL, SET DEFAULT, RESTRICT, or NO ACTION.
DELETE_RULE	-	The deletion rule of the foreign key constraint. The value in the column can be CASCADE, SET NULL, SET DEFAULT, RESTRICT, or NO ACTION.
TABLE_NAME	-	The name of the table.
REFERENCED_TABLE_NAME	-	The name of the table that is referenced by the foreign key constraint.

 **Note** The value in the TABLE_NAME column is the same as that in the TABLE_NAME column of the INFORMATION_SCHEMA.TABLE_CONSTRAINTS table. For more information, see [INFORMATION_SCHEMA.TABLE_CONSTRAINTS table](#).

18.5.21.15. INFORMATION_SCHEMA.KEY_COLUMN_USAGE table

This table describes the key columns that have constraints.

Table information

Standard name	SHOW name	Remarks
CONSTRAINT_CATALOG	-	def
CONSTRAINT_SCHEMA	-	The name of the database to which the constraint belongs.
CONSTRAINT_NAME	-	The name of the constraint.
TABLE_CATALOG	-	The name of the catalog to which the table belongs. The table contains the field on which the constraint is applied.
TABLE_SCHEMA	-	The name of the schema that stores the table. The table contains the field on which the constraint is applied.
TABLE_NAME	-	The name of the table that includes the constraint. The table contains the field on which the constraint is applied.
COLUMN_NAME	-	The name of the column on which the constraint is implemented.
ORDINAL_POSITION	-	The column position in the constraint. The column position is represented by the sequence number.
POSITION_IN_UNIQUE_CONSTRAINT	-	The ordinal position in the key of the table that is referenced.
REFERENCED_TABLE_SCHEMA	-	The name of the database that is referenced by the constraint.
REFERENCED_TABLE_NAME	-	The name of the table that is referenced by the constraint.
REFERENCED_COLUMN_NAME	-	The name of the column that is referenced by the constraint.

Notes:

- If the constraint is a foreign key, the value in the `COLUMN_NAME` column is the name of the foreign key. The value is not the name of the column that the foreign key references.
- The value in the `ORDINAL_POSITION` column specifies the column position in the constraint instead of the column position in the table. Column positions are represented by sequence numbers that start from 1.
- For unique and primary key constraints, the value in the `POSITION_IN_UNIQUE_CONSTRAINT` column is `NULL`. For foreign key constraints, the value in the `POSITION_IN_UNIQUE_CONSTRAINT` column is the ordinal position in the key of the table that is referenced.

For example, execute the following statements to create the t1 and t3 tables:

```

CREATE TABLE t1
(
s1 INT,
s2 INT,
s3 INT,
PRIMARY KEY(s3)
);

CREATE TABLE t3
(
s1 INT,
s2 INT,
s3 INT,
KEY(s1),
CONSTRAINT CO FOREIGN KEY (s2) REFERENCES t1(s3)
);

```

The KEY_COLUMN_USAGE table stores two rows for the t1 and t3 tables:

- One row contains CONSTRAINT_NAME='PRIMARY ', TABLE_NAME='t1', COLUMN_NAME='s3 ', ORDINAL_POSITION=1, POSITION_IN_UNIQUE_CONSTRAINT=NULL.
- The other row contains CONSTRAINT_NAME='CO ', TABLE_NAME='t3', COLUMN_NAME='s2 ', ORDINAL_POSITION=1, POSITION_IN_UNIQUE_CONSTRAINT=1 .

18.5.21.16. INFORMATION_SCHEMA.ROUTINES table

This table provides the information about stored subprograms: stored procedures and stored functions.

Table information

Standard name	Description
SPECIFIC_NAME	The name of the function.
ROUTINE_CATALOG	def
ROUTINE_SCHEMA	The name of the schema or database to which the function belongs.
ROUTINE_NAME	The name of the function.
ROUTINE_TYPE	The type of the stored subprogram. Valid values: PROCEDURE and FUNCTION.
DATA_TYPE	The data type of the return value.
CHARACTER_MAXIMUM_LENGTH	The maximum length of the character set.
CHARACTER_OCTET_LENGTH	The maximum length of the value. If single-byte character sets are used, the CHARACTER_OCTET_LENGTH value is the same as the CHARACTER_MAXIMUM_LENGTH value. If multi-byte character sets are used, the values are different.

Standard name	Description
NUMERIC_PRECISION	The numerical precision.
NUMERIC_SCALE	The range of numeric values.
DATETIME_PRECISION	The date and time granularity.
CHARACTER_SET_NAME	The name of the character set.
COLLATION_NAME	The collation name for the character set.
DTD_IDENTIFIER	The identifier of the data type descriptor. The data type is returned by the function. The identifier is unique among the data type descriptors that are assigned to the function.
ROUTINE_BODY	The query language that is used to define the function. The value is always SQL.
ROUTINE_DEFINITION	The source code text of the function. If the function is not owned by the current user, the value is NULL.
EXTERNAL_NAME	NULL
EXTERNAL_LANGUAGE	NULL
PARAMETER_STYLE	SQL
IS_DETERMINISTIC	Specifies whether the function is declared as immutable. If the function is declared as immutable, the value is YES. In this scenario, the function is a deterministic function in SQL. If the function is not declared as immutable, the value is NO.
SQL_DATA_ACCESS	The setting of data access. The value is always MODIFIES. This means that the function may modify SQL data.
SQL_PATH	NULL
SECURITY_TYPE	If the function runs based on the permissions of the current user, the value is INVOKER. If the function runs based on the permissions of the user who defines the function, the value is DEFINER.
CREATED	The date and time when the function was created. The value is a TIMESTAMP value.
LAST_ALTERED	The date and time when the stored subprogram was last modified. The value is a TIMESTAMP value.
SQL_MODE	-
ROUTINE_COMMENT	The description.
DEFINER	The user who defines the function.
CHARACTER_SET_CLIENT	The session value of the character_set_client system variable when the function was created.
COLLATION_CONNECTION	The session value of the collation_connection system variable when the stored subprogram was created.

Standard name	Description
DATABASE_COLLATION	The collation of the database with which the stored subprogram is associated.

18.5.21.17. INFORMATION_SCHEMA.VIEWS table

This table provides the information about the views that are stored in databases.

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	The name of the schema or database to which the view belongs.
TABLE_NAME	-	The name of the view.
VIEW_DEFINITION	-	The SELECT statement that defines the view. If the current user is not the owner of the view, the value is NULL.
CHECK_OPTION	-	NONE
IS_UPDATABLE	-	Specifies whether the view can be updated.
DEFINER	-	The user who created the view.
SECURITY_TYPE	-	Valid values: DEFINER and INVOKER.
CHARACTER_SET_CLIENT	-	The session value of the character_set_client system variable when the view was created.
COLLATION_CONNECTION	-	The session value of the collation_connection system variable when the view was created.

Notes:

- If you do not have the permission to execute the `SHOW VIEW` statement, you cannot access the `INFORMATION_SCHEMA.VIEWS` table. This permission is newly added.
- The `VIEW_DEFINITION` column stores most of the information about the fields that you use to create views. You can execute the `SHOW CREATE VIEW` statement to view the statement that creates the specified view. To obtain the `SELECT` statement that creates the specified view, skip the words that are located before the `SELECT` keyword and skip the `WITH CHECK OPTION` words in the SQL statement.

For example, you can execute the following SQL statement to create a view:

```
CREATE VIEW v AS
SELECT s2,s1 FROM t
WHERE s1 > 5
ORDER BY s1
WITH CHECK OPTION;
```

In this example, the view is created based on the following SELECT statement:

```
SELECT s2,s1 FROM t WHERE s1 > 5 ORDER BY s1
```

Notes:

- The value in the CHECK_OPTION column is always NONE.
- If the view can be updated, the value in the IS_UPDATABLE column is YES. If the view cannot be updated, the value in the column is NO.
- The DEFINER column specifies the user who defined the view. The valid values in the SECURITY_TYPE column are DEFINER or INVOKER.

18.5.21.18. INFORMATION_SCHEMA.TRIGGERS table

This table provides the information about triggers.

You must have the SUPER permission to view the INFORMATION_SCHEMA.TRIGGERS table.

Table information

Standard name	SHOW name	Remarks
TRIGGER_CATALOG	-	def
TRIGGER_SCHEMA	-	The name of the database to which the trigger belongs.
TRIGGER_NAME	Trigger	The name of the trigger.
EVENT_MANIPULATION	Event	The event that activates the trigger. The value in the column is INSERT, UPDATE, or DELETE.
EVENT_OBJECT_CATALOG	-	def
EVENT_OBJECT_SCHEMA	-	The name of the database. The database stores the table with which the trigger is associated.
EVENT_OBJECT_TABLE	Table	The name of the table with which the trigger is associated.
ACTION_ORDER	-	-
ACTION_CONDITION	-	NULL
ACTION_STATEMENT	Statement	The statement that is executed if the trigger is activated. The value in the column is always EXECUTE PROCEDURE. <i>function(...)</i> .
ACTION_ORIENTATION	-	The value is always ROW. The value indicates that the trigger applies to each processed row.
ACTION_TIMING	Timing	Specifies whether the trigger is activated before or after the event. The valid values are BEFORE and AFTER.
ACTION_REFERENCE_OLD_TABLE	-	NULL
ACTION_REFERENCE_NEW_TABLE	-	NULL

Standard name	SHOW name	Remarks
ACTION_REFERENCE_OLD_ROW	-	OLD
ACTION_REFERENCE_NEW_ROW	-	NEW
CREATED	Created	NULL
SQL_MODE	sql_mode	
DEFINER	Definer	The user who created the trigger.
CHARACTER_SET_CLIENT	character_set_client	The session value of the character_set_client system variable when the trigger was created.
COLLATION_CONNECTION	collation_connection	The session value of the collation_connection system variable when the trigger was created.
DATABASE_COLLATION	Database Collation	The collation of the database with which the trigger is associated.

Notes:

- The TRIGGER_NAME column specifies the name of the trigger. The TRIGGER_SCHEMA column specifies the name of the database where the trigger occurs.
- The valid values in the EVENT_MANIPULATION column are INSERT, DELETE, and UPDATE.
- Each trigger is associated with only one table. The EVENT_OBJECT_SCHEMA column specifies the name of the database. The EVENT_OBJECT_TABLE column specifies the name of the table. The database stores the table with which the trigger is associated.
- The ACTION_ORDER column specifies the ordinal position of the trigger action in the list of similar triggers that are applied on the same table. The value in the ACTION_ORDER column is always 0. The system does not allow two or more triggers that have the same EVENT_MANIPULATION and ACTION_TIMING values to take effect on the same table.
- The ACTION_STATEMENT column specifies the statement that is executed if the trigger is activated. The text in this column is the same as the text in the Statement column that is displayed in the output of the `SHOW TRIGGERS` statement.

 **Note** SHOW The text in the Statement column in the output of SHOW TRIGGERS statement uses UTF-8 encoding.

- The value in the ACTION_ORIENTATION column is always ROW.
- The valid values in the ACTION_TIMING column are BEFORE and AFTER.
- The ACTION_REFERENCE_OLD_ROW column specifies the previous column identifier, and the ACTION_REFERENCE_NEW_ROW column specifies the new column identifier. Therefore, the value in the ACTION_REFERENCE_OLD_ROW column is always OLD, and the value in the ACTION_REFERENCE_NEW_ROW column is always NEW.
- The SQL_MODE column specifies the valid server SQL mode that was used when the trigger was created.

 **Note** The trigger remains valid after the trigger is activated, regardless of the current server SQL mode.

The value range for the SQL_MODE column is the same as that of the sql_mode system variable.

- The values in the following columns are always NULL: TRIGGER_CATALOG, EVENT_OBJECT_CATALOG, ACTION_CONDITION, ACTION_REFERENCE_OLD_TABLE, and CREATED.

18.5.21.19. INFORMATION_SCHEMA.TABLESPACE table

This table provides the information about tablespaces. However, ApsaraDB for OceanBase does not use tablespaces. Therefore, the INFORMATION_SCHEMA.TABLESPACE table is not required in ApsaraDB for OceanBase. The INFORMATION_SCHEMA.TABLESPACE table in ApsaraDB for OceanBase is an empty table.

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLESPACE_NAME	-	-
ENGINE	-	-
TABLESPACE_TYPE	-	-
LOGFILE_GROUP_NAME	-	-
EXTENT_SIZE	-	-
AUTOEXTEND_SIZE	-	-
MAXIMUM_SIZE	-	-
NODEGROUP_ID	-	-
TABLESPACE_COMMENT	-	-

18.5.21.20. INFORMATION_SCHEMA.PARTITIONS table

Table information

Standard name	SHOW name	Remarks
TABLE_CATALOG	-	def
TABLE_SCHEMA	-	The name of the database.
TABLE_NAME	-	The name of the table.
PARTITION_NAME	-	The name of the partition.
SUBPARTITION_NAME	-	The name of the subpartition.
PARTITION_ORDINAL_POSITION	-	The original ordinal position of the partition.
SUBPARTITION_ORDINAL_POSITION	-	The original ordinal position of the subpartition.
PARTITION_METHOD	-	The partitioning type. For example, hash partitioning or list partitioning is used. The default partitioning type is hash partitioning.

Standard name	SHOW name	Remarks
SUBPARTITION_METHOD	-	The subpartitioning type. The default subpartitioning type is hash partitioning.
PARTITION_EXPRESSION	-	The partition expression.
SUBPARTITION_EXPRESSION	-	The subpartition expression.
PARTITION_DESCRIPTION	-	The partition description.
TABLE_ROWS	-	The number of table rows in the partition.
AVG_ROW_LENGTH	-	The average length of the data in the table rows.
DATA_LENGTH	-	The data length. Unit: bytes.
MAX_DATA_LENGTH	-	The maximum length of the data in the table rows that are stored in the partition or subpartition. Unit: bytes.
INDEX_LENGTH	-	The size of the index file. Unit: bytes.
DATA_FREE	-	
CREATE_TIME	-	The time when the partition or subpartition was created.
UPDATE_TIME	-	The time when the partition or subpartition was last modified.
CHECK_TIME	-	The time when the table that includes the partition or subpartition was last checked.
CHECKSUM	-	The checksum value.
PARTITION_COMMENT	-	The additional information about the partition.
NODEGROUP	-	-
TABLESPACE_NAME	-	The name of the tablespace.

Notes:

- The PARTITIONS table is not a standard SQL table in the INFORMATION_SCHEMA database.
- One row is stored in the PARTITIONS table for each non-partitioned table. The values in the following columns are NULL: PARTITION_NAME, SUBPARTITION_NAME, PARTITION_ORDINAL_POSITION, SUBPARTITION_ORDINAL_POSITION, PARTITION_METHOD, SUBPARTITION_METHOD, PARTITION_EXPRESSION, SUBPARTITION_EXPRESSION, and PARTITION_DESCRIPTION. The columns store no data.

18.5.21.21. INFORMATION_SCHEMA.EVENTS table

This table provides the information about events.

Standard name	SHOW name	Remarks
EVENT_CATALOG	-	def

Standard name	SHOW name	Remarks
EVENT_SCHEMA	Db	The name of the database to which the event belongs.
EVENT_NAME	Name	The name of the event.
DEFINER	Definer	user_name!'@'host_name'
TIME_ZONE	Time zone	The time zone.
EVENT_BODY	-	The query language.
EVENT_DEFINITION	-	The statement that is executed if the event occurs.
EVENT_TYPE	Type	The repetition type of the event. Events are divided into one-time events and recurring events.
EXECUTE_AT	Execute at	The time when the event occurred.
INTERVAL_VALUE	Interval value	The time interval at which the event recurs.
INTERVAL_FIELD	Interval field	-
SQL_MODE	-	-
STARTS	Starts	The start date and time of a recurring event.
ENDS	Ends	The end date and time of a recurring event.
STATUS	Status	The event status. Valid values: ENABLED, DISABLED, and SLAVESIDE_DISABLED.
ON_COMPLETION	-	Valid values: PRESERVE and NOT PRESERVE.
CREATED	-	The date and time when the event was created. The value is a TIMESTAMP value.
LAST_ALTERED	-	The date and time when the event was last modified.
LAST_EXECUTED	-	The date and time when the event was last run.
EVENT_COMMENT	-	The description about the event.
ORIGINATOR	Originator	The ID of the server on which the event was created.
CHARACTER_SET_CLIENT	character_set_client	The session value of the character_set_client system variable when the event was created.

Standard name	SHOW name	Remarks
COLLATION_CONNECTION	collation_connection	The session value of the collation_connection system variable when the event was created.
DATABASE_COLLATION	Database Collation	The collation of the database with which the event is associated.

18.5.21.22. INFORMATION_SCHEMA.FILES table

This table provides the information about tablespace data.

Standard name	SHOW name	Remarks
FILE_ID	-	-
FILE_NAME	-	-
FILE_TYPE	-	-
TABLESPACE_NAME	-	-
TABLE_CATALOG	-	-
TABLE_SCHEMA	-	-
TABLE_NAME	-	-
LOGFILE_GROUP_NAME	-	-
LOGFILE_GROUP_NUMBER	-	-
ENGINE	-	-
FULLTEXT_KEYS	-	-
DELETED_ROWS	-	-
UPDATE_COUNT	-	-
FREE_EXTENTS	-	-
TOTAL_EXTENTS	-	-
EXTENT_SIZE	-	-
INITIAL_SIZE	-	-
MAXIMUM_SIZE	-	-
AUTOEXTEND_SIZE	-	-
CREATION_TIME	-	-
LAST_UPDATE_TIME	-	-
LAST_ACCESS_TIME	-	-
RECOVER_TIME	-	-

Standard name	SHOW name	Remarks
TRANSACTION_COUNTER	-	-
VERSION	-	-
ROW_FORMAT	-	-
TABLE_ROWS	-	-
AVG_ROW_LENGTH	-	-
DATA_LENGTH	-	-
MAX_DATA_LENGTH	-	-
INDEX_LENGTH	-	-
DATA_FREE	-	-
CREATE_TIME	-	-
UPDATE_TIME	-	-
CHECK_TIME	-	-
CHECKSUM	-	-
STATUS	-	-
EXTRA	-	-

Notes:

- The values in the FILE_ID column are auto-incremented.
- The FILE_NAME column specifies the name of the data file that is created by executing the `CREATE TABLESPACE` or `ALTER TABLESPACE` statement.
- The FILE_TYPE column specifies the type of the tablespace file.
- The TABLESPACE_NAME column specifies the name of the tablespace.
- The value in the TABLESPACE_CATALOG column is always NULL.
- The TABLE_NAME column specifies the name of the table.
- The value in the EXTENT_SIZE column is always 0.
- You cannot execute SHOW statements to view the data that is stored in the INFORMATION_SCHEMA.FILES table.

18.5.21.23. INFORMATION_SCHEMA.GLOBAL_STATUS table

This table provides the information about the status of global variables. To view the global variables, execute the `SHOW GLOBAL STATUS` statement.

Standard name	SHOW name	Remarks
VARIABLE_NAME	Variable_name	The name of the variable.
VARIABLE_VALUE	Value	The value of the variable.

The data type of the VARIABLE_VALUE column is VARCHAR(1024).

18.5.21.24. INFORMATION_SCHEMA.GLOBAL_VARIABLES table

This table provides the information about global variables. To view the global variables, execute the `SHOW GLOBAL VARIABLES` statement.

Standard name	SHOW name	Remarks
VARIABLE_NAME	Variable_name	The name of the variable.
VARIABLE_VALUE	Value	The value of the variable.

The data type of the VARIABLE_VALUE column is VARCHAR(1024).

18.5.21.25. INFORMATION_SCHEMA.PROCESSLIST table

This table provides the information about running threads.

Standard name	SHOW name	Remarks
ID	Id	The ID of the thread.
USER	User	The user.
HOST	Host	The name of the host.
DB	db	The name of the database.
COMMAND	Command	The command.
TIME	Time	The time.
STATE	State	The current status of the thread.
INFO	Info	The statement that is being executed on the thread.

The following statements are equivalent:

```
SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST
```

```
SHOW FULL PROCESSLIST
```

18.5.21.26. INFORMATION_SCHEMA.SESSION_STATUS table

This table provides the information about local variables. To view the local variables, execute the `SHOW SESSION STATUS` statement.

Standard name	SHOW name	Remarks
VARIABLE_NAME	Variable_name	The name of the variable.
VARIABLE_VALUE	Value	The value of the variable.

The data type of the VARIABLE_VALUE column is VARCHAR(1024).

18.5.21.27. INFORMATION_SCHEMA.SESSION_VARIABLES table

This table provides the information about local variables. To view the local variables, execute the `SHOW SESSION VARIABLES` statement.

Standard name	SHOW name	Remarks
VARIABLE_NAME	Variable_name	The name of the variable.
VARIABLE_VALUE	Value	The value of the variable.

The data type of the VARIABLE_VALUE column is VARCHAR(1024).

18.5.21.28. INFORMATION_SCHEMA.PROFILING table

This table provides the information about statement profiling. The profiling information in this table is the same as that in the output of the SHOW PROFILES or SHOW PROFILE statement.

Standard name	SHOW name	Remarks
QUERY_ID	Query_ID	-
SEQ	-	-
STATE	Status	-
DURATION	Duration	-
CPU_USER	CPU_user	-
CPU_SYSTEM	CPU_system	-
CONTEXT_VOLUNTARY	Context_voluntary	-
CONTEXT_INVOLUNTARY	Context_involuntary	-
BLOCK_OPS_IN	Block_ops_in	-
BLOCK_OPS_OUT	Block_ops_out	-
MESSAGES_SENT	Messages_sent	-
MESSAGES_RECEIVED	Messages_received	-
PAGE_FAULTS_MAJOR	Page_faults_major	-
PAGE_FAULTS_MINOR	Page_faults_minor	-
SWAPS	Swaps	-
SOURCE_FUNCTION	Source_function	-
SOURCE_FILE	Source_file	-
SOURCE_LINE	Source_line	-

18.5.21.29. INFORMATION_SCHEMA.PARAMETERS table

This table provides the information about the parameters for functions and methods.

Standard name	SHOW name	Remarks
SPECIFIC_CATALOG	-	def
SPECIFIC_SCHEMA	db	The name of the database that stores the function.
SPECIFIC_NAME	name	The name of the function.
ORDINAL_POSITION	-	The ordinal position of the parameter in the parameter list of the function. For example, the value can be 1, 2, or 3. The value 0 indicates the ordinal position of the row that describes the return value of the function.
PARAMETER_MODE	-	The type of the parameter. Valid values: <ul style="list-style-type: none"> IN. OUT. INOUT. If the value of the parameter is returned by a function, the PARAMETER_MODE value is NULL.
PARAMETER_NAME	-	The name of the parameter. NULL is returned for a response parameter.
DATA_TYPE	-	The data type of the parameter.
CHARACTER_MAXIMUM_LENGTH	-	The maximum length of the parameter value. The length is measured in characters.
CHARACTER_OCTET_LENGTH	-	The maximum length of the value. If single-byte character sets are used, the CHARACTER_OCTET_LENGTH value is the same as the CHARACTER_MAXIMUM_LENGTH value. If multi-byte character sets are used, the values are different.
NUMERIC_PRECISION	-	The numeric precision.
NUMERIC_SCALE	-	The range of numeric values.
DATETIME_PRECISION	-	The time granularity.
CHARACTER_SET_NAME	-	The name of the character set.
COLLATION_NAME	-	The collation name.
DTD_IDENTIFIER	-	The identifier of the data type descriptor for the parameter.
ROUTINE_TYPE	type	{PROCEDURE FUNCTION}

18.5.21.30. INFORMATION_SCHEMA.OPTIMIZER_TRACE table

Standard name	SHOW name	Remarks
QUERY	-	The query statement.
TRACE	-	The trace record.
MISSING_BYTES_BEYOND_MAX_MEM_SIZE	-	Reserved.
MISSING_PRIVILEGES	-	Reserved.

18.5.21.31. INFORMATION_SCHEMA.ENGINES table

Standard name	SHOW name	Remarks
ENGINE	-	-
SUPPORT	-	-
COMMENT	-	-
TRANSACTIONS	-	-
XA	-	-
SAVEPOINTS	-	-

18.5.21.32. INFORMATION_SCHEMA.PLUGINS table

Standard name	SHOW name	Remarks
NAME	-	-
OWNER	-	-
DB_NAME	-	-
USED	-	-
TIMESTAMP	-	-
VERSION	-	-
SQL_TEXT	-	-
SIGNATURE	-	-
COMPATIBLE	-	-
ENABLED	-	-
FORMAT	-	-
OUTLINE_CONTENT	-	-
OUTLINE_TARGET	-	-

18.5.22. MySQL dictionary tables

To ensure compatibility with MySQL, ApsaraDB for OceanBase provides a MySQL database for each tenant. You can execute the `SHOW DATABASES` statement to view the MySQL database.

The MySQL database stores a set of dictionary tables that have the same schema as MySQL tables. [MySQL dictionary tables](#) describes these dictionary tables.

MySQL dictionary tables

Dictionary table	Description
db	Provides the information about databases and permissions.
user	Provides the information about users and permissions.
time_zone	Provides the information about time zones.
time_zone_name	Provides the information about time zones.
time_zone_name	
time_zone_transition_type	Provides the information about time zones.

18.5.23. Error codes

18.5.23.1. ApsaraDB for OceanBase error codes

```

DEFINE_ERROR(OB_SUCCESS, 0, 0, "00000", "Success");

////////////////////////////////////
//error code for common -4000 ---- -4500
////////////////////////////////////
DEFINE_ERROR(OB_ERROR, -4000, -1, "HY000", "Common error");
DEFINE_ERROR(OB_OBJ_TYPE_ERROR, -4001, -1, "HY004", "Object type error");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT, -4002, ER_WRONG_ARGUMENTS, "HY000", "Invalid argument", "Incorrect arg
uments to %s");
DEFINE_ERROR(OB_ARRAY_OUT_OF_RANGE, -4003, -1, "42000", "Array index out of range");
DEFINE_ERROR(OB_SERVER_LISTEN_ERROR, -4004, -1, "08S01", "Failed to listen to the port");
DEFINE_ERROR(OB_INIT_TWICE, -4005, -1, "HY000", "The object is initialized twice");
DEFINE_ERROR(OB_NOT_INIT, -4006, -1, "HY000", "The object is not initialized");
DEFINE_ERROR_EXT(OB_NOT_SUPPORTED, -4007, ER_NOT_SUPPORTED_YET, "0A000", "Not supported feature or function"
, "%s not supported");
DEFINE_ERROR(OB_ITER_END, -4008, -1, "HY000", "End of iteration");
DEFINE_ERROR(OB_IO_ERROR, -4009, -1, "58030", "IO error");
DEFINE_ERROR(OB_ERROR_FUNC_VERSION, -4010, -1, "HY000", "Wrong RPC command version");
DEFINE_ERROR(OB_PACKET_NOT_SENT, -4011, -1, "HY000", "Can not send packet");
DEFINE_ERROR(OB_TIMEOUT, -4012, -1, "HY000", "Timeout");
DEFINE_ERROR(OB_ALLOCATE_MEMORY_FAILED, -4013, -1, "HY001", "No memory or reach tenant memory limit");
DEFINE_ERROR(OB_INNER_STAT_ERROR, -4014, -1, "HY000", "Inner state error");
DEFINE_ERROR(OB_ERR_SYS, -4015, -1, "HY000", "System error");
DEFINE_ERROR_EXT(OB_ERR_UNEXPECTED, -4016, -1, "HY000", "Oooooooooooooops". "%s");

```

```

DEFINE_ERROR(ER_ACCESS_DENIED_ERROR, -4016, -1, "HY000", "Access denied");
DEFINE_ERROR(OB_ENTRY_EXIST, -4017, -1, "HY000", "Entry already exist");
DEFINE_ERROR(OB_ENTRY_NOT_EXIST, -4018, -1, "HY000", "Entry not exist");
DEFINE_ERROR(OB_SIZE_OVERFLOW, -4019, -1, "HY000", "Size overflow");
DEFINE_ERROR(OB_REF_NUM_NOT_ZERO, -4020, -1, "HY000", "Reference count is not zero");
DEFINE_ERROR(OB_CONFLICT_VALUE, -4021, -1, "HY000", "Conflict value");
DEFINE_ERROR(OB_ITEM_NOT_SETTED, -4022, -1, "HY000", "Item not set");
DEFINE_ERROR(OB_EAGAIN, -4023, -1, "HY000", "Try again");
DEFINE_ERROR(OB_BUF_NOT_ENOUGH, -4024, -1, "HY000", "Buffer not enough");
DEFINE_ERROR(OB_PARTIAL_FAILED, -4025, -1, "HY000", "Partial failed");
DEFINE_ERROR(OB_READ_NOTHING, -4026, -1, "02000", "Nothing to read");
DEFINE_ERROR(OB_FILE_NOT_EXIST, -4027, ER_FILE_NOT_FOUND, "HY000", "File not exist");
DEFINE_ERROR(OB_DISCONTINUOUS_LOG, -4028, -1, "HY000", "Log entry not continuous");
DEFINE_ERROR(OB_SCHEMA_ERROR, -4029, -1, "HY000", "Schema error");
DEFINE_ERROR(OB_TENANT_OUT_OF_MEM, -4030, -1, "HY000", "Over tenant memory limits");
DEFINE_ERROR(OB_UNKNOWN_OBJ, -4031, -1, "HY004", "Unknown object");
DEFINE_ERROR(OB_NO_MONITOR_DATA, -4032, -1, "02000", "No monitor data");
DEFINE_ERROR(OB_SERIALIZE_ERROR, -4033, -1, "HY000", "Serialize error");
DEFINE_ERROR(OB_DESERIALIZE_ERROR, -4034, -1, "HY000", "Deserialize error");
DEFINE_ERROR(OB_AIO_TIMEOUT, -4035, -1, "HY000", "Asynchronous IO error");
DEFINE_ERROR(OB_NEED_RETRY, -4036, -1, "HY000", "Need retry");
DEFINE_ERROR(OB_TOO_MANY_SSTABLE, -4037, -1, "HY000", "Too many sstable");
DEFINE_ERROR(OB_NOT_MASTER, -4038, -1, "HY000", "The observer or zone is not the master");
DEFINE_ERROR(OB_DECRYPT_FAILED, -4041, -1, "HY000", "Decrypt error");
DEFINE_ERROR(OB_USER_NOT_EXIST, -4042, ER_PASSWORD_NO_MATCH, "42000", "Can not find any matching row in the user table");
DEFINE_ERROR_EXT(OB_PASSWORD_WRONG, -4043, ER_ACCESS_DENIED_ERROR, "42000", "Access denied for user", "Access denied for user '%.*s'@'%.*s' (using password: %s)");
DEFINE_ERROR(OB_SKEY_VERSION_WRONG, -4044, -1, "HY000", "Wrong skey version");
DEFINE_ERROR(OB_NOT_REGISTERED, -4048, -1, "HY000", "Not registered");
DEFINE_ERROR(OB_WAITQUEUE_TIMEOUT, -4049, 4012, "HY000", "Task timeout and not executed");
DEFINE_ERROR(OB_NOT_THE_OBJECT, -4050, -1, "HY000", "Not the object");
DEFINE_ERROR(OB_ALREADY_REGISTERED, -4051, -1, "HY000", "Already registered");
DEFINE_ERROR(OB_LAST_LOG_RUINED, -4052, -1, "HY000", "Corrupted log entry");
DEFINE_ERROR(OB_NO_CS_SELECTED, -4053, -1, "HY000", "No ChunkServer selected");
DEFINE_ERROR(OB_NO_TABLETS_CREATED, -4054, -1, "HY000", "No tablets created");
DEFINE_ERROR(OB_INVALID_ERROR, -4055, -1, "HY000", "Invalid entry");
DEFINE_ERROR(OB_DECIMAL_OVERFLOW_WARN, -4057, -1, "HY000", "Decimal overflow warning");
DEFINE_ERROR(OB_DECIMAL_UNLEGAL_ERROR, -4058, -1, "HY000", "Decimal overflow error");
DEFINE_ERROR(OB_OBJ_DIVIDE_ERROR, -4060, -1, "HY000", "Divide error");
DEFINE_ERROR(OB_NOT_A_DECIMAL, -4061, -1, "HY000", "Not a decimal");
DEFINE_ERROR(OB_DECIMAL_PRECISION_NOT_EQUAL, -4062, -1, "HY104", "Decimal precision error");
DEFINE_ERROR(OB_EMPTY_RANGE, -4063, -1, "HY000", "Empty range");
DEFINE_ERROR(OB_SESSION_KILLED, -4064, -1, "HY000", "Session killed");
DEFINE_ERROR(OB_LOG_NOT_SYNC, -4065, -1, "HY000", "Log not sync");
DEFINE_ERROR(OB_DIR_NOT_EXIST, -4066, ER_CANT_READ_DIR, "HY000", "Directory not exist");
DEFINE_ERROR(OB_SESSION_NOT_FOUND, -4067, 4012, "HY000", "RPC session not found");

```

```

DEFINE_ERROR(OB_INVALID_LOG, -4068, -1, "HY000", "Invalid log");
DEFINE_ERROR(OB_INVALID_DATA, -4070, -1, "HY000", "Invalid data");
DEFINE_ERROR(OB_ALREADY_DONE, -4071, -1, "HY000", "Already done");
DEFINE_ERROR(OB_CANCELED, -4072, -1, "HY000", "Operation canceled");
DEFINE_ERROR(OB_LOG_SRC_CHANGED, -4073, -1, "HY000", "Log source changed");
DEFINE_ERROR(OB_LOG_NOT_ALIGN, -4074, -1, "HY000", "Log not aligned");
DEFINE_ERROR(OB_LOG_MISSING, -4075, -1, "HY000", "Log entry missed");
DEFINE_ERROR(OB_NEED_WAIT, -4076, -1, "HY000", "Need wait");
DEFINE_ERROR(OB_NOT_IMPLEMENT, -4077, -1, "0A000", "Not implemented feature");
DEFINE_ERROR(OB_DIVISION_BY_ZERO, -4078, ER_DIVISION_BY_ZERO, "42000", "Divided by zero");
DEFINE_ERROR(OB_EXCEED_MEM_LIMIT, -4080, -1, "HY013", "exceed memory limit");
DEFINE_ERROR(OB_RESULT_UNKNOWN, -4081, -1, "HY000", "Unknown result");
DEFINE_ERROR(OB_NO_RESULT, -4084, -1, "02000", "No result");
DEFINE_ERROR(OB_QUEUE_OVERFLOW, -4085, -1, "HY000", "Queue overflow");
DEFINE_ERROR(OB_TERM_LAGGED, -4097, -1, "HY000", "Term lagged");
DEFINE_ERROR(OB_TERM_NOT_MATCH, -4098, -1, "HY000", "Term not match");
DEFINE_ERROR(OB_START_LOG_CURSOR_INVALID, -4099, -1, "HY000", "Invalid log cursor");
DEFINE_ERROR(OB_LOCK_NOT_MATCH, -4100, -1, "HY000", "Lock not match");
DEFINE_ERROR(OB_DEAD_LOCK, -4101, ER_LOCK_DEADLOCK, "HY000", "Deadlock");
DEFINE_ERROR(OB_PARTIAL_LOG, -4102, -1, "HY000", "Incomplete log entry");
DEFINE_ERROR(OB_CHECKSUM_ERROR, -4103, -1, "42000", "Data checksum error");
DEFINE_ERROR(OB_INIT_FAIL, -4104, -1, "HY000", "Initialize error");
DEFINE_ERROR(OB_NOT_ENOUGH_STORE, -4106, -1, "HY000", "not enough commitlog store");
DEFINE_ERROR(OB_BLOCK_SWITCHED, -4107, -1, "HY000", "block switched when fill commitlog");
DEFINE_ERROR(OB_STATE_NOT_MATCH, -4109, -1, "HY000", "Server state or role not the same as expected");
DEFINE_ERROR(OB_READ_ZERO_LOG, -4110, -1, "HY000", "Read zero log");
DEFINE_ERROR(OB_BLOCK_NEED_FREEZE, -4111, -1, "HY000", "block need freeze");
DEFINE_ERROR(OB_BLOCK_FROZEN, -4112, -1, "HY000", "block frozen");
DEFINE_ERROR(OB_IN_FATAL_STATE, -4113, -1, "HY000", "In FATAL state");
DEFINE_ERROR(OB_IN_STOP_STATE, -4114, -1, "08S01", "In STOP state");
DEFINE_ERROR(OB_UPS_MASTER_EXISTS, -4115, -1, "HY000", "Master UpdateServer already exists");
DEFINE_ERROR(OB_LOG_NOT_CLEAR, -4116, -1, "42000", "Log not clear");
DEFINE_ERROR(OB_FILE_ALREADY_EXIST, -4117, ER_FILE_EXISTS_ERROR, "58000", "File already exist");
DEFINE_ERROR(OB_UNKNOWN_PACKET, -4118, ER_UNKNOWN_COM_ERROR, "HY001", "Unknown packet");
DEFINE_ERROR(OB_RPC_PACKET_TOO_LONG, -4119, -1, "08000", "RPC packet to send too long");
DEFINE_ERROR(OB_LOG_TOO_LARGE, -4120, -1, "HY000", "Log too large");
DEFINE_ERROR(OB_RPC_SEND_ERROR, -4121, -1, "08000", "RPC send error");
DEFINE_ERROR(OB_RPC_POST_ERROR, -4122, -1, "08000", "RPC post error");
DEFINE_ERROR(OB_LIBEASY_ERROR, -4123, -1, "08000", "Libeasy error");
DEFINE_ERROR(OB_CONNECT_ERROR, -4124, -1, "HY000", "Connect error");
DEFINE_ERROR(OB_NOT_FREE, -4125, -1, "HY000", "Not free");
DEFINE_ERROR(OB_INIT_SQL_CONTEXT_ERROR, -4126, -1, "HY000", "Init SQL context error");
DEFINE_ERROR(OB_SKIP_INVALID_ROW, -4127, -1, "42000", "Skip invalid row");
DEFINE_ERROR(OB_RPC_PACKET_INVALID, -4128, -1, "HY000", "RPC packet is invalid");
DEFINE_ERROR(OB_NO_TABLET, -4133, -1, "HY000", "No tablets");
DEFINE_ERROR(OB_SNAPSHOT_DISCARDED, -4138, -1, "HY000", "Request to read too old versioned data");
DEFINE_ERROR(OB_DATA_NOT_UPTODATE, -4139, -1, "HY000", "State is stale");

```

```

DEFINE_ERROR(OB_ROW_MODIFIED, -4142, -1, "HY000", "Row modified");
DEFINE_ERROR(OB_VERSION_NOT_MATCH, -4143, -1, "42000", "Version not match");
DEFINE_ERROR(OB_BAD_ADDRESS, -4144, -1, "42000", "Bad address");
DEFINE_ERROR(OB_ENQUEUE_FAILED, -4146, -1, "HY000", "Enqueue error");
DEFINE_ERROR(OB_INVALID_CONFIG, -4147, -1, "HY000", "Invalid config");
DEFINE_ERROR(OB_STMT_EXPIRED, -4149, -1, "HY000", "Expired statement");
DEFINE_ERROR(OB_ERR_MIN_VALUE, -4150, -1, "42000", "Min value");
DEFINE_ERROR(OB_ERR_MAX_VALUE, -4151, -1, "42000", "Max value");
DEFINE_ERROR_EXT(OB_ERR_NULL_VALUE, -4152, -1, "42000", "Null value", "%s");
DEFINE_ERROR(OB_RESOURCE_OUT, -4153, ER_OUT_OF_RESOURCES, "53000", "Out of resource");
DEFINE_ERROR(OB_ERR_SQL_CLIENT, -4154, -1, "HY000", "Internal SQL client error");
DEFINE_ERROR(OB_META_TABLE_WITHOUT_USE_TABLE, -4155, -1, "HY000", "Meta table without use table");
DEFINE_ERROR(OB_DISCARD_PACKET, -4156, -1, "HY000", "Discard packet");
DEFINE_ERROR_EXT(OB_OPERATE_OVERFLOW, -4157, ER_DATA_OUT_OF_RANGE, "22003", "value is out of range", "%s value is out of range in '%s'");
DEFINE_ERROR_EXT(OB_INVALID_DATE_FORMAT, -4158, ER_TRUNCATED_WRONG_VALUE, "22007", "Incorrect value", "%s=%d must between %d and %d");
DEFINE_ERROR(OB_POOL_REGISTERED_FAILED, -4159, -1, "HY000", "register pool failed");
DEFINE_ERROR(OB_POOL_UNREGISTERED_FAILED, -4160, -1, "HY000", "unregister pool failed");
DEFINE_ERROR(OB_INVALID_ARGUMENT_NUM, -4161, -1, "42000", "Invalid argument num");
DEFINE_ERROR(OB_LEASE_NOT_ENOUGH, -4162, -1, "HY000", "reserved lease not enough");
DEFINE_ERROR(OB_LEASE_NOT_MATCH, -4163, -1, "HY000", "ups lease not match with rs");
DEFINE_ERROR(OB_UPS_SWITCH_NOT_HAPPEN, -4164, -1, "HY000", "ups switch not happen");
DEFINE_ERROR(OB_EMPTY_RESULT, -4165, -1, "HY000", "Empty result");
DEFINE_ERROR(OB_CACHE_NOT_HIT, -4166, -1, "HY000", "Cache not hit");
DEFINE_ERROR(OB_NESTED_LOOP_NOT_SUPPORT, -4167, -1, "HY000", "Nested loop not support");
DEFINE_ERROR(OB_LOG_INVALID_MOD_ID, -4168, -1, "HY000", "Invalid log module id");
DEFINE_ERROR_EXT(OB_LOG_MODULE_UNKNOWN, -4169, -1, "HY000", "Unknown module name", "Unknown module name. Invalid Setting: '%s'. Syntax: parMod.subMod:level, parMod.subMod:level");
DEFINE_ERROR_EXT(OB_LOG_LEVEL_INVALID, -4170, -1, "HY000", "Invalid level", "Invalid level. Invalid setting: '%s'. Syntax: parMod.subMod:level, parMod.subMod:level");
DEFINE_ERROR_EXT(OB_LOG_PARSER_SYNTAX_ERR, -4171, -1, "HY000", "Syntax to set log_level error", "Syntax to set log_level error. Invalid setting: '%s'. Syntax: parMod.subMod:level, parMod.subMod:level");
DEFINE_ERROR(OB_INDEX_OUT_OF_RANGE, -4172, -1, "HY000", "Index out of range");
DEFINE_ERROR(OB_INT_UNDERFLOW, -4173, -1, "HY000", "Int underflow");
DEFINE_ERROR_EXT(OB_UNKNOWN_CONNECTION, -4174, ER_NO_SUCH_THREAD, "HY000", "Unknown thread id", "Unknown thread id: %lu");
DEFINE_ERROR(OB_ERROR_OUT_OF_RANGE, -4175, -1, "42000", "Out of range");
DEFINE_ERROR(OB_CACHE_SHRINK_FAILED, -4176, -1, "HY001", "shrink cache failed, no available cache");
DEFINE_ERROR(OB_OLD_SCHEMA_VERSION, -4177, -1, "42000", "Schema version too old");
DEFINE_ERROR(OB_RELEASE_SCHEMA_ERROR, -4178, -1, "HY000", "Release schema error");
DEFINE_ERROR_EXT(OB_OP_NOT_ALLOW, -4179, -1, "HY000", "Operation not allowed now", "%s not allowed");
DEFINE_ERROR(OB_NO_EMPTY_ENTRY, -4180, -1, "HY000", "No empty entry");
DEFINE_ERROR(OB_ERR_ALREADY_EXISTS, -4181, -1, "42S01", "Already exist");
DEFINE_ERROR(OB_SEARCH_NOT_FOUND, -4182, -1, "HY000", "Value not found");
DEFINE_ERROR(OB_BEYOND_THE_RANGE, -4183, -1, "HY000", "Key out of range");
DEFINE_ERROR(OB_CS_OUT_OF_DISK_SPACE, -4184, -1, "53100", "ChunkServer out of disk space");

```

```

DEFINE_ERROR(OB_CS_OUT_OF_DISK_SPACE, -4184, -1, "HY000", "ChunkServer out of disk space");
DEFINE_ERROR(OB_COLUMN_GROUP_NOT_FOUND, -4185, -1, "HY000", "Column group not found");
DEFINE_ERROR(OB_CS_COMPRESS_LIB_ERROR, -4186, -1, "HY000", "ChunkServer failed to get compress library");
DEFINE_ERROR(OB_ITEM_NOT_MATCH, -4187, -1, "HY000", "Item not match");
DEFINE_ERROR(OB_SCHEDULER_TASK_CNT_MISMATCH, -4188, -1, "HY000", "Running task cnt and unfinished task cnt not consistent");
DEFINE_ERROR(OB_HASH_EXIST, -4200, -1, "HY000", "hash map/set entry exist");
DEFINE_ERROR(OB_HASH_NOT_EXIST, -4201, -1, "HY000", "hash map/set entry not exist");
DEFINE_ERROR(OB_HASH_GET_TIMEOUT, -4204, -1, "HY000", "hash map/set get timeout");
DEFINE_ERROR(OB_HASH_PLACEMENT_RETRY, -4205, -1, "HY000", "hash map/set retry");
DEFINE_ERROR(OB_HASH_FULL, -4206, -1, "HY000", "hash map/set full");
DEFINE_ERROR(OB_PACKET_PROCESSED, -4207, -1, "HY000", "packet processed");
DEFINE_ERROR(OB_WAIT_NEXT_TIMEOUT, -4208, -1, "HY000", "wait next packet timeout");
DEFINE_ERROR(OB_LEADER_NOT_EXIST, -4209, -1, "HY000", "partition has not leader");
DEFINE_ERROR(OB_PREPARE_MAJOR_FREEZE_FAILED, -4210, -1, "HY000", "prepare major freeze failed");
DEFINE_ERROR(OB_COMMIT_MAJOR_FREEZE_FAILED, -4211, -1, "HY000", "commit major freeze failed");
DEFINE_ERROR(OB_ABORT_MAJOR_FREEZE_FAILED, -4212, -1, "HY000", "abort major freeze failed");
DEFINE_ERROR(OB_MAJOR_FREEZE_NOT_FINISHED, -4213, -1, "HY000", "last major freeze not finish");
DEFINE_ERROR(OB_PARTITION_NOT_LEADER, -4214, -1, "HY000", "partition is not leader partition");
DEFINE_ERROR(OB_WAIT_MAJOR_FREEZE_RESPONSE_TIMEOUT, -4215, -1, "HY000", "wait major freeze response timeout");
;
DEFINE_ERROR(OB_CURL_ERROR, -4216, -1, "HY000", "curl error");
DEFINE_ERROR_EXT(OB_MAJOR_FREEZE_NOT_ALLOW, -4217, -1, "HY000", "Major freeze not allowed now", "%s");
DEFINE_ERROR(OB_PREPARE_FREEZE_FAILED, -4218, -1, "HY000", "prepare freeze failed");
DEFINE_ERROR_EXT(OB_INVALID_DATE_VALUE, -4219, ER_TRUNCATED_WRONG_VALUE, "22007", "Incorrect value", "Incorrect datetime value: '%s' for column '%s' at row %d");
DEFINE_ERROR(OB_INACTIVE_SQL_CLIENT, -4220, -1, "HY000", "Inactive sql client, only read allowed");
DEFINE_ERROR(OB_INACTIVE_RPC_PROXY, -4221, -1, "HY000", "Inactive rpc proxy, can not send RPC request");
DEFINE_ERROR(OB_INTERVAL_WITH_MONTH, -4222, -1, "42000", "Interval with year or month can not be converted to microseconds");
DEFINE_ERROR(OB_TOO_MANY_DATETIME_PARTS, -4223, -1, "42000", "Interval has too many datetime parts");
DEFINE_ERROR_EXT(OB_DATA_OUT_OF_RANGE, -4224, ER_WARN_DATA_OUT_OF_RANGE, "22003", "Out of range value for column", "Out of range value for column '%.*s' at row %ld");
DEFINE_ERROR(OB_PARTITION_NOT_EXIST, -4225, -1, "HY000", "Partition entry not exists");
DEFINE_ERROR_EXT(OB_ERR_TRUNCATED_WRONG_VALUE_FOR_FIELD, -4226, ER_TRUNCATED_WRONG_VALUE_FOR_FIELD, "HY000", "Incorrect integer value", "Incorrect integer value: '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_NO_DEFAULT_FOR_FIELD, -4227, ER_NO_DEFAULT_FOR_FIELD, "HY000", "Field doesn't have a default value", "Field '%s' doesn't have a default value");
DEFINE_ERROR_EXT(OB_ERR_FIELD_SPECIFIED_TWICE, -4228, ER_FIELD_SPECIFIED_TWICE, "42000", "Column specified twice", "Column '%s' specified twice");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_TABLE_COMMENT, -4229, ER_TOO_LONG_TABLE_COMMENT, "HY000", "Comment for table is too long", "Comment for table is too long (max = %ld)");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_FIELD_COMMENT, -4230, ER_TOO_LONG_FIELD_COMMENT, "HY000", "Comment for field is too long", "Comment for field is too long (max = %ld)");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_INDEX_COMMENT, -4231, ER_TOO_LONG_INDEX_COMMENT, "HY000", "Comment for index is too long", "Comment for index is too long (max = %ld)");
DEFINE_ERROR(OB_NOT_FOLLOWER, -4232, -1, "HY000", "The observer or zone is not a follower");

```

```

DEFINE_ERROR(OB_ERR_OUT_OF_LOWER_BOUND, -4233, -1, "HY000", "smaller than container lower bound");
DEFINE_ERROR(OB_ERR_OUT_OF_UPPER_BOUND, -4234, -1, "HY000", "bigger than container upper bound");
DEFINE_ERROR_EXT(OB_BAD_NULL_ERROR, -4235, ER_BAD_NULL_ERROR, "23000", "Column cannot be null", "Column '%.s'
cannot be null");
DEFINE_ERROR(OB_OBCONFIG_RETURN_ERROR, -4236, -1, "HY000", "ObConfig return error code");
DEFINE_ERROR(OB_OBCONFIG_APPNAME_MISMATCH, -4237, -1, "HY000", "Appname mismatch with obconfig result");
DEFINE_ERROR(OB_ERR_VIEW_SELECT_DERIVED, -4238, ER_VIEW_SELECT_DERIVED, "HY000", "View's SELECT contains a sub
query in the FROM clause");
DEFINE_ERROR(OB_CANT_MJ_PATH, -4239, -1, "HY000", "Can not use merge-join to join the tables without join conditions
");
DEFINE_ERROR(OB_ERR_NO_JOIN_ORDER_GENERATED, -4240, -1, "HY000", "No join order generated");
DEFINE_ERROR(OB_ERR_NO_PATH_GENERATED, -4241, -1, "HY000", "No join path generated");
DEFINE_ERROR(OB_ERR_WAIT_REMOTE_SCHEMA_REFRESH, -4242, -1, "HY000", "Schema error");
DEFINE_ERROR(OB_FILE_NOT_OPENED, -4243, -1, "HY000", "file not opened");
DEFINE_ERROR(OB_TIMER_TASK_HAS_SCHEDULED, -4244, -1, "HY000", "Timer task has been scheduled");
DEFINE_ERROR(OB_TIMER_TASK_HAS_NOT_SCHEDULED, -4245, -1, "HY000", "Timer task has not been scheduled");
DEFINE_ERROR(OB_PARSE_DEBUG_SYNC_ERROR, -4246, -1, "HY000", "parse debug sync string error");
DEFINE_ERROR(OB_UNKNOWN_DEBUG_SYNC_POINT, -4247, -1, "HY000", "unknown debug sync point");
DEFINE_ERROR(OB_ERR_INTERRUPTED, -4248, -1, "HY000", "task is interrupted while running");
DEFINE_ERROR(OB_ERR_DATA_TRUNCATED, -4249, WARN_DATA_TRUNCATED, "01000", "Data truncated for argument");
// used by modules in partition service only, and not returned to client
DEFINE_ERROR(OB_NOT_RUNNING, -4250, -1, "HY000", "module is not running");
DEFINE_ERROR(OB_INVALID_PARTITION, -4251, -1, "HY000", "partition not valid");
DEFINE_ERROR(OB_ERR_TIMEOUT_TRUNCATED, -4252, WARN_DATA_TRUNCATED, "01000", "Timeout value truncated to 10
2 years");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_TENANT_COMMENT, -4253, -1, "HY000", "Comment for tenant is too long", "Comm
ent for tenant is too long (max = %ld)");
DEFINE_ERROR(OB_ERR_NET_PACKET_TOO_LARGE, -4254, ER_NET_PACKET_TOO_LARGE, "08501", "Got a packet bigger tha
n \max_allowed_packet\ bytes");
DEFINE_ERROR(OB_TRACE_DESC_NOT_EXIST, -4255, -1, "HY000", "trace log title or key not exist describe");
DEFINE_ERROR_EXT(OB_ERR_NO_DEFAULT, -4256, ER_NO_DEFAULT, "42000", "Variable doesn't have a default value", "Var
iable '%.s' doesn't have a default value");
DEFINE_ERROR(OB_ERR_COMPRESS_DECOMPRESS_DATA, -4257, -1, "HY000", "compress data or decompress data failed")
;
DEFINE_ERROR_EXT(OB_ERR_INCORRECT_STRING_VALUE, -4258, ER_TRUNCATED_WRONG_VALUE_FOR_FIELD, "HY000", "Inc
orrect string value", "Incorrect string value for column '%.s' at row %lld");
DEFINE_ERROR_EXT(OB_ERR_DISTRIBUTED_NOT_SUPPORTED, -4259, ER_NOT_SUPPORTED_YET, "0A000", "Not supported f
eature or function", "%s not supported");
DEFINE_ERROR(OB_IS_CHANGING_LEADER, -4260, -1, "HY000", "the partition is changing leader");
DEFINE_ERROR(OB_DATETIME_FUNCTION_OVERFLOW, -4261, ER_DATETIME_FUNCTION_OVERFLOW, "22008", "Datetime ove
rflow");
DEFINE_ERROR_EXT(OB_ERR_DOUBLE_TRUNCATED, -4262, ER_TRUNCATED_WRONG_VALUE, "01000", "Truncated incorrect
DOUBLE value", "Truncated incorrect DOUBLE value: '%.s'");
DEFINE_ERROR_EXT(OB_MINOR_FREEZE_NOT_ALLOW, -4263, -1, "HY000", "Minor freeze not allowed now", "%s");
DEFINE_ERROR(OB_LOG_OUTOF_DISK_SPACE, -4264, -1, "HY000", "Log out of disk space");
DEFINE_ERROR(OB_RPC_CONNECT_ERROR, -4265, -1, "HY000", "Rpc connect error");
DEFINE_ERROR(OB_MINOR_MERGE_NOT_ALLOW, -4266, -1, "HY000", "minor merge not allow");

```

```

DEFINE_ERROR(OB_CACHE_INVALID, -4267, -1, "HY000", "Cache invalid");
DEFINE_ERROR(OB_REACH_SERVER_DATA_COPY_IN_CONCURRENCY_LIMIT, -4268, -1, "HY000", "reach server data copy in c
oncurrency");
DEFINE_ERROR(OB_WORKING_PARTITION_EXIST, -4269, -1, "HY000", "Working partition entry already exists");
DEFINE_ERROR(OB_WORKING_PARTITION_NOT_EXIST, -4270, -1, "HY000", "Working partition entry does not exists");
DEFINE_ERROR(OB_LIBEASY_REACH_MEM_LIMIT, -4271, -1, "HY000", "LIBEASY reach memory limit");
DEFINE_ERROR_EXT(OB_MISS_ARGUMENT, -4272, ER_WRONG_ARGUMENTS, "HY000", "Miss argument", "Miss argument for
%s");

////////////////////////////////////
//error code for root server & server management -4500 ---- -5000
////////////////////////////////////
DEFINE_ERROR(OB_IMPORT_NOT_IN_SERVER, -4505, -1, "HY000", "Import not in service");
DEFINE_ERROR(OB_CONVERT_ERROR, -4507, -1, "42000", "Convert error");
DEFINE_ERROR(OB_BYPASS_TIMEOUT, -4510, -1, "HY000", "Bypass timeout");
DEFINE_ERROR(OB_RS_STATE_NOT_ALLOW, -4512, -1, "HY000", "RootServer state error");
DEFINE_ERROR(OB_NO_REPLICA_VALID, -4515, -1, "HY000", "No replica is valid");
DEFINE_ERROR(OB_NO_NEED_UPDATE, -4517, -1, "HY000", "No need to update");
DEFINE_ERROR(OB_CACHE_TIMEOUT, -4518, -1, "HY000", "Cache timeout");
DEFINE_ERROR(OB_ITER_STOP, -4519, -1, "HY000", "Iteration was stopped");
DEFINE_ERROR(OB_ZONE_ALREADY_MASTER, -4523, -1, "HY000", "The zone is the master already");
DEFINE_ERROR(OB_IP_PORT_IS_NOT_SLAVE_ZONE, -4524, -1, "HY000", "Not slave zone");
DEFINE_ERROR(OB_ZONE_IS_NOT_SLAVE, -4525, -1, "HY000", "Not slave zone");
DEFINE_ERROR(OB_ZONE_IS_NOT_MASTER, -4526, -1, "HY000", "Not master zone");
DEFINE_ERROR(OB_CONFIG_NOT_SYNC, -4527, -1, "F0000", "Configuration not sync");
DEFINE_ERROR(OB_IP_PORT_IS_NOT_ZONE, -4528, -1, "42000", "Not a zone address");
DEFINE_ERROR(OB_MASTER_ZONE_NOT_EXIST, -4529, -1, "HY000", "Master zone not exist");
DEFINE_ERROR_EXT(OB_ZONE_INFO_NOT_EXIST, -4530, -1, "HY000", "Zone info not exist", "Zone info '%s' not exist");
DEFINE_ERROR(OB_GET_ZONE_MASTER_UPS_FAILED, -4531, -1, "HY000", "Failed to get master UpdateServer");
DEFINE_ERROR(OB_MULTIPLE_MASTER_ZONES_EXIST, -4532, -1, "HY000", "Multiple master zones");
DEFINE_ERROR(OB_INDEXING_ZONE_INVALID, -4533, -1, "HY000", "indexing zone is not exist anymore or not active");
DEFINE_ERROR(OB_ROOT_TABLE_RANGE_NOT_EXIST, -4537, -1, "HY000", "Tablet range not exist");
DEFINE_ERROR(OB_ROOT_MIGRATE_CONCURRENCY_FULL, -4538, -1, "HY000", "Migrate concurrency full");
DEFINE_ERROR(OB_ROOT_MIGRATE_INFO_NOT_FOUND, -4539, -1, "HY000", "Migrate info not found");
DEFINE_ERROR(OB_NOT_DATA_LOAD_TABLE, -4540, -1, "HY000", "No data to load");
DEFINE_ERROR(OB_DATA_LOAD_TABLE_DUPLICATED, -4541, -1, "HY000", "Duplicated table data to load");
DEFINE_ERROR(OB_ROOT_TABLE_ID_EXIST, -4542, -1, "HY000", "Table ID exist");
DEFINE_ERROR(OB_INDEX_TIMEOUT, -4543, -1, "HY000", "Building index timeout");
DEFINE_ERROR(OB_ROOT_NOT_INTEGRATED, -4544, -1, "42000", "Root not integrated");
DEFINE_ERROR(OB_INDEX_INELIGIBLE, -4545, -1, "HY000", "index data not unique");
DEFINE_ERROR(OB_REBALANCE_EXEC_TIMEOUT, -4546, -1, "HY000", "execute replication or migration task timeout");
DEFINE_ERROR(OB_MERGE_NOT_STARTED, -4547, -1, "HY000", "global merge not started");
DEFINE_ERROR(OB_MERGE_ALREADY_STARTED, -4548, -1, "HY000", "merge already started");
DEFINE_ERROR(OB_ROOTSERVICE_EXIST, -4549, -1, "HY000", "rootservice already exist");
DEFINE_ERROR(OB_RS_SHUTDOWN, -4550, -1, "HY000", "rootservice is shutdown");
DEFINE_ERROR(OB_SERVER_MIGRATE_IN_DENIED, -4551, -1, "HY000", "server migrate in denied");
DEFINE_ERROR(OB_REBALANCE_TASK_CANT_EXEC, -4552, -1, "HY000", "rebalance task can not executing now");

```

```

DEFINE_ERROR(OB_REBALANCE_TASK_CANNOT_EXEC, -4552, -1, "HY000", "rebalance task can not executing now");
DEFINE_ERROR(OB_PARTITION_CNT_REACH_ROOTSERVER_LIMIT, -4553, -1, "HY000", "rootserver can not hold more partition");
DEFINE_ERROR(OB_DATA_SOURCE_NOT_EXIST, -4600, -1, "HY000", "Data source not exist");
DEFINE_ERROR(OB_DATA_SOURCE_TABLE_NOT_EXIST, -4601, -1, "HY000", "Data source table not exist");
DEFINE_ERROR(OB_DATA_SOURCE_RANGE_NOT_EXIST, -4602, -1, "HY000", "Data source range not exist");
DEFINE_ERROR(OB_DATA_SOURCE_DATA_NOT_EXIST, -4603, -1, "HY000", "Data source data not exist");
DEFINE_ERROR(OB_DATA_SOURCE_SYS_ERROR, -4604, -1, "HY000", "Data source sys error");
DEFINE_ERROR(OB_DATA_SOURCE_TIMEOUT, -4605, -1, "HY000", "Data source timeout");
DEFINE_ERROR(OB_DATA_SOURCE_CONCURRENCY_FULL, -4606, -1, "53000", "Data source concurrency full");
DEFINE_ERROR(OB_DATA_SOURCE_WRONG_URI_FORMAT, -4607, -1, "42000", "Data source wrong URI format");

DEFINE_ERROR(OB_SSTABLE_VERSION_UNEQUAL, -4608, -1, "42000", "SSTable version not equal");
DEFINE_ERROR(OB_UPS_RENEW_LEASE_NOT_ALLOWED, -4609, -1, "HY000", "ups should not renew its lease");
DEFINE_ERROR(OB_UPS_COUNT_OVER_LIMIT, -4610, -1, "HY000", "ups count over limit");
DEFINE_ERROR(OB_NO_UPS_MAJORITY, -4611, -1, "HY000", "ups not form a majority");
DEFINE_ERROR(OB_INDEX_COUNT_REACH_THE_LIMIT, -4613, -1, "HY000", "created index tables count has reach the limit: 128");
DEFINE_ERROR(OB_TASK_EXPIRED, -4614, -1, "HY000", "task expired");
DEFINE_ERROR(OB_TABLEGROUP_NOT_EMPTY, -4615, -1, "HY000", "tablegroup is not empty");
DEFINE_ERROR(OB_INVALID_SERVER_STATUS, -4620, -1, "HY000", "server status is not valid");
DEFINE_ERROR(OB_WAIT_ELEC_LEADER_TIMEOUT, -4621, -1, "HY000", "wait elect partition leader timeout");
DEFINE_ERROR(OB_WAIT_ALL_RS_ONLINE_TIMEOUT, -4622, -1, "HY000", "wait all rs online timeout");
DEFINE_ERROR(OB_ALL_REPLICAS_ON_MERGE_ZONE, -4623, -1, "HY000", "all replicas of partition group are on zones to merge");
DEFINE_ERROR(OB_MACHINE_RESOURCE_NOT_ENOUGH, -4624, -1, "HY000", "machine resource is not enough to hold a new unit");
DEFINE_ERROR(OB_NOT_SERVER_CAN_HOLD_SOFTLY, -4625, -1, "HY000", "not server can hold the unit and not over soft limit");
DEFINE_ERROR_EXT(OB_RESOURCE_POOL_ALREADY_GRANTED, -4626, -1, "HY000", "resource pool has already been granted to a tenant", "resource pool '%s\' has already been granted to a tenant");
DEFINE_ERROR(OB_SERVER_ALREADY_DELETED, -4628, -1, "HY000", "server has already been deleted");
DEFINE_ERROR(OB_SERVER_NOT_DELETING, -4629, -1, "HY000", "server is not in deleting status");
DEFINE_ERROR(OB_SERVER_NOT_IN_WHITE_LIST, -4630, -1, "HY000", "server not in server white list");
DEFINE_ERROR(OB_SERVER_ZONE_NOT_MATCH, -4631, -1, "HY000", "server zone not match");
DEFINE_ERROR(OB_OVER_ZONE_NUM_LIMIT, -4632, -1, "HY000", "zone num has reach max zone num");
DEFINE_ERROR(OB_ZONE_STATUS_NOT_MATCH, -4633, -1, "HY000", "zone status not match");
DEFINE_ERROR_EXT(OB_RESOURCE_UNIT_IS_REFERENCED, -4634, -1, "HY000", "resource unit is referenced by resource pool", "resource unit '%s\' is referenced by some resource pool");
DEFINE_ERROR(OB_DIFFERENT_PRIMARY_ZONE, -4636, -1, "HY000", "table schema primary zone different with other table in same tablegroup");
DEFINE_ERROR(OB_SERVER_NOT_ACTIVE, -4637, -1, "HY000", "server is not active");
DEFINE_ERROR(OB_RS_NOT_MASTER, -4638, -1, "HY000", "The RootServer is not the master");
DEFINE_ERROR(OB_CANDIDATE_LIST_ERROR, -4639, -1, "HY000", "The candidate list is invalid");
DEFINE_ERROR(OB_PARTITION_ZONE_DUPLICATED, -4640, -1, "HY000", "The chosen partition servers belong to same zone.");
DEFINE_ERROR_EXT(OB_ZONE_DUPLICATED, -4641, -1, "HY000", "Duplicated zone in zone list", "Duplicated zone '%s\' in

```

```

zone list %s");
DEFINE_ERROR(OB_NOT_ALL_ZONE_ACTIVE, -4642, -1, "HY000", "Not all zone in zone list are active");
DEFINE_ERROR_EXT(OB_PRIMARY_ZONE_NOT_IN_ZONE_LIST, -4643, -1, "HY000", "primary zone not in zone list", "primary
zone \"%s\" not in zone list %s");
DEFINE_ERROR(OB_REPLICA_NUM_NOT_MATCH, -4644, -1, "HY000", "replica num not same with zone count");
DEFINE_ERROR_EXT(OB_ZONE_LIST_POOL_LIST_NOT_MATCH, -4645, -1, "HY000", "zone list not a subset of resource pool
list", "zone list %s not a subset of resource pool zone list %s");
DEFINE_ERROR_EXT(OB_INVALID_TENANT_NAME, -4646, -1, "HY000", "tenant name is too long", "tenant name \"%s\" over
max_tenant_name_length %ld");
DEFINE_ERROR(OB_EMPTY_RESOURCE_POOL_LIST, -4647, -1, "HY000", "resource pool list is empty");
DEFINE_ERROR_EXT(OB_RESOURCE_UNIT_NOT_EXIST, -4648, -1, "HY000", "resource unit not exist", "resource unit \"%s\" n
ot exist");
DEFINE_ERROR_EXT(OB_RESOURCE_UNIT_EXIST, -4649, -1, "HY000", "resource unit already exist", "resource unit \"%s\" alr
eady exist");
DEFINE_ERROR_EXT(OB_RESOURCE_POOL_NOT_EXIST, -4650, -1, "HY000", "resource pool not exist", "resource pool \"%s\"
not exist");
DEFINE_ERROR_EXT(OB_RESOURCE_POOL_EXIST, -4651, -1, "HY000", "resource pool already exist", "resource pool \"%s\" e
xist");
DEFINE_ERROR(OB_WAIT_LEADER_SWITCH_TIMEOUT, -4652, -1, "HY000", "wait leader switch timeout");
DEFINE_ERROR(OB_LOCATION_NOT_EXIST, -4653, -1, "HY000", "location not exist");
DEFINE_ERROR(OB_LOCATION_LEADER_NOT_EXIST, -4654, -1, "HY000", "location leader not exist");
DEFINE_ERROR(OB_ZONE_NOT_ACTIVE, -4655, -1, "HY000", "zone not active");
DEFINE_ERROR(OB_UNIT_NUM_OVER_SERVER_COUNT, -4656, -1, "HY000", "resource pool unit num is bigger than zone ser
ver count");
DEFINE_ERROR_EXT(OB_POOL_SERVER_INTERSECT, -4657, -1, "HY000", "resource pool list unit server intersect", "resourc
e pool list %s unit servers intersect");
DEFINE_ERROR_EXT(OB_NOT_SINGLE_RESOURCE_POOL, -4658, -1, "HY000", "create tenant only support single resource p
ool now", "create tenant only support single resource pool now, but pool list is %s");
DEFINE_ERROR_EXT(OB_INVALID_RESOURCE_UNIT, -4659, -1, "HY000", "invalid resource unit", "invalid resource unit, %s\
s min value is %s");
DEFINE_ERROR_EXT(OB_STOP_SERVER_IN_MULTIPLE_ZONES, -4660, -1, "HY000", "Can not stop server in multiple zones", "
Can not stop server in multiple zones, there are already servers stopped in zone:%s");
DEFINE_ERROR(OB_SESSION_ENTRY_EXIST, -4661, -1, "HY000", "Session already exist");
DEFINE_ERROR_EXT(OB_GOT_SIGNAL_ABORTING, -4662, ER_GOT_SIGNAL, "01000", "Got signal. Aborting!", "%s: Got signal
%d. Aborting!");
DEFINE_ERROR(OB_SERVER_NOT_ALIVE, -4663, -1, "HY000", "server is not alive");
DEFINE_ERROR(OB_GET_LOCATION_TIME_OUT, -4664, 4012, "HY000", "Timeout");
DEFINE_ERROR(OB_UNIT_IS_MIGRATING, -4665, -1, "HY000", "Unit is migrating, can not migrate again");
DEFINE_ERROR_EXT(OB_CLUSTER_NO_MATCH, -4666, -1, "HY000", "cluster name is not match", "cluster name is not match
to \"%s\"");
DEFINE_ERROR(OB_CHECK_ZONE_MERGE_ORDER, -4667, -1, "HY000", "Please check new zone in zone_merge_order. You c
an show parameters like 'zone_merge_order'");
DEFINE_ERROR_EXT(OB_ERR_ZONE_NOT_EMPTY, -4668, -1, "HY000", "zone not empty", "The zone is not empty and can no
t be deleted. You should delete the servers of the zone. There are %ld servers alive and %ld not alive.");
////////////////////////////////////
// SQL & Schema specific error code, -5000 ~ -6000
////////////////////////////////////

```

```

DEFINE_ERROR(OB_ERR_PARSER_INIT, -5000, ER_PARSE_ERROR, "0B000", "Failed to init SQL parser");
DEFINE_ERROR_EXT(OB_ERR_PARSE_SQL, -5001, ER_PARSE_ERROR, "42000", "Parse error", "%s near '%s\' at line %d");
DEFINE_ERROR(OB_ERR_RESOLVE_SQL, -5002, -1, "HY000", "Resolve error");
DEFINE_ERROR(OB_ERR_GEN_PLAN, -5003, -1, "HY000", "Generate plan error");
DEFINE_ERROR(OB_ERR_PARSER_SYNTAX, -5006, ER_SYNTAX_ERROR, "42000", "You have an error in your SQL syntax; che
ck the manual that corresponds to your MySQL server version for the right syntax to use");
DEFINE_ERROR(OB_ERR_COLUMN_SIZE, -5007, ER_WRONG_NUMBER_OF_COLUMNS_IN_SELECT, "21000", "The used SELECT
statements have a different number of columns");
// xiyu@TODO: will be replaced by OB_NON_UNIQ_ERROR
DEFINE_ERROR_EXT(OB_ERR_COLUMN_DUPLICATE, -5008, ER_DUP_FIELDNAME, "42S21", "Duplicate column name", "Duplica
te column name '%s'");
DEFINE_ERROR(OB_ERR_OPERATOR_UNKNOWN, -5010, -1, "21000", "Unknown operator");
DEFINE_ERROR(OB_ERR_STAR_DUPLICATE, -5011, -1, "42000", "Duplicated star");
DEFINE_ERROR_EXT(OB_ERR_ILLEGAL_ID, -5012, -1, "HY000", "Illegal ID", "%s");
DEFINE_ERROR(OB_ERR_ILLEGAL_VALUE, -5014, -1, "HY000", "Illegal value");
DEFINE_ERROR(OB_ERR_COLUMN_AMBIGUOUS, -5015, ER_AMBIGUOUS_FIELD_TERM, "42000", "Ambiguous column");
DEFINE_ERROR(OB_ERR_LOGICAL_PLAN_FAILED, -5016, -1, "HY000", "Generate logical plan error");
DEFINE_ERROR(OB_ERR_SCHEMA_UNSET, -5017, -1, "HY000", "Schema not set");
DEFINE_ERROR(OB_ERR_ILLEGAL_NAME, -5018, -1, "42000", "Illegal name");
DEFINE_ERROR_EXT(OB_ERR_TABLE_EXIST, -5020, ER_TABLE_EXISTS_ERROR, "42S01", "Table already exists", "Table '%s'
already exists");
DEFINE_ERROR_EXT(OB_TABLE_NOT_EXIST, -5019, ER_NO_SUCH_TABLE, "42S02", "Table doesn't exist", "Table '%s.%s' d
oesn't exist");
DEFINE_ERROR(OB_ERR_EXPR_UNKNOWN, -5022, -1, "42000", "Unknown expression");
DEFINE_ERROR_EXT(OB_ERR_ILLEGAL_TYPE, -5023, -1, "S1004", "Illegal type", "unsupport MySQL type %d. Maybe you sho
uld use java.sql.Timestamp instead of java.util.Date.");
DEFINE_ERROR_EXT(OB_ERR_KEY_NAME_DUPLICATE, -5025, ER_DUP_KEYNAME, "42000", "Duplicated key name", "Duplicate
key name '%s'");
DEFINE_ERROR_EXT(OB_ERR_PRIMARY_KEY_DUPLICATE, -5024, ER_DUP_ENTRY, "23000", "Duplicated primary key", "Duplic
ate entry '%s' for key '%s'");
DEFINE_ERROR(OB_ERR_CREATETIME_DUPLICATE, -5026, -1, "42000", "Duplicated createtime");
DEFINE_ERROR(OB_ERR_MODIFYTIME_DUPLICATE, -5027, -1, "42000", "Duplicated modifytime");
DEFINE_ERROR(OB_ERR_ILLEGAL_INDEX, -5028, ER_NO_SUCH_INDEX, "42S12", "Illegal index");
DEFINE_ERROR(OB_ERR_INVALID_SCHEMA, -5029, -1, "HY000", "Invalid schema");
DEFINE_ERROR(OB_ERR_INSERT_NULL_ROWKEY, -5030, ER_PRIMARY_CANT_HAVE_NULL, "42000", "Insert null rowkey");
DEFINE_ERROR(OB_ERR_COLUMN_NOT_FOUND, -5031, -1, "HY000", "Column not found");
DEFINE_ERROR(OB_ERR_DELETE_NULL_ROWKEY, -5032, -1, "23000", "Delete null rowkey");
DEFINE_ERROR(OB_ERR_USER_EMPTY, -5034, -1, "01007", "No user");
DEFINE_ERROR(OB_ERR_USER_NOT_EXIST, -5035, ER_NO_SUCH_USER, "01007", "User not exist");
DEFINE_ERROR_EXT(OB_ERR_NO_PRIVILEGE, -5036, ER_SPECIFIC_ACCESS_DENIED_ERROR, "42501", "Access denied", "Acces
s denied; you need (at least one of) the %s privilege(s) for this operation");
DEFINE_ERROR(OB_ERR_NO_AVAILABLE_PRIVILEGE_ENTRY, -5037, -1, "HY000", "No privilege entry");
DEFINE_ERROR(OB_ERR_WRONG_PASSWORD, -5038, ER_PASSWORD_NO_MATCH, "42000", "Incorrect password");
DEFINE_ERROR(OB_ERR_USER_IS_LOCKED, -5039, -1, "01007", "User locked");
DEFINE_ERROR(OB_ERR_UPDATE_ROWKEY_COLUMN, -5040, -1, "42000", "Can not update rowkey column");
DEFINE_ERROR(OB_ERR_UPDATE_JOIN_COLUMN, -5041, -1, "42000", "Can not update join column");

```

```

DEFINE_ERROR_EX1(OB_ERR_INVALID_COLUMN_NUM, -5042, ER_OPERAND_COLUMNS, "21000", "Invalid column number", "O
perand should contain %d column(s)");
DEFINE_ERROR_EXT(OB_ERR_PREPARE_STMT_NOT_FOUND, -5043, ER_UNKNOWN_STMT_HANDLER, "HY007", "Unknown pre
pared statement", "statement not prepared, stmt_id=%u");
DEFINE_ERROR_EXT(OB_ERR_SYS_VARIABLE_UNKNOWN, -5044, ER_UNKNOWN_SYSTEM_VARIABLE, "HY000", "Unknown sys
tem variable", "Unknown system variable '%.s'");
DEFINE_ERROR(OB_ERR_OLDER_PRIVILEGE_VERSION, -5046, -1, "HY000", "Older privilege version");
DEFINE_ERROR_EXT(OB_ERR_LACK_OF_ROWKEY_COL, -5047, ER_REQUIRES_PRIMARY_KEY, "42000", "No rowkey column spe
cified", "Primary key column(s) not specified in the WHERE clause");
DEFINE_ERROR(OB_ERR_USER_EXIST, -5050, -1, "42710", "User exists");
DEFINE_ERROR(OB_ERR_PASSWORD_EMPTY, -5051, -1, "HY000", "Empty password");
DEFINE_ERROR(OB_ERR_GRANT_PRIVILEGES_TO_CREATE_TABLE, -5052, -1, "42000", "Failed to grant privelege");
DEFINE_ERROR_EXT(OB_ERR_WRONG_DYNAMIC_PARAM, -5053, -1, "HY093", "Wrong dynamic parameters", "Incorrect argu
ments number to EXECUTE, need %ld arguments but give %ld");
DEFINE_ERROR_EXT(OB_ERR_PARAM_SIZE, -5054, ER_WRONG_PARAMCOUNT_TO_NATIVE_FCT, "42000", "Incorrect paramet
er count", "Incorrect parameter count in the call to native function '%.s'");
DEFINE_ERROR_EXT(OB_ERR_FUNCTION_UNKNOWN, -5055, ER_SP_DOES_NOT_EXIST, "42000", "FUNCTION does not exist", "
%s %s does not exist");
DEFINE_ERROR(OB_ERR_CREAT_MODIFY_TIME_COLUMN, -5056, -1, "23000", "CreateTime or ModifyTime column cannot be
modified");
DEFINE_ERROR(OB_ERR_MODIFY_PRIMARY_KEY, -5057, -1, "23000", "Primary key cannot be modified");
DEFINE_ERROR(OB_ERR_PARAM_DUPLICATE, -5058, -1, "42000", "Duplicated parameters");
DEFINE_ERROR(OB_ERR_TOO_MANY_SESSIONS, -5059, ER_TOO_MANY_USER_CONNECTIONS, "42000", "Too many sessions"
);
DEFINE_ERROR(OB_ERR_TOO_MANY_PS, -5061, -1, "54023", "Too many prepared statements");
DEFINE_ERROR(OB_ERR_HINT_UNKNOWN, -5063, -1, "42000", "Unknown hint");
DEFINE_ERROR(OB_ERR_WHEN_UNSATISFIED, -5064, -1, "23000", "When condition not satisfied");
DEFINE_ERROR(OB_ERR_QUERY_INTERRUPTED, -5065, ER_QUERY_INTERRUPTED, "70100", "Query execution was interrupte
d");
DEFINE_ERROR(OB_ERR_SESSION_INTERRUPTED, -5066, -1, "HY000", "Session interrupted");
DEFINE_ERROR(OB_ERR_UNKNOWN_SESSION_ID, -5067, -1, "HY000", "Unknown session ID");
DEFINE_ERROR(OB_ERR_PROTOCOL_NOT_RECOGNIZE, -5068, -1, "HY000", "Incorrect protocol");
DEFINE_ERROR(OB_ERR_WRITE_AUTH_ERROR, -5069, -1, "HY000", "Write auth packet error");
DEFINE_ERROR(OB_ERR_PARSE_JOIN_INFO, -5070, -1, "42000", "Wrong join info");
DEFINE_ERROR(OB_ERR_ALTER_INDEX_COLUMN, -5071, -1, "42000", "Cannot alter index column");
DEFINE_ERROR(OB_ERR_MODIFY_INDEX_TABLE, -5072, -1, "42000", "Cannot modify index table");
DEFINE_ERROR(OB_ERR_INDEX_UNAVAILABLE, -5073, ER_NO_SUCH_INDEX, "42000", "Index unavailable");
DEFINE_ERROR(OB_ERR_NOP_VALUE, -5074, -1, "23000", "NOP cannot be used here");
DEFINE_ERROR(OB_ERR_PS_TOO_MANY_PARAM, -5080, ER_PS_MANY_PARAM, "54000", "Prepared statement contains too
many placeholders");
DEFINE_ERROR(OB_ERR_READ_ONLY, -5081, -1, "25000", "The server is read only now");
DEFINE_ERROR_EXT(OB_ERR_INVALID_TYPE_FOR_OP, -5083, -1, "22000", "Invalid data type for the operation", "invalid obj
type for type promotion, left_type=%d right_type=%d");
DEFINE_ERROR(OB_ERR_CAST_VARCHAR_TO_BOOL, -5084, -1, "22000", "Can not cast varchar value to bool type");
DEFINE_ERROR(OB_ERR_CAST_VARCHAR_TO_NUMBER, -5085, -1, "22000", "Not a number Can not cast varchar value to nu
mber type");
DEFINE_ERROR(OB_ERR_CAST_VARCHAR_TO_TIME, -5086, -1, "22000", "Not timestamp Can not cast varchar value to times

```

```

tamp type");
DEFINE_ERROR(OB_ERR_CAST_NUMBER_OVERFLOW, -5087, -1, "22000", "Result value was out of range when cast to number");
DEFINE_ERROR_EXT(OB_INTEGER_PRECISION_OVERFLOW, -5088, -1, "22000", "Result value was out of range when cast varchar to number", "value larger than specified precision(%ld,%ld) allowed for this column");
DEFINE_ERROR_EXT(OB_DECIMAL_PRECISION_OVERFLOW, -5089, -1, "22000", "Result value was out of range when cast varchar to number", "value(%s) larger than specified precision(%ld,%ld) allowed for this column");
DEFINE_ERROR(OB_SCHEMA_NUMBER_PRECISION_OVERFLOW, -5090, -1, "22000", "Precision was out of range");
DEFINE_ERROR(OB_SCHEMA_NUMBER_SCALE_OVERFLOW, -5091, -1, "22000", "Scale value was out of range");
DEFINE_ERROR(OB_ERR_INDEX_UNKNOWN, -5092, -1, "42000", "Unknown index");
DEFINE_ERROR(OB_NUMERIC_OVERFLOW, -5093, -1, "22000", "numeric overflow");
DEFINE_ERROR(OB_ERR_TOO_MANY_JOIN_TABLES, -5094, -1, "HY000", "too many joined tables");
DEFINE_ERROR_EXT(OB_ERR_VARCHAR_TOO_LONG, -5098, -1, "22001", "Varchar value is too long for the column", "Data too long(%d>%ld) for column '%s'");
DEFINE_ERROR(OB_ERR_SYS_CONFIG_UNKNOWN, -5099, -1, "42000", "System config unknown");
DEFINE_ERROR_EXT(OB_ERR_LOCAL_VARIABLE, -5100, ER_LOCAL_VARIABLE, "HY000", "Local variable", "Variable \%.s\ is a SESSION variable and can't be used with SET GLOBAL");
DEFINE_ERROR_EXT(OB_ERR_GLOBAL_VARIABLE, -5101, ER_GLOBAL_VARIABLE, "HY000", "Global variable", "Variable \%.s\ is a GLOBAL variable and should be set with SET GLOBAL");
DEFINE_ERROR_EXT(OB_ERR_VARIABLE_IS_READONLY, -5102, ER_VARIABLE_IS_READONLY, "HY000", "variable is read only", "\%.s variable '%.s' is read-only. Use SET %.s to assign the value");
DEFINE_ERROR_EXT(OB_ERR_INCORRECT_GLOBAL_LOCAL_VAR, -5103, ER_INCORRECT_GLOBAL_LOCAL_VAR, "HY000", "incorrect global or local variable", "Variable '%.s' is a %.s variable");
DEFINE_ERROR_EXT(OB_ERR_EXPIRE_INFO_TOO_LONG, -5104, -1, "42000", "Expire expression too long", "length(%d) of expire_info is larger than the max allowed(%ld)");
DEFINE_ERROR_EXT(OB_ERR_EXPIRE_COND_TOO_LONG, -5105, -1, "42000", "Expire condition too long", "total length(%ld) of expire_info and its expression is larger than the max allowed(%ld)");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_EXTRACT, -5106, -1, "42000", "Invalid argument for extract()", "EXTRACT() expected timestamp or a string as date argument");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_IS, -5107, -1, "42000", "Invalid argument for IS operator", "Invalid operand type for IS operator, lval=%s");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_LENGTH, -5108, -1, "42000", "Invalid argument for length()", "function LENGTH() expected a varchar argument");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_SUBSTR, -5109, -1, "42000", "Invalid argument for substr()", "invalid input format. ret=%d text=%s start=%s length=%s");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_TIME_TO_USEC, -5110, -1, "42000", "Invalid argument for time_to_usec()", "TIME_TO_USEC() expected timestamp or a string as date argument");
DEFINE_ERROR_EXT(OB_INVALID_ARGUMENT_FOR_USEC_TO_TIME, -5111, -1, "42000", "Invalid argument for usec_to_time()", "USEC_TO_TIME expected a interger number as usec argument");
DEFINE_ERROR_EXT(OB_ERR_USER_VARIABLE_UNKNOWN, -5112, -1, "42P01", "Unknown user variable", "Variable %.s does not exists");
DEFINE_ERROR_EXT(OB_ILLEGAL_USAGE_OF_MERGING_FROZEN_TIME, -5113, -1, "42000", "Illegal usage of merging_frozen_time()", "MERGING_FROZEN_TIME() system function only be used in daily merging.");
DEFINE_ERROR_EXT(OB_INVALID_NUMERIC, -5114, -1, "42000", "Invalid numeric", "Invalid numeric char '%c'");
DEFINE_ERROR(OB_ERR_REGEXP_ERROR, -5115, ER_REGEXP_ERROR, "42000", "Got error 'empty (sub)expression' from regexp");
DEFINE_ERROR(OB_SQL_LOG_OP_SETCHILD_OVERFLOW, -5116, -1, "HY000", "Logical operator child index overflow");

```

```

DEFINE_ERROR(OB_SQL_EXPLAIN_FAILED, -5117, -1, "HY000", "fail to explain plan");
DEFINE_ERROR(OB_SQL_OPT_COPY_OP_FAILED, -5118, -1, "HY000", "fail to copy logical operator");
DEFINE_ERROR(OB_SQL_OPT_GEN_PLAN_FAILED, -5119, -1, "HY000", "fail to generate plan");;
DEFINE_ERROR(OB_SQL_OPT_CREATE_RAWEXPR_FAILED, -5120, -1, "HY000", "fail to create raw expr");
DEFINE_ERROR(OB_SQL_OPT_JOIN_ORDER_FAILED, -5121, -1, "HY000", "fail to generate join order");
DEFINE_ERROR(OB_SQL_OPT_ERROR, -5122, -1, "HY000", "optimizer general error");
DEFINE_ERROR(OB_SQL_RESOLVER_NO_MEMORY, -5130, -1, "HY000", "sql resolver no memory");
DEFINE_ERROR(OB_SQL_DML_ONLY, -5131, -1, "HY000", "plan cache support dml only");
DEFINE_ERROR(OB_ERR_NO_GRANT, -5133, -1, "42000", "No such grant defined");
DEFINE_ERROR(OB_ERR_NO_DB_SELECTED, -5134, ER_NO_DB_ERROR, "3D000", "No database selected");
DEFINE_ERROR(OB_SQL_PC_OVERFLOW, -5135, -1, "HY000", "plan cache is overflow");
DEFINE_ERROR(OB_SQL_PC_PLAN_DUPLICATE, -5136, -1, "HY000", "plan exists in plan cache already");
DEFINE_ERROR(OB_SQL_PC_PLAN_EXPIRE, -5137, -1, "HY000", "plan is expired");
DEFINE_ERROR(OB_SQL_PC_NOT_EXIST, -5138, -1, "HY000", "no plan exist");
DEFINE_ERROR(OB_SQL_PARAMS_LIMIT, -5139, -1, "HY000", "too many params, plan cache not support" );
DEFINE_ERROR(OB_SQL_PC_PLAN_SIZE_LIMIT, -5140, -1, "HY000", "plan is too big to add to plan cache");
DEFINE_ERROR_EXT(OB_ERR_UNKNOWN_CHARSET, -5142, ER_UNKNOWN_CHARACTER_SET, "42000", "Unknown character set", "Unknown character set: '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_UNKNOWN_COLLATION, -5143, ER_UNKNOWN_COLLATION, "HY000", "Unknown collation", "Unknown collation: '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_COLLATION_MISMATCH, -5144, ER_COLLATION_CHARSET_MISMATCH, "42000", "The collation is not valid for the character set", "COLLATION '%.*s' is not valid for CHARACTER SET '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_WRONG_VALUE_FOR_VAR, -5145, ER_WRONG_VALUE_FOR_VAR, "42000", "Variable can't be set to the value", "Variable \\\'%.*s\' can't be set to the value of \\\'%.*s\'");
DEFINE_ERROR_EXT(OB_UNKNOWN_PARTITION, -5146, ER_UNKNOWN_PARTITION, "HY000", "Unknown partition", "Unkown partition '%.*s' in table '%.*s'");
DEFINE_ERROR(OB_PARTITION_NOT_MATCH, -5147, ER_ROW_DOES_NOT_MATCH_GIVEN_PARTITION_SET, "HY000", "Found a row not matching the given partition set");
DEFINE_ERROR(OB_ER_PASSWD_LENGTH, -5148, -1, "HY000", " Password hash should be a 40-digit hexadecimal number" );
;
DEFINE_ERROR(OB_ERR_INSERT_INNER_JOIN_COLUMN, -5149, -1, "07000", "Insert inner join column error");
DEFINE_ERROR(OB_TENANT_NOT_IN_SERVER, -5150, -1, "HY000", "Tenant not in this server");
DEFINE_ERROR(OB_TABLEGROUP_NOT_EXIST, -5151, -1, "42P01", "tablegroup not exist");
DEFINE_ERROR(OB_SUBQUERY_TOO_MANY_ROW, -5153, ER_SUBQUERY_NO_1_ROW, "21000", "Subquery returns more than 1 row");
DEFINE_ERROR_EXT(OB_ERR_BAD_DATABASE, -5154, ER_BAD_DB_ERROR, "42000", "Unknown database", "Unknown database '%.*s'");
DEFINE_ERROR_EXT(OB_CANNOT_USER, -5155, ER_CANNOT_USER, "HY000", "User operation failed", "Operation '%.*s' failed for '%.*s'");
DEFINE_ERROR_EXT(OB_TENANT_EXIST, -5156, -1, "HY000", "tenant already exist", "tenant \\\'%s\' already exist");
DEFINE_ERROR_EXT(OB_TENANT_NOT_EXIST, -5157, -1, "HY000", "Unknown tenant", "Unknown tenant '%.*s'");
DEFINE_ERROR_EXT(OB_DATABASE_EXIST, -5158, ER_DB_CREATE_EXISTS, "HY000", "Can't create database;database exists", "Can't create database '%.*s'; database exists");
DEFINE_ERROR(OB_TABLEGROUP_EXIST, -5159, -1, "HY000", "tablegroup already exist");
DEFINE_ERROR(OB_ERR_INVALID_TENANT_NAME, -5160, -1, "HY000", "invalid tenant name specified in connection string");
;

```

```

DEFINE_ERROR(OB_EMPTY_TENANT, -5161, -1, "HY000", "tenant is empty");
DEFINE_ERROR_EXT(OB_WRONG_DB_NAME, -5162, ER_WRONG_DB_NAME, "42000", "Incorrect database name", "Incorrect database name '%.*s'");
DEFINE_ERROR_EXT(OB_WRONG_TABLE_NAME, -5163, ER_WRONG_TABLE_NAME, "42000", "Incorrect table name", "Incorrect table name '%.*s'");
DEFINE_ERROR_EXT(OB_WRONG_COLUMN_NAME, -5164, ER_WRONG_COLUMN_NAME, "42000", "Incorrect column name", "Incorrect column name '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_COLUMN_SPEC, -5165, ER_WRONG_FIELD_SPEC, "42000", "Incorrect column specifier", "Incorrect column specifier for column '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_DB_DROP_EXISTS, -5166, ER_DB_DROP_EXISTS, "HY000", "Can't drop database; database doesn't exist", "Can't drop database '%.*s'; database doesn't exist");
DEFINE_ERROR_EXT(OB_ERR_DATA_TOO_LONG, -5167, ER_DATA_TOO_LONG, "22001", "Data too long for column", "Data too long for column '%.*s' at row %lld");
DEFINE_ERROR_EXT(OB_ERR_WRONG_VALUE_COUNT_ON_ROW, -5168, ER_WRONG_VALUE_COUNT_ON_ROW, "21501", "column count does not match value count", "column count does not match value count at row %ld");
DEFINE_ERROR(OB_ERR_CREATE_USER_WITH_GRANT, -5169, ER_CANT_CREATE_USER_WITH_GRANT, "42000", "You are not allowed to create a user with GRANT");
DEFINE_ERROR_EXT(OB_ERR_NO_DB_PRIVILEGE, -5170, ER_DBACCESS_DENIED_ERROR, "42000", "Access denied for user to database", "Access denied for user '%.*s'@'%.*s' to database '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_NO_TABLE_PRIVILEGE, -5171, ER_TABLEACCESS_DENIED_ERROR, "42000", "Command denied to user for table", "%.*s command denied to user '%.*s'@'%.*s' for table '%.*s'");
DEFINE_ERROR_EXT(OB_INVALID_ON_UPDATE, -5172, ER_INVALID_ON_UPDATE, "HY000", "Invalid ON UPDATE clause", "Invalid ON UPDATE clause for '%.*s' column");
DEFINE_ERROR_EXT(OB_INVALID_DEFAULT, -5173, ER_INVALID_DEFAULT, "42000", "Invalid default value", "Invalid default value for '%.*s'");
DEFINE_ERROR_EXT(OB_ERR_UPDATE_TABLE_USED, -5174, ER_UPDATE_TABLE_USED, "HY000", "Update table used", "You can't specify target table '%.*s' for update in FROM clause");
DEFINE_ERROR_EXT(OB_ERR_COULUMN_VALUE_NOT_MATCH, -5175, ER_WRONG_VALUE_COUNT_ON_ROW, "21501", "Column count doesn't match value count", "Column count doesn't match value count at row %ld");
DEFINE_ERROR(OB_ERR_INVALID_GROUP_FUNC_USE, -5176, ER_INVALID_GROUP_FUNC_USE, "HY000", "Invalid use of group function");
DEFINE_ERROR_EXT(OB_CANT_AGGREGATE_2COLLATIONS, -5177, ER_CANT_AGGREGATE_2COLLATIONS, "HY000", "Illegal mix of collations", "Illegal mix of collations");
DEFINE_ERROR_EXT(OB_ERR_FIELD_TYPE_NOT_ALLOWED_AS_PARTITION_FIELD, -5178, ER_FIELD_TYPE_NOT_ALLOWED_AS_PARTITION_FIELD, "HY000", "Field is of a not allowed type for this type of partitioning", "Field '%.*s' is of a not allowed type for this type of partitioning");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_IDENT, -5179, ER_TOO_LONG_IDENT, "42000", "Identifier name is too long", "Identifier name '%.*s' is too long");
DEFINE_ERROR_EXT(OB_ERR_WRONG_TYPE_FOR_VAR, -5180, ER_WRONG_TYPE_FOR_VAR, "42000", "Incorrect argument type to variable", "Incorrect argument type to variable '%.*s'");
DEFINE_ERROR_EXT(OB_WRONG_USER_NAME_LENGTH, -5181, ER_WRONG_STRING_LENGTH, "HY000", "String is too long for user_name (should be no longer than 16)", "String '%.*s' is too long for user name (should be no longer than 16)");
DEFINE_ERROR(OB_ERR_PRIV_USAGE, -5182, ER_WRONG_USAGE, "HY000", "Incorrect usage of DB GRANT and GLOBAL PRIVILEGES");
DEFINE_ERROR(OB_ILLEGAL_GRANT_FOR_TABLE, -5183, ER_ILLEGAL_GRANT_FOR_TABLE, "42000", "Illegal GRANT/REVOKE command; please consult the manual to see which privileges can be used");
DEFINE_ERROR(OB_ERR_REACH_AUTOINC_MAX, -5184, ER_AUTOINC_READ_FAILED, "HY000", "Failed to read auto-increment");

```

```

t value from storage engine");
DEFINE_ERROR(OB_ERR_NO_TABLES_USED, -5185, ER_NO_TABLES_USED, "HY000", "No tables used");
DEFINE_ERROR(OB_CANT_REMOVE_ALL_FIELDS, -5187, ER_CANT_REMOVE_ALL_FIELDS, "42000", "You can't delete all columns with ALTER TABLE; use DROP TABLE instead");
DEFINE_ERROR(OB_TOO_MANY_PARTITIONS_ERROR, -5188, ER_TOO_MANY_PARTITIONS_ERROR, "HY000", "Too many partitions (including subpartitions) were defined", "Too many partitions (including subpartitions) were defined");
DEFINE_ERROR(OB_NO_PARTS_ERROR, -5189, ER_NO_PARTS_ERROR, "HY000", "Number of partitions = 0 is not an allowed value", "Number of partitions = 0 is not an allowed value");
DEFINE_ERROR(OB_WRONG_SUB_KEY, -5190, ER_WRONG_SUB_KEY, "HY000", "Incorrect prefix key; the used key part isn't a string, the used length is longer than the key part, or the storage engine doesn't support unique prefix keys");
DEFINE_ERROR_EXT(OB_KEY_PART_0, -5191, ER_KEY_PART_0, "HY000", "Key part length cannot be 0", "Key part \'.*s\' length cannot be 0");
DEFINE_ERROR_EXT(OB_ERR_UNKNOWN_TIME_ZONE, -5192, ER_UNKNOWN_TIME_ZONE, "HY000", "Unknown or incorrect time zone", "Unknown or incorrect time zone: \'.*s\'");
DEFINE_ERROR(OB_ERR_WRONG_AUTO_KEY, -5193, ER_WRONG_AUTO_KEY, "42000", "Incorrect table definition; there can be only one auto column");
DEFINE_ERROR_EXT(OB_ERR_TOO_MANY_KEYS, -5194, ER_TOO_MANY_KEYS, "42000", "Too many keys specified", "Too many keys specified; max %d keys allowed");
DEFINE_ERROR_EXT(OB_ERR_TOO_MANY_ROWKEY_COLUMNS, -5195, ER_TOO_MANY_KEY_PARTS, "42000", "Too many key parts specified", "Too many key parts specified; max %d parts allowed");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_KEY_LENGTH, -5196, ER_TOO_LONG_KEY, "42000", "Specified key was too long", "Specified key was too long; max key length is %d bytes");
DEFINE_ERROR(OB_ERR_TOO_MANY_COLUMNS, -5197, ER_TOO_MANY_FIELDS, "42000", "Too many columns");
DEFINE_ERROR_EXT(OB_ERR_TOO_LONG_COLUMN_LENGTH, -5198, ER_TOO_BIG_FIELDLNGTH, "42000", "Column length too big", "Column length too big for column '%s' (max = %d)");
DEFINE_ERROR(OB_ERR_TOO_BIG_ROWSIZE, -5199, ER_TOO_BIG_ROWSIZE, "42000", "Row size too large");
DEFINE_ERROR_EXT(OB_ERR_UNKNOWN_TABLE, -5200, ER_UNKNOWN_TABLE, "42502", "Unknown table", "Unknown table '%s' in %s");
DEFINE_ERROR_EXT(OB_ERR_BAD_TABLE, -5201, ER_BAD_TABLE_ERROR, "42502", "Unknown table", "Unknown table '%s'");
DEFINE_ERROR(OB_ERR_TOO_BIG_SCALE, -5202, ER_TOO_BIG_SCALE, "42000", "Too big scale %d specified for column '%s'. Maximum is %ld.");
DEFINE_ERROR(OB_ERR_TOO_BIG_PRECISION, -5203, ER_TOO_BIG_PRECISION, "42000", "Too big precision %d specified for column '%s'. Maximum is %ld.");
DEFINE_ERROR(OB_ERR_M_BIGGER_THAN_D, -5204, ER_M_BIGGER_THAN_D, "42000", "For float(M,D), double(M,D) or decimal(M,D), M must be >= D (column '%s').");
DEFINE_ERROR(OB_ERR_TOO_BIG_DISPLAYWIDTH, -5205, ER_TOO_BIG_DISPLAYWIDTH, "42000", "Display width out of range for column '%s' (max = %ld)");
DEFINE_ERROR(OB_WRONG_GROUP_FIELD, -5206, ER_WRONG_GROUP_FIELD, "42000", "Can't group on '%s'");
DEFINE_ERROR_EXT(OB_NON_UNIQ_ERROR, -5207, ER_NON_UNIQ_ERROR, "23000", "Column is ambiguous", "Column '%s' in %s is ambiguous");
DEFINE_ERROR_EXT(OB_ERR_NONUNIQ_TABLE, -5208, ER_NONUNIQ_TABLE, "42000", "Not unique table/alias", "Not unique table/alias: \'.*s\'");
DEFINE_ERROR_EXT(OB_ERR_CANT_DROP_FIELD_OR_KEY, -5209, ER_CANT_DROP_FIELD_OR_KEY, "42000", "Can't DROP Column; check that column/key exists", "Can't DROP '%s'; check that column/key exists");
DEFINE_ERROR(OB_ERR_MULTIPLE_PRI_KEY, -5210, ER_MULTIPLE_PRI_KEY, "42000", "Multiple primary key defined");
DEFINE_ERROR_EXT(OB_ERR_KEY_COLUMN_DOES_NOT_EXISTS, -5211, ER_KEY_COLUMN_DOES_NOT_EXISTS, "42000", "Key column does not exist");

```

```

DEFINE_ERROR_EXT(OB_ERR_KEY_COLUMN_EXISTS, -5210, ER_KEY_COLUMN_EXISTS, "42000", "Key column doesn't exist in table", "Key column '%s' doesn't exist in table");
DEFINE_ERROR_EXT(OB_ERR_AUTO_PARTITION_KEY, -5212, -1, "42000", "auto-increment column should not be part of partition key", "auto-increment column '%s' should not be part of partition key");
DEFINE_ERROR_EXT(OB_ERR_CANT_USE_OPTION_HERE, -5213, ER_CANT_USE_OPTION_HERE, "42000", "Incorrect usage/placement", "Incorrect usage/placement of '%s'");
DEFINE_ERROR_EXT(OB_ERR_WRONG_OBJECT, -5214, ER_WRONG_OBJECT, "HY000", "Wrong object", "\'%s.%s\' is not %s");
DEFINE_ERROR_EXT(OB_ERR_ON_RENAME, -5215, ER_ERROR_ON_RENAME, "HY000", "Error on rename table", "Error on rename of \'%s.%s\' to \'%s.%s\'");
DEFINE_ERROR_EXT(OB_ERR_WRONG_KEY_COLUMN, -5216, ER_WRONG_KEY_COLUMN, "42000", "The used storage engine can't index column", "The used storage engine can't index column '%s'");
DEFINE_ERROR_EXT(OB_ERR_BAD_FIELD_ERROR, -5217, ER_BAD_FIELD_ERROR, "42S22", "Unknown column", "Unknown column '%s' in '%s'");
DEFINE_ERROR_EXT(OB_ERR_WRONG_FIELD_WITH_GROUP, -5218, ER_WRONG_FIELD_WITH_GROUP, "42000", "column is not in GROUP BY", "\'%s\' is not in GROUP BY");
DEFINE_ERROR(OB_ERR_CANT_CHANGE_TX_CHARACTERISTICS, -5219, ER_CANT_CHANGE_TX_CHARACTERISTICS, "25001", "Transaction characteristics can't be changed while a transaction is in progress");
DEFINE_ERROR(OB_ERR_CANT_EXECUTE_IN_READ_ONLY_TRANSACTION, -5220, ER_CANT_EXECUTE_IN_READ_ONLY_TRANSACTION, "25006", "Cannot execute statement in a READ ONLY transaction.");
DEFINE_ERROR(OB_ERR_MIX_OF_GROUP_FUNC_AND_FIELDS, -5221, ER_MIX_OF_GROUP_FUNC_AND_FIELDS, "42000", "Mixing of GROUP columns (MIN(),MAX(),COUNT(),...) with no GROUP columns is illegal if there is no GROUP BY clause");
DEFINE_ERROR_EXT(OB_ERR_TRUNCATED_WRONG_VALUE, -5222, ER_TRUNCATED_WRONG_VALUE, "22007", "Incorrect value", "Truncated incorrect %s value: '%s'");
DEFINE_ERROR(OB_ERR_WRONG_IDENT_NAME, -5223, -1, "42000", "wrong ident name");
DEFINE_ERROR_EXT(OB_WRONG_NAME_FOR_INDEX, -5224, ER_WRONG_NAME_FOR_INDEX, "42000", "Incorrect index name", "Incorrect index name '%s'");
DEFINE_ERROR_EXT(OB_ILLEGAL_REFERENCE, -5225, ER_ILLEGAL_REFERENCE, "42S22", "Reference not supported (reference to group function)", "Reference '%s' not supported (reference to group function)");
DEFINE_ERROR(OB_REACH_MEMORY_LIMIT, -5226, -1, "42000", "plan cache memory used reach the high water mark.");
DEFINE_ERROR(OB_ERR_PASSWORD_FORMAT, -5227, ER_PASSWORD_FORMAT, "42000", "The password hash doesn't have the expected format. Check if the correct password algorithm is being used with the PASSWORD() function.");
DEFINE_ERROR(OB_ERR_NON_UPDATABLE_TABLE, -5228, ER_NON_UPDATABLE_TABLE, "HY000", "The target table of the UPDATE is not updatable");
DEFINE_ERROR_EXT(OB_ERR_WARN_DATA_OUT_OF_RANGE, -5229, ER_WARN_DATA_OUT_OF_RANGE, "22003", "Out of range value for column", "Out of range value for column '%s' at row %ld");
DEFINE_ERROR(OB_ERR_WRONG_EXPR_IN_PARTITION_FUNC_ERROR, -5230, ER_WRONG_EXPR_IN_PARTITION_FUNC_ERROR, "HY000", "Constant or random or timezone-dependent expressions in (sub)partitioning function are not allowed");
DEFINE_ERROR_EXT(OB_ERR_VIEW_INVALID, -5231, ER_VIEW_INVALID, "42S22", "view invalid", "View \'%s.%s\' references invalid table(s) or column(s) or function(s) or definer/invoke of view lack rights to use them");
DEFINE_ERROR(OB_ERR_OPTION_PREVENTS_STATEMENT, -5233, ER_OPTION_PREVENTS_STATEMENT, "HY000", "The MySQL server is running with the --read-only option so it cannot execute this statement");
DEFINE_ERROR(OB_ERR_DB_READ_ONLY, -5234, -1, "HY000", "The database is read only so it cannot execute this statement", "The database \'%s\' is read only so it cannot execute this statement");
DEFINE_ERROR(OB_ERR_TABLE_READ_ONLY, -5235, -1, "HY000", "The table is read only so it cannot execute this statement", "The table \'%s.%s\' is read only so it cannot execute this statement");
DEFINE_ERROR(OB_ERR_LOCK_OR_ACTIVE_TRANSACTION, -5236, ER_LOCK_OR_ACTIVE_TRANSACTION, "HY000", "Can't execute the given command because you have active locked tables or an active transaction");

```

```

DEFINE_ERROR_EXT(OB_ERR_SAME_NAME_PARTITION_FIELD, -5237, ER_SAME_NAME_PARTITION_FIELD, "HY000", "Duplicate
e partition field name", "Duplicate partition field name '%.s'");
DEFINE_ERROR_EXT(OB_ERR_TABLENAME_NOT_ALLOWED_HERE, -5238, ER_TABLENAME_NOT_ALLOWED_HERE, "42000", "Ta
ble from one of the SELECTs cannot be used in global ORDER clause", "Table \%.s\ from one of the SELECTs cannot be
used in global ORDER clause");
DEFINE_ERROR_EXT(OB_ERR_VIEW_RECURSIVE, -5239, ER_VIEW_RECURSIVE, "42522", "view contains recursion", "\%.s\ '%.
*s\ contains view recursion");
DEFINE_ERROR(OB_ERR_QUALIFIER, -5240, -1, "HY000", "Column part of USING clause cannot have qualifier");
DEFINE_ERROR(OB_ERR_WRONG_VALUE, -5241, ER_WRONG_VALUE, "HY000", "Incorrect %s value: '%s'");
DEFINE_ERROR(OB_ERR_VIEW_WRONG_LIST, -5242, ER_VIEW_WRONG_LIST, "HY000", "View's SELECT and view's field list h
ave different column counts");
DEFINE_ERROR(OB_SYS_VARS_MAYBE_DIFF_VERSION, -5243, -1, "HY000", "system variables' version maybe different");
DEFINE_ERROR(OB_ERR_AUTO_INCREMENT_CONFLICT, -5244, ER_AUTO_INCREMENT_CONFLICT, "HY000", "Auto-increment v
alue in UPDATE conflicts with internally generated values");
DEFINE_ERROR_EXT(OB_ERR_TASK_SKIPPED, -5245, -1, "HY000", "some tasks are skipped", "some tasks are skipped, skip
ped server addr is '%s', the original error code is %d");
DEFINE_ERROR_EXT(OB_ERR_NAME_BECOMES_EMPTY, -5246, ER_NAME_BECOMES_EMPTY, "HY000", "Name has become '',
'Name \%.s\ has become '");
DEFINE_ERROR_EXT(OB_ERR_REMOVED_SPACES, -5247, ER_REMOVED_SPACES, "HY000", "Leading spaces are removed fro
m name ", "Leading spaces are removed from name \%.s\");
DEFINE_ERROR_EXT(OB_WARN_ADD_AUTOINCREMENT_COLUMN, -5248, -1, "HY000", "Alter table add auto_increment colu
mn is dangerous", "Alter table add auto_increment column is dangerous, table_name=\%.s\, column_name=\%.s\");
DEFINE_ERROR_EXT(OB_WARN_CHAMGE_NULL_ATTRIBUTE, -5249, -1, "HY000", "Alter table change nullable column to not
nullable is dangerous", "Alter table change nullable column to not nullable is dangerous, table_name=\%.s\, column_
name=\%.s\");
DEFINE_ERROR_EXT(OB_ERR_INVALID_CHARACTER_STRING, -5250, ER_INVALID_CHARACTER_STRING, "HY000", "Invalid char
acter string", "Invalid %s character string: \%.s\");
DEFINE_ERROR_EXT(OB_ERR_KILL_DENIED, -5251, ER_KILL_DENIED_ERROR, "HY000", "You are not owner of thread", "You a
re not owner of thread %lu");
DEFINE_ERROR_EXT(OB_ERR_COLUMN_DEFINITION_AMBIGUOUS, -5252, -1, "HY000", "Column definition is ambiguous. Colu
mn has both NULL and NOT NULL attributes", "Column definition is ambiguous. Column has both NULL and NOT NULL at
tributes");
DEFINE_ERROR(OB_ERR_EMPTY_QUERY, -5253, ER_EMPTY_QUERY, "42000", "Query was empty");
DEFINE_ERROR_EXT(OB_ERR_CUT_VALUE_GROUP_CONCAT, -5254, ER_CUT_VALUE_GROUP_CONCAT, "42000", "Row was cut
by GROUP_CONCAT()", "Row %ld was cut by GROUP_CONCAT()");
DEFINE_ERROR(OB_ERR_FILED_NOT_FOUND_PART, -5255, ER_FIELD_NOT_FOUND_PART_ERROR, "HY000", "Field in list of fie
lds for partition function not found in table");
DEFINE_ERROR(OB_ERR_PRIMARY_CANT_HAVE_NULL, -5256, ER_PRIMARY_CANT_HAVE_NULL, "42000", "All parts of a PRIM
ARY KEY must be NOT NULL; if you need NULL in a key, use UNIQUE instead");
DEFINE_ERROR(OB_ERR_PARTITION_FUNC_NOT_ALLOWED_ERROR, -5257, ER_PARTITION_FUNC_NOT_ALLOWED_ERROR, "HY
000", "The PARTITION function returns the wrong type");
DEFINE_ERROR(OB_ERR_INVALID_BLOCK_SIZE, -5258, -1, "HY000", "Invalid block size, block size should between 16384 an
d 1048576");
DEFINE_ERROR_EXT(OB_ERR_UNKNOWN_STORAGE_ENGINE, -5259, ER_UNKNOWN_STORAGE_ENGINE, "42000", "Unknown st
orage engine", "Unknown storage engine \%.s\");
DEFINE_ERROR_EXT(OB_ERR_TENANT_IS_LOCKED, -5260, -1, "HY000", "Tenant is locked", "Tenant \%.s\ is locked");
DEFINE_ERROR_EXT(OB_EER_UNIQUE_KEY_NEED_ALL_FIELDS_IN_PF, -5261, ER_UNIQUE_KEY_NEED_ALL_FIELDS_IN_PF, "HYO

```

```

00", "A UNIQUE INDEX/PRIMARY KEY must include all columns in the table's partitioning function", "A %s must include all
columns in the table's partitioning function";
DEFINE_ERROR(OB_ERR_PARTITION_FUNCTION_IS_NOT_ALLOWED, -5262, ER_PARTITION_FUNCTION_IS_NOT_ALLOWED, "H
Y000", "This partition function is not allowed");
DEFINE_ERROR_EXT(OB_ERR_AGGREGATE_ORDER_FOR_UNION, -5263, ER_AGGREGATE_ORDER_FOR_UNION, "HY000", "aggre
gate order for union", "Expression #%d of ORDER BY contains aggregate function and applies to a UNION");
DEFINE_ERROR_EXT(OB_ERR_OUTLINE_EXIST, -5264, -1, "HY000", "Outline exists", "Outline '%.s' already exists");
DEFINE_ERROR_EXT(OB_OUTLINE_NOT_EXIST, -5265, -1, "HY000", "Outline not exists", "Outline \%.s. %.s\ doesn't exis
t");
DEFINE_ERROR_EXT(OB_WARN_OPTION_BELOW_LIMIT, -5266, WARN_OPTION_BELOW_LIMIT, "HY000", "The value should b
e no less than the limit", "The value of \%.s\ should be no less than the value of \%.s\");
DEFINE_ERROR_EXT(OB_INVALID_OUTLINE, -5267, -1, "HY000", "invalid outline", "invalid outline ,error info:%s");
DEFINE_ERROR_EXT(OB_REACH_MAX_CONCURRENT_NUM, -5268, -1, "HY000", "SQL reach max concurrent num", "SQL reac
h max concurrent num %ld");
DEFINE_ERROR(OB_ERR_OPERATION_ON_RECYCLE_OBJECT, -5269, -1, "HY000", "can not perform DDL/DML over objects in
Recycle Bin");
DEFINE_ERROR(OB_ERR_OBJECT_NOT_IN_RECYCLEBIN, -5270, -1, "HY000", "object not in RECYCLE BIN");
DEFINE_ERROR(OB_ERR_CON_COUNT_ERROR, -5271, ER_CON_COUNT_ERROR, "08004", "Too many connections");
DEFINE_ERROR_EXT(OB_ERR_OUTLINE_CONTENT_EXIST, -5272, -1, "HY000", "Outline content already exists when added",
"Outline content '%.s' of outline '%.s' already exists when added");
DEFINE_ERROR_EXT(OB_ERR_OUTLINE_MAX_CONCURRENT_EXIST, -5273, -1, "HY000", "Max concurrent already exists whe
n added", "Max concurrent in outline '%.s' already exists when added");
DEFINE_ERROR_EXT(OB_ERR_VALUES_IS_NOT_INT_TYPE_ERROR, -5274, ER_VALUES_IS_NOT_INT_TYPE_ERROR, "HY000", "V
ALUES value for partition must have type INT", "VALUES value for partition \%.s\ must have type INT");
DEFINE_ERROR(OB_ERR_WRONG_TYPE_COLUMN_VALUE_ERROR, -5275, ER_WRONG_TYPE_COLUMN_VALUE_ERROR, "HY000"
, "Partition column values of incorrect type");
DEFINE_ERROR(OB_ERR_PARTITION_COLUMN_LIST_ERROR, -5276, ER_PARTITION_COLUMN_LIST_ERROR, "HY000", "Inconsis
tency in usage of column lists for partitioning");
DEFINE_ERROR(OB_ERR_TOO_MANY_VALUES_ERROR, -5277, ER_TOO_MANY_VALUES_ERROR, "HY000", "Cannot have more
than one value for this type of RANGE partitioning");
DEFINE_ERROR(OB_ERR_PARTITION_FUNCTION_IS_NOT_ALLOWED, -5278, ER_PARTITION_FUNCTION_IS_NOT_ALLOWED, "H
Y000", "This partition function is not allowed");
DEFINE_ERROR(OB_ERR_PARTITION_INTERVAL_ERROR, -5279, -1, "HY000", "Partition interval must have type INT");
DEFINE_ERROR_EXT(OB_ERR_SAME_NAME_PARTITION, -5280, ER_SAME_NAME_PARTITION, "HY000", "Duplicate partition na
me", "Duplicate partition name \%.s\");
DEFINE_ERROR(OB_ERR_RANGE_NOT_INCREASING_ERROR, -5281, ER_RANGE_NOT_INCREASING_ERROR, "HY000", "VALUES L
ESS THAN value must be strictly increasing for each partition");
DEFINE_ERROR(OB_ERR_PARSE_PARTITION_RANGE, -5282, ER_PARSE_ERROR, "42000", "Wrong number of partitions define
d, mismatch with previous setting");
DEFINE_ERROR(OB_ERR_UNIQUE_KEY_NEED_ALL_FIELDS_IN_PF, -5283, ER_UNIQUE_KEY_NEED_ALL_FIELDS_IN_PF, "HY000", "
A PRIMARY KEY must include all columns in the table\'s partitioning function");
DEFINE_ERROR(OB_NO_PARTITION_FOR_GIVEN_VALUE, -5284, ER_NO_PARTITION_FOR_GIVEN_VALUE, "HY000", "Table has
no partition for value");
DEFINE_ERROR(OB_EER_NULL_IN_VALUES_LESS_THAN, -5285, ER_NULL_IN_VALUES_LESS_THAN, "HY000", "Not allowed to
use NULL value in VALUES LESS THAN");
DEFINE_ERROR(OB_ERR_PARTITION_CONST_DOMAIN_ERROR, -5286, ER_PARTITION_CONST_DOMAIN_ERROR, "HY000", "Part
ition constant is out of partition function domain");

```

```

non constant is out of partition function domain );
DEFINE_ERROR(OB_ERR_TOO_MANY_PARTITION_FUNC_FIELDS, -5287, ER_TOO_MANY_PARTITION_FUNC_FIELDS_ERROR, "HY000", "Too many fields in \list of partition fields\");
DEFINE_ERROR_EXT(OB_ERR_BAD_FT_COLUMN, -5288, ER_BAD_FT_COLUMN, "HY000", "Column cannot be part of FULLTEXT index", "Column '%.s' cannot be part of FULLTEXT index");
DEFINE_ERROR_EXT(OB_ERR_KEY_DOES_NOT_EXISTS, -5289, ER_KEY_DOES_NOT_EXISTS, "42000", "key does not exist in table", "Key '%.s' doesn't exist in table '%.s'");
DEFINE_ERROR_EXT(OB_NON_DEFAULT_VALUE_FOR_GENERATED_COLUMN, -5290, ER_NON_DEFAULT_VALUE_FOR_GENERATED_COLUMN, "HY000", "non-default value for generated column is not allowed", "The value specified for generated column '%.s' in table '%.s' is not allowed");
DEFINE_ERROR(OB_ERR_BAD_CTXCAT_COLUMN, -5291, -1, "HY000", "The CTXCAT column must be contiguous in the index column list");
DEFINE_ERROR_EXT(OB_ERR_UNSUPPORTED_ACTION_ON_GENERATED_COLUMN, -5292, ER_UNSUPPORTED_ACTION_ON_GENERATED_COLUMN, "HY000", "not supported for generated columns", "%s' is not supported for generated columns.");
DEFINE_ERROR_EXT(OB_ERR_DEPENDENT_BY_GENERATED_COLUMN, -5293, ER_DEPENDENT_BY_GENERATED_COLUMN, "HY000", "Column has a generated column dependency", "Column '%.s' has a generated column dependency");
DEFINE_ERROR(OB_ERR_TOO_MANY_ROWS, -5294, ER_TOO_MANY_ROWS, "42000", "Result consisted of more than one row");
DEFINE_ERROR(OB_WRONG_FIELD_TERMINATORS, -5295, ER_WRONG_FIELD_TERMINATORS, "42000", "Field separator argument is not what is expected; check the manual");
DEFINE_ERROR(OB_NO_READABLE_REPLICA, -5296, -1, "42000", "there has no readable replica");
DEFINE_ERROR(OB_ERR_PARTITION_MGMT_ON_NONPARTITIONED, -5302, ER_PARTITION_MGMT_ON_NONPARTITIONED, "HY000", "Partition management on a not partitioned table is not possible");
DEFINE_ERROR_EXT(OB_ERR_DROP_PARTITION_NON_EXISTENT, -5303, ER_DROP_PARTITION_NON_EXISTENT, "HY000", "Error in list of partitions", "Error in list of partitions to %s");
DEFINE_ERROR(OB_ERR_PARTITION_MGMT_ON_TWOPART_TABLE, -5304, -1, "HY000", "Partition management on a two-part table is not possible");
DEFINE_ERROR(OB_ERR_ONLY_ON_RANGE_LIST_PARTITION, -5305, ER_ONLY_ON_RANGE_LIST_PARTITION, "HY000", "can only be used on RANGE/LIST partitions", "%s PARTITION can only be used on RANGE/LIST partitions");
DEFINE_ERROR(OB_ERR_DROP_LAST_PARTITION, -5306, ER_DROP_LAST_PARTITION, "HY000", "Cannot remove all partitions, use DROP TABLE instead");

////////////////////////////////////
//error code for transaction, mvcc and commitlog -6001 ---- -7000
////////////////////////////////////
DEFINE_ERROR(OB_TRANSACTION_SET_VIOLATION, -6001, -1, "25000", "Transaction set changed during the execution");
DEFINE_ERROR_EXT(OB_TRANS_ROLLED_BACK, -6002, -1, "40000", "Transaction rolled back", "transaction is rolled back");
DEFINE_ERROR(OB_ERR_EXCLUSIVE_LOCK_CONFLICT, -6003, ER_LOCK_WAIT_TIMEOUT, "HY000", "Lock wait timeout exceeded; try restarting transaction");
DEFINE_ERROR(OB_ERR_SHARED_LOCK_CONFLICT, -6004, -1, "HY000", "Shared lock conflict");
DEFINE_ERROR(OB_TRY_LOCK_ROW_CONFLICT, -6005, -1, "HY000", "Try lock row conflict");
DEFINE_ERROR(OB_CLOCK_OUT_OF_ORDER, -6201, -1, "25000", "Clock out of order");
DEFINE_ERROR(OB_MASK_SET_NO_NODE, -6203, -1, "25000", "Mask set has no node");
DEFINE_ERROR(OB_TRANS_HAS_DECIDED, -6204, -1, "HY000", "Transaction has been decided");
DEFINE_ERROR(OB_TRANS_INVALID_STATE, -6205, -1, "HY000", "Transaction state invalid");
DEFINE_ERROR(OB_TRANS_STATE_NOT_CHANGE, -6206, -1, "HY000", "Transaction state not changed");
DEFINE_ERROR(OB_TRANS_PROTOCOL_ERROR, -6207, -1, "HY000", "Transaction protocol error");

```

```

DEFINE_ERROR(OB_TRANS_INVALID_MESSAGE, -6208, -1, "HY000", "Transaction message invalid");
DEFINE_ERROR(OB_TRANS_INVALID_MESSAGE_TYPE, -6209, -1, "HY000", "Transaction message type invalid");
DEFINE_ERROR(OB_TRANS_TIMEOUT, -6210, 4012, "25000", "Transaction is timeout");
DEFINE_ERROR(OB_TRANS_KILLED, -6211, 6002, "25000", "Transaction is killed");
DEFINE_ERROR(OB_TRANS_STMT_TIMEOUT, -6212, 4012, "25000", "Statement is timeout");
DEFINE_ERROR(OB_TRANS_CTX_NOT_EXIST, -6213, 6002, "HY000", "Transaction context does not exist");
DEFINE_ERROR(OB_PARTITION_IS_FROZEN, -6214, 6002, "25000", "Partition is frozen");
DEFINE_ERROR(OB_PARTITION_IS_NOT_FROZEN, -6215, -1, "HY000", "Partition is not frozen");
DEFINE_ERROR(OB_TRANS_INVALID_LOG_TYPE, -6219, -1, "HY000", "Transaction invalid log type");
DEFINE_ERROR(OB_TRANS_SQL_SEQUENCE_ILLEGAL, -6220, -1, "HY000", "SQL sequence illegal");
DEFINE_ERROR(OB_TRANS_CANNOT_BE_KILLED, -6221, -1, "HY000", "Transaction context cannot be killed");
DEFINE_ERROR(OB_TRANS_STATE_UNKNOWN, -6222, -1, "HY000", "Transaction state unknown");
DEFINE_ERROR(OB_TRANS_IS_EXITING, -6223, 6002, "25000", "Transaction exiting");
DEFINE_ERROR(OB_TRANS_NEED_ROLLBACK, -6224, 6002, "25000", "transaction need rollback");
DEFINE_ERROR(OB_TRANS_UNKNOWN, -6225, 4012, "25000", "Transaction result is unknown");
DEFINE_ERROR(OB_ERR_READ_ONLY_TRANSACTION, -6226, ER_CANT_EXECUTE_IN_READ_ONLY_TRANSACTION, "25006", "C
annot execute statement in a READ ONLY transaction");
DEFINE_ERROR(OB_PARTITION_IS_NOT_STOPPED, -6227, -1, "HY000", "Partition is not stopped");
DEFINE_ERROR(OB_PARTITION_IS_STOPPED, -6228, -1, "HY000", "Partition has been stopped");
DEFINE_ERROR(OB_PARTITION_IS_BLOCKED, -6229, -1, "HY000", "Partition has been blocked");
DEFINE_ERROR(OB_TRANS_RPC_TIMEOUT, -6230, 4012, "25000", "transaction rpc timeout");
DEFINE_ERROR(OB_REPLICA_NOT_READABLE, -6231, -1, "HY000", "replica is not readable");

// for clog
DEFINE_ERROR(OB_LOG_ID_NOT_FOUND, -6301, -1, "HY000", "log id not found");
DEFINE_ERROR(OB_LSR_THREAD_STOPPED, -6302, -1, "HY000", "log scan runnable thread stop");
DEFINE_ERROR(OB_NO_LOG, -6303, -1, "HY000", "no log ever scanned");
DEFINE_ERROR(OB_LOG_ID_RANGE_ERROR, -6304, -1, "HY000", "log id range error");
DEFINE_ERROR(OB_LOG_ITER_ENOUGH, -6305, -1, "HY000", "iter scans enough files");
DEFINE_ERROR(OB_CLOG_INVALID_ACK, -6306, -1, "HY000", "invalid ack msg");
DEFINE_ERROR(OB_CLOG_CACHE_INVALID, -6307, -1, "HY000", "clog cache invalid");
DEFINE_ERROR(OB_EXT_HANDLE_UNFINISH, -6308, -1, "HY000", "external executor handle do not finish");
DEFINE_ERROR(OB_CURSOR_NOT_EXIST, -6309, -1, "HY000", "cursor not exist");
DEFINE_ERROR(OB_STREAM_NOT_EXIST, -6310, -1, "HY000", "stream not exist");
DEFINE_ERROR(OB_STREAM_BUSY, -6311, -1, "HY000", "stream busy");
DEFINE_ERROR(OB_FILE_RECYCLED, -6312, -1, "HY000", "file recycled");
DEFINE_ERROR(OB_REPLAY_EAGAIN_TOO_MUCH_TIME, -6313, -1, "HY000", "replay eagain cost too much time");
DEFINE_ERROR(OB_MEMBER_CHANGE_FAILED, -6314, -1, "HY000", "member change log sync failed");

////////////////////////////////////
//error code for election -7000 ---- -7100
////////////////////////////////////
DEFINE_ERROR(OB_ELECTION_WARN_LOGBUF_FULL, -7000, -1, "HY000", "The log buffer is full");
DEFINE_ERROR(OB_ELECTION_WARN_LOGBUF_EMPTY, -7001, -1, "HY000", "The log buffer is empty");
DEFINE_ERROR(OB_ELECTION_WARN_NOT_RUNNING, -7002, -1, "HY000", "The object is not running");
DEFINE_ERROR(OB_ELECTION_WARN_IS_RUNNING, -7003, -1, "HY000", "The object is running");
DEFINE_ERROR(OB_ELECTION_WARN_NOT_REACH_MAJORITY, -7004, -1, "HY000", "Election does not reach majority");

```

```

DEFINE_ERROR(OB_ELECTION_WARN_INVALID_SERVER, -7005, -1, "HY000", "The server is not valid");
DEFINE_ERROR(OB_ELECTION_WARN_INVALID_LEADER, -7006, -1, "HY000", "The leader is not valid");
DEFINE_ERROR(OB_ELECTION_WARN_LEADER_LEASE_EXPIRED, -7007, -1, "HY000", "The leader lease is expired");
DEFINE_ERROR(OB_ELECTION_WARN_INVALID_MESSAGE, -7010, -1, "HY000", "The message is not valid");
DEFINE_ERROR(OB_ELECTION_WARN_MESSAGE_NOT_INTIME, -7011, -1, "HY000", "The message is not intime");
DEFINE_ERROR(OB_ELECTION_WARN_NOT_CANDIDATE, -7012, -1, "HY000", "The server is not candidate");
DEFINE_ERROR(OB_ELECTION_WARN_NOT_CANDIDATE_OR_VOTER, -7013, -1, "HY000", "The server is not candidate or voter");
DEFINE_ERROR(OB_ELECTION_WARN_PROTOCOL_ERROR, -7014, -1, "HY000", "Election protocol error");
DEFINE_ERROR(OB_ELECTION_WARN_RUNTIME_OUT_OF_RANGE, -7015, -1, "HY000", "The task run time out of range");
DEFINE_ERROR(OB_ELECTION_WARN_LAST_OPERATION_NOT_DONE, -7021, -1, "HY000", "Last operation has not done");
DEFINE_ERROR(OB_ELECTION_WARN_CURRENT_SERVER_NOT_LEADER, -7022, -1, "HY000", "Current server is not leader");
DEFINE_ERROR(OB_ELECTION_WARN_NO_PREPARE_MESSAGE, -7024, -1, "HY000", "There is not prepare message");
DEFINE_ERROR(OB_ELECTION_ERROR_MULTI_PREPARE_MESSAGE, -7025, -1, "HY000", "There is more than one prepare message");
DEFINE_ERROR(OB_ELECTION_NOT_EXIST, -7026, -1, "HY000", "Election does not exist");
DEFINE_ERROR(OB_ELECTION_MGR_IS_RUNNING, -7027, -1, "HY000", "Election manager is running");
DEFINE_ERROR(OB_ELECTION_WARN_NO_MAJORITY_PREPARE_MESSAGE, -7029, -1, "HY000", "Election msg pool not have majority prepare message");
DEFINE_ERROR(OB_ELECTION_ASYNC_LOG_WARN_INIT, -7030, -1, "HY000", "Election async log init error");
DEFINE_ERROR(OB_ELECTION_WAIT_LEADER_MESSAGE, -7031, -1, "HY000", "Election waiting leader message");

////////////////////////////////////
// !!! Fatal errors and the client should close the connection, -8000 ~ -8999
////////////////////////////////////
DEFINE_ERROR(OB_SERVER_IS_INIT, -8001, -1, "08004", "Server is initializing");
DEFINE_ERROR(OB_SERVER_IS_STOPPING, -8002, -1, "08004", "Server is stopping");
DEFINE_ERROR(OB_PACKET_CHECKSUM_ERROR, -8003, -1, "08004", "Packet checksum error");
DEFINE_ERROR(OB_PACKET_CLUSTER_ID_NOT_MATCH, -8004, -1, "08004", "Packet cluster_id not match");
    
```

18.5.24. Logs

Log levels

The following log levels in ApsaraDB for OceanBase are listed based on severity in descending order: ERROR, USER_ERROR, WARN, INFO, TRACE, and DEBUG.

[Log levels and descriptions](#) describes each of the log levels.

Log levels and descriptions

Log level	Description
ERROR	The unexpected errors that must be manually handled.
USER_ERROR	The errors that are caused by invalid input data.
WARN	The expected errors that can be resolved by applications.

Log level	Description
INFO	The messages that describe the changes of the system status.
TRACE	The fine-grained debugging messages that allow you to trace the details of each request. For example, the execution of an SQL statement is divided into different stages and a trace log entry is generated for each stage.
DEBUG	The detailed debugging messages.

Log print settings

Logs are displayed by program module. Program modules consist of parent modules and child modules.

Examples:

- `SQL_OPT_LOG` records the information about the `SQL.OPT` child module. The `SQL_OPT_LOG` log entries are generated based on the log level settings of the `SQL.OPT` child module.
- `SQL_LOG` records the information about the SQL module. The `SQL_LOG` log entries are generated based on the log level settings of the SQL module.
- `OB_LOG` records the information about all the modules. The `OB_LOG` log entries are generated based on global settings of log levels.

Program logs divided into statement, session, and system logs.

Program logs are generated based on priorities in the following descending order: statement logs, session logs, and system logs.

Use the following methods to specify log print settings:

- System logs

- Method 1: Execute the `ALTER SYSTEM SET OB_LOG_LEVEL` statement

Execute the `ALTER SYSTEM SET OB_LOG_LEVEL` statement to change the log level of system logs.

Example:

```
alter system set ob_log_level='sql. *:debug, common. *:error';
```

`alter SYSTEM SET` allows you to specify the server or zone. For more information, see the `ALTER SYSTEM SET` syntax.

- Method 2: Run the `kill -41` and `kill -42` commands

You can run the `kill -41` command to increase the log level. For example, if the previous log level is `INFO`, you can run the `kill -41` command to change the log level to `TRACE`.

`kill` You can run the `kill -42` command to decrease the log level. For example, if the previous log level is `INFO`, you can run the `kill -42` command to change the log level to `WARN`.

`kill` You can run the `kill -41` command to increase the log level to the highest level: `DEBUG`. You can run the `kill -42` command to decrease the log level to the lowest level: `ERROR`.

We recommend that you use the first method. If the server updates the configuration file, the `ob_log_level` configurations in the file are imported. In this scenario, the `kill` commands fail to be run.

- Session logs

Run the `set @@ob_log_level='par_mod.sub_mod:level, par_mod.sub_mod:level'` command.

Example:

```
set @@ob_log_level='sql.*:debug, common.*:info';
```

- Statement logs

Use hints to set the log levels of statement logs.

Example:

```
select /*+log_level('debug')*/ * from t;
```

```
select /*+log_level('sql.*:debug, common.*:info')*/ * from t;
```

To view the log details in an easy way when you debug OBServers, you can execute the `ALTER SYSTEM SET ob_log_level='error'` statement to set the log level to `error`. You can run the `set @@ob_log_level='debug'` command to specify the log level of the session log as `debug`. If you execute a statement to retrieve data from multiple OBServers, the global settings of log levels apply to each of the OBServers.

19.Data Transmission Service (DTS)

19.1. What is DTS?

Data Transmission Service (DTS) is a data service that is provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases and big data systems.

Features

DTS has the following advantages over traditional data migration and synchronization tools: high compatibility, high performance, security, reliability, and ease of use. DTS allows you to simplify data transmission and focus on business development.

Feature	Description
Data migration	You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature applies to the following scenarios: data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out.
Data synchronization	You can use DTS to synchronize data between data sources. This feature applies to the following scenarios: disaster recovery, data backup, load balancing, cloud BI systems, and real-time data warehousing.
Change tracking	You can use DTS to track data changes from user-created MySQL databases, ApsaraDB RDS for MySQL instances, Cloud Native Distributed Database PolarDB-X instances (formerly known as DRDS), and user-created Oracle databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations.

19.2. Log on to the DTS console

This topic describes how to log on to the Data Transmission Service (DTS) console. Google Chrome is used in this example.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.

4. In the top navigation bar, choose **Products > Data Transmission Service**.
5. On the **DTS** page, select an **organization** and **region**, and then click **DTS**.

 **Note** If the top navigation bar is enabled for the DTS console, you can select the organization and resource set in the top navigation bar after you log on to the DTS console.

19.3. Data migration

19.3.1. Database and migration types

You can use Data Transmission Service (DTS) to migrate data between homogeneous and heterogeneous data sources. Typical scenarios include data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out. This topic describes the database types, database versions, and migration types that are supported by data migration.

Source database	Destination database	Migration type
<ul style="list-style-type: none"> • User-created MySQL database Versions 5.1, 5.5, 5.6, and 5.7 • RDS MySQL All versions 	User-created MySQL database Versions 5.1, 5.5, 5.6, and 5.7	<ul style="list-style-type: none"> • Schema migration • Full data migration • Incremental data migration
	RDS MySQL Versions 5.6 and 5.7	<ul style="list-style-type: none"> • Schema migration • Full data migration • Incremental data migration
	Cloud Native Distributed Database PolarDB-X (formerly known as DRDS) All versions	<ul style="list-style-type: none"> • Full data migration • Incremental data migration
	User-created Oracle database (RAC or non-RAC architecture) Versions 9i, 10g, 11g, and 12c	<ul style="list-style-type: none"> • Full data migration • Incremental data migration

Source database	Destination database	Migration type
<p>User-created SQL Server database</p> <p>Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAG). If the version of the source database is 2005, incremental data migration is not supported. </div>	<ul style="list-style-type: none"> User-created SQL Server database <p>Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> Note DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAG).</p> </div> <ul style="list-style-type: none"> RDS SQL Server <p>Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016</p>	<ul style="list-style-type: none"> Schema migration Full data migration Incremental data migration
<p>User-created Oracle database (RAC or non-RAC architecture)</p> <p>Versions 9i, 10g, 11g, and 12c</p>	<p>User-created Oracle database (RAC or non-RAC architecture)</p> <p>Versions 9i, 10g, 11g, and 12c</p>	<ul style="list-style-type: none"> Schema migration Full data migration Incremental data migration
	<p>PolarDB (formerly known as ApsaraDB RDS for PPAS)</p> <p>Version 11</p>	<ul style="list-style-type: none"> Schema migration Full data migration Incremental data migration
	<p>User-created MySQL database</p> <p>Versions 5.1, 5.5, 5.6, and 5.7</p>	<ul style="list-style-type: none"> Schema migration Full data migration Incremental data migration
	<p>RDS MySQL</p> <p>Versions 5.6 and 5.7</p>	<ul style="list-style-type: none"> Schema migration Full data migration Incremental data migration
	<p>Cloud Native Distributed Database PolarDB-X</p> <p>All versions</p>	<ul style="list-style-type: none"> Full data migration Incremental data migration

Source database	Destination database	Migration type
	AnalyticDB for MySQL Versions 2.0 and 3.0	<ul style="list-style-type: none"> • Schema migration • Full data migration • Incremental data migration
<ul style="list-style-type: none"> • User-created PostgreSQL database Versions 9.4, 9.5, 9.6, and 10.x • RDS PostgreSQL Versions 9.4 and 10 	<ul style="list-style-type: none"> • User-created PostgreSQL database Versions 9.4, 9.5, 9.6, and 10.x • RDS PostgreSQL Versions 9.4 and 10 	<ul style="list-style-type: none"> • Schema migration • Full data migration • Incremental data migration

19.3.2. Create a data migration instance

Before you configure a task to migrate data, you must create a data migration instance. This topic describes how to create a data migration instance in the Data Transmission Service (DTS) console.

Procedure

1. [Log on to the DTS console.](#)
2. In the left-side navigation pane, click **Data Migration**.
3. In the upper-right corner, click **Create Migration Task**.
4. In the Create DTS Instances dialog box, select a region, and enter the number of data migration instances that you want to create.

 **Note** In the Create DTS Instances dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

5. Click **Create**.

19.3.3. Configure data migration tasks

19.3.3.1. Migrate data between ApsaraDB RDS for PostgreSQL instances

instances

This topic describes how to migrate data between ApsaraDB RDS for PostgreSQL instances by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

The network type of both the source and destination ApsaraDB RDS for PostgreSQL instances is VPC.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise,

the destination database may contain duplicate data records.

- To ensure that the delay time of incremental data migration is accurate, DTS adds a heartbeat table named *dts_postgres_heartbeat* to the source database.
- During incremental data migration, DTS creates a replication slot for the source database. The replication slot is prefixed with *dts_sync_*. DTS automatically clears historical replication slots every 90 minutes to prevent them from occupying disk space.

 **Note** If a migration task is released or fails, DTS automatically clears the replication slot. After a switchover between primary and secondary ApsaraDB RDS for PostgreSQL databases, you must log on to the secondary database to clear the replication slot.



- If a data migration task fails, DTS automatically resumes the task. Before you switch your workloads to the destination instance, stop or release the data migration task. Otherwise, the data in the source instance will overwrite the data in the destination instance after the task is resumed.

Limits

- A single data migration task can migrate data from only one database. To migrate data from multiple databases, you must create a data migration task for each database.
- During incremental data migration, DTS synchronizes only data manipulation language (DML) operations. DML operations include INSERT, DELETE, and UPDATE.

Data migration process

The following table describes how DTS migrates the schemas and data of the source PostgreSQL database. The process avoids data migration failures that are caused by dependencies between objects.

Data migration process	Description
1. Schema migration	DTS migrates the schemas of tables, views, sequences, functions, user-defined types, rules, domains, operations, and aggregates to the destination database.  Note DTS does not migrate functions that are written in the C programming language.
2. Full data migration	DTS migrates historical data of the required objects to the destination database.
3. Schema migration	DTS migrates the schemas of triggers and foreign keys to the destination database.
4. Incremental data migration	DTS migrates incremental data of the required objects to the destination database.  Note Incremental data migration does not support the BIT data type.

Before you begin

If you need to perform incremental data migration, you must change the value of the *wal_level* parameter to *logical* for the source RDS instance.

Warning After you change the value of the `wal_level` parameter, you must restart the instance to apply the change. We recommend that you evaluate the impact on your business and change the parameter setting during off-peak hours.

Procedure

1. Create a data migration instance.
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the **Actions** column.
3. Configure the source and destination databases.

1. Configure Source and Destination
2. Configure Migration Types and
3. Advanced Settings
4. Precheck

* Task Name:

Source Database

* Instance Type: DTS support type

* Instance Region:

* RDS Instance ID: RDS Instances of Other Apsara Stack Accounts

* Database Name:

* Database Account:

* Database Password:

Destination Database

* Instance Type:

* Instance Region:

* RDS Instance ID:

* Database Name:

* Database Account:

* Database Password:

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select RDS instance.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the source RDS instance.
	Database Name	Enter the name of the source database.

Section	Parameter	Description
	Database Account	Enter the database account of the source RDS instance. The account must have the read and write permissions on the source database.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select RDS instance.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination RDS instance.
	Database Name	Enter the name of the destination database. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> ? Note The name of the destination database can be different from the name of the source database. </div>
	Database Account	Enter the database account of the destination RDS instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the migration types and the objects to be migrated.

1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

*** Migration Types:** Schema Migration Full Data Migration Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Available

If you search globally, please expand the

- public
- testschema
 - Tables
 - Views
 - Sequences
 - Functions
 - User Defined Types
 - Rules
 - Domains
 - Operations
 - Aggregates
 - Extensions

[Select All](#)

>

<

Selected (To edit an object name or its filter, hover over the object and click Edit.) [Learn more.](#)

- testschema (10Objects)
 - customer

[Remove All](#)

***Name batch change:** No Yes

Information:
 1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
 2. DDL operations are not supported during data migration because this can cause migration failures.

Cancel Previous Save Precheck

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Schema Migration and Full Data Migration. To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <p> Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p> Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

 **Note** If **Incremental Data Migration** is not selected, wait until the migration task is completed. If **Incremental Data Migration** is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.2. Migrate incremental data from a user-created SQL Server database to an ApsaraDB RDS for SQL Server instance

This topic describes how to migrate incremental data from a user-created SQL Server database to an ApsaraDB RDS for SQL Server instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created SQL Server database is 2005, 2008, 2008 R2, 2012, 2014, or 2016.

 Note

- DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAG).
- If the version of the source database is 2005, incremental data migration is not supported.
- If you migrate data between different versions of databases, make sure that the database versions are compatible.

- The tables to be migrated from the user-created SQL Server database must have primary keys or UNIQUE NOT NULL indexes.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- To ensure that the incremental data migration task runs as expected, do not frequently back up the source database. We recommend that you retain log files for more than three days. Otherwise, you cannot retrieve log files after they are truncated.
- To ensure that the delay time of incremental data migration is accurate, DTS adds a heartbeat table to the user-created SQL Server database. The name of the heartbeat table is `Source table name_dts_mysql_heartbeat`.
- DTS automatically creates a destination database in the ApsaraDB RDS for SQL Server instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for SQL Server instance before you configure the data migration task.
- If a data migration task fails, DTS automatically resumes the task. Before you switch your workloads to the destination instance, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination instance after the task is resumed.

Limits

- A single data migration task can migrate incremental data from only one database. To migrate incremental data from multiple databases, you must create a data migration task for each database.
- DTS cannot migrate the schemas of assemblies, service brokers, full-text indexes, full-text catalogs, distributed schemas, distributed functions, CLR stored procedures, CLR scalar-valued functions, CLR table-valued functions, internal tables, systems, or aggregate functions.
- DTS cannot migrate data of the `sql_variant` type.
- DTS cannot migrate tables that contain computed columns.

Migration types

- Schema migration

DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, synonym, SQL stored procedure, SQL function, plan guide, user-defined type, rule, and default.

- Full data migration

DTS migrates historical data of the required objects from the user-created SQL Server database to the destination ApsaraDB RDS for SQL Server instance.

- Incremental data migration

After full data migration is complete, DTS migrates incremental data from the user-created SQL Server database to the destination ApsaraDB RDS for SQL Server instance. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created SQL Server database to Alibaba Cloud.

SQL operations that can be synchronized during incremental data migration

- INSERT, UPDATE, and DELETE

 **Note** DTS cannot synchronize the UPDATE operations that update only the large fields.

- CREATE TABLE

 **Note** If a CREATE TABLE operation creates a partition table or a table that contains functions, DTS does not synchronize the CREATE TABLE operation.

- ALTER TABLE operations, including only ADD COLUMN, DROP COLUMN, and RENAME COLUMN
- DROP TABLE
- RENAME TABLE, TRUNCATE TABLE, and CREATE INDEX

Migration process

To avoid data migration failures caused by dependencies among objects, DTS migrates the schemas and data from the source SQL Server database in the following order:

1. Migrate the schemas of tables, views, synonyms, user-defined types, rules, defaults, and plan guides.
2. Perform full data migration.
3. Migrate the schemas of SQL stored procedures, SQL functions, triggers, and foreign keys.
4. Perform incremental data migration.

 **Note** During schema migration and full data migration, do not perform data definition language (DDL) operations on the required objects. Otherwise, the objects may fail to be migrated.

Before you begin

Before you configure a data migration task, configure log settings on the user-created SQL Server database.

 **Note** Skip this step if you do not need to perform incremental data migration.

1. Run the following command in the user-created SQL Server database to change the recovery mode to full mode:

```
use master;
GO
ALTER DATABASE <database_name> SET RECOVERY FULL WITH ROLLBACK IMMEDIATE;
GO
```

Parameters:

<database_name>: the name of the source database.

Example:

```
use master;
GO
ALTER DATABASE mytestdata SET RECOVERY FULL WITH ROLLBACK IMMEDIATE;
GO
```

2. Run the following command to create a logical backup for the source database. Skip this step if you have already created a logical backup.

```
BACKUP DATABASE <database_name> TO DISK='<physical_backup_device_name>';
GO
```

Parameters:

- <database_name>: the name of the source database.
- <physical_backup_device_name>: the storage path and file name of the backup file.

Example:

```
BACKUP DATABASE mytestdata TO DISK='D:\backup\dbdata.bak';  
GO
```

3. Run the following command to back up the log entries of the source database:

```
BACKUP LOG <database_name> to DISK='<physical_backup_device_name>' WITH init;  
GO
```

Parameters:

- <database_name>: the name of the source database.
- <physical_backup_device_name>: the storage path and file name of the backup file.

Example:

```
BACKUP LOG mytestdata TO DISK='D:\backup\dblog.bak' WITH init;  
GO
```

Procedure

1. Create a data migration instance.
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the **Actions** column.
3. Configure the source and destination databases.

The screenshot shows a configuration window for a data migration task. At the top, there are four tabs: "1. Configure Source and Destination" (active), "2. Configure Migration Types and Objects", "3. Map name modification", and "4. Precheck".

Task Name: SQL Server

Source Database:

- Instance Type: User-Created Database with Public IP Address
- Instance Region: cn-qingdao
- Database Type: SQLServer
- Hostname or IP Address: [Redacted]
- Port Number: 1433
- Database Account: dtstest
- Database Password: [Redacted]

Test Connectivity: Passed

Destination Database:

- Instance Type: RDS Instance
- Instance Region: cn-qingdao
- RDS Instance ID: [Redacted]
- Database Account: dtstest
- Database Password: [Redacted]

Test Connectivity: Passed

Buttons: Cancel, Set Whitelist and Next

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on where the source database is deployed. In this example, select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select SQL Server .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created SQL Server database. In this example, enter the public IP address.
	Port Number	Enter the service port number of the user-created SQL Server database. The default port number is 1433.
	Database Account	Enter the account that is used to log on to the user-created SQL Server database. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p> Note To perform incremental data migration, the account must have the sysadmin permission. To perform schema migration or full data migration, the account must have the SELECT permission on the objects that you want to migrate.</p> </div>
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select RDS Instance .
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination ApsaraDB RDS for SQL Server instance.
	Database Account	Enter the database account of the destination ApsaraDB RDS for SQL Server instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration types and the objects to be migrated.

1.Configure Source and Destination 2.Configure Migration Types and Objects 3.Map name modification 4.Precheck

* Migration Types: Schema Migration Full Data Migration Incremental Data Migration

Available

If you search globally, please expand the

- mytestdata
 - Tables
 - Views
 - Synonyms
 - Procedures
 - Functions
 - Types
 - Rules
 - Defaults
 - Plan Guides

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) [Learn more.](#)

- mytestdata (2Objects)
 - dbo.customer

Remove All

*Name batch change: No Yes

Information:
 1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
 2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description
Migration Types	<ul style="list-style-type: none"> ○ To perform only full data migration, select Schema Migration and Full Data Migration. ○ To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <div style="background-color: #e0f2f1; padding: 10px; margin-top: 10px;"> <p>? Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p> </div>

Parameter	Description
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p>Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

Note If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until The migration task is not delayed appears and manually stop the task.

19.3.3.3. Migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, or 5.7.
- If you need to perform incremental data migration, you must enable the binary logging feature. The following requirements must be met:
 - The value of the `binlog_format` parameter is set to `row`.
 - The value of the `binlog_row_image` parameter is set to `full`.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the FLOAT or DOUBLE data type. If you do not specify a precision, DTS sets the precision for the FLOAT data type to 38 digits and the precision for the DOUBLE data type to 308 digits. You must check whether the precision settings meet your business requirements.
- DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.

Migration types

- Schema migration

DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function.

Note

- During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
- DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- Full data migration

DTS migrates historical data of the required objects from the user-created MySQL database to the destination database in the ApsaraDB RDS for MySQL instance.

Note

During full data migration, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source database.

- Incremental data migration

After full data migration is complete, DTS retrieves binary log files from the user-created MySQL database. Then, DTS synchronizes incremental data from the user-created MySQL database to the destination ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created MySQL database to Alibaba Cloud.

SQL operations that can be synchronized during incremental data migration

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	<ul style="list-style-type: none"> • ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Procedure

1. Create a data migration instance.
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name:

Source Database

* Instance Type:

* Instance Region:

* Database Type:

* Hostname or IP Address:

* Port Number:

* Database Account:

* Database Password:

Test Connectivity ✔ Passed

Destination Database

* Instance Type:

* Instance Region:

* RDS Instance ID:

* Database Account:

* Database Password:

Test Connectivity ✔ Passed

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select MySQL .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created MySQL database.
	Port Number	Enter the service port number of the user-created MySQL database. The default port number is 3306.
	Database Account	Enter the account of the user-created MySQL database. To perform incremental data migration, the account must have the SELECT permission on the required objects, the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, and the SHOW VIEW permission. To perform schema migration or full data migration, the account must have the SELECT permission on the required objects.

Section	Parameter	Description
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select RDS Instance.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination RDS instance.
	Database Account	Enter the database account of the destination RDS instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

- In the lower-right corner of the page, click **Set Whitelist and Next**.
- Select the migration types and the objects to be migrated.

1. Configure Source and Destination
2. Configure Migration Types and Objects
3. Advanced Settings
4. Precheck

* Task Name:

Source Database

* Instance Type:

* Instance Region: [Get IP Address Segment of DTS](#)

* Database Type:

* Hostname or IP Address:

* Port Number:

* Database Account:

* Database Password: ✔ Passed

Destination Database

* Instance Type:

* Instance Region:

* RDS Instance ID:

* Database Account:

* Database Password: ✔ Passed

Parameter	Description

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Schema Migration and Full Data Migration. To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <p>Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p>Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

Note If **Incremental Data Migration** is not selected, wait until the migration task is completed. If **Incremental Data Migration** is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.4. Migrate data from a user-created MySQL database to a Cloud Native Distributed Database PolarDB-X instance

This topic describes how to migrate data from a user-created MySQL database to a Cloud Native Distributed Database PolarDB-X instance by using Data Transmission Service (DTS). Cloud Native Distributed Database PolarDB-X is formerly known as Distributed Relational Database Service (DRDS).

Precautions

- DTS cannot migrate schemas from a user-created MySQL database to a Cloud Native Distributed Database PolarDB-X instance.

 **Note** During schema migration, DTS migrates the schemas of the required objects, such as tables, from the source database to the destination database.

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the FLOAT or DOUBLE data type. If you do not specify a precision, DTS sets the precision for the FLOAT data type to 38 digits and the precision for the DOUBLE data type to 308 digits. You must check whether the precision settings meet your business requirements.

SQL operations that can be synchronized during incremental data migration

INSERT, UPDATE, DELETE, and REPLACE

Permissions required for database accounts

Database	Full data migration	Incremental data migration
User-created MySQL database	The SELECT permission	The REPLICATION SLAVE, REPLICATION CLIENT, SHOW VIEW, and SELECT permissions
Cloud Native Distributed Database PolarDB-X	The read and write permissions	The read and write permissions

Procedure

1. Create databases and tables in the destination PolarDB-X instance based on the schemas of the source tables.
2. [Create a data migration instance.](#)
3. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the Actions column.
4. Configure the source and destination databases.

1. Configure Source and Destination Databases
2. Configure Migration Types and Objects
3. Map name modification
4. Precheck

* Task Name:

Source Database

* Instance Type:

* Instance Region:

* Database Type:

* Hostname or IP Address:

* Port Number:

* Database Account:

* Database Password:

Test Connectivity ✔ Passed

Destination Database

* Instance Type:

* Instance Region:

* DRDS Instance ID:

* Database Name:

* Database Account:

* Database Password:

Test Connectivity ✔ Passed

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select MySQL .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created MySQL database.
	Port Number	Enter the service port number of the user-created MySQL database. The default port number is 3306.
	Database Account	Enter the account of the user-created MySQL database. For more information about the permissions that are required for the account, see Permissions required for database accounts .
	Database Password	Enter the password of the source database account.
	Instance Type	Select DRDS Instance .
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the Cloud Native Distributed Database PolarDB-X instance.

Destination Database	Parameter	Description
	Database Name	Enter the name of the destination database.
	Database Account	Enter the database account of the Cloud Native Distributed Database PolarDB-X instance. For more information about the permissions that are required for the account, see Permissions required for database accounts .
	Database Password	Enter the password of the destination database account.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.

6. Select the migration types and the objects to be migrated.

* Migration Types: Full Data Migration Incremental Data Migration

Data migration applies to short-term migration scenarios. Typical scenarios include migrating data to the cloud, scaling and sharding databases, and migrating data between Apsara Stack databases.
For long-term data synchronization in real time, use the data synchronization feature.

Available

If you search globally, please expand the

- data123
- mysqltest
- mysqltestnew
- sys

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) [Learn more](#).

- mysqltest (2Objects)
 - customer
 - vipinfo

Remove All

*Name batch change: No Yes

Information:

- Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
- DDL operations are not supported during data migration because this can cause migration failures.

Cancel Previous Save **Precheck**

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Full Data Migration. To ensure service continuity during data migration, select Full Data Migration and Incremental Data Migration. <p>Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>

Parameter	Description
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> ○ You can select columns, tables, or databases as the objects to be migrated. ○ After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated. </div>

7. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

 **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until The migration task is not delayed appears and manually stop the task.

19.3.3.5. Migrate data from an ApsaraDB RDS for MySQL instance to a user-created Oracle database

This topic describes how to migrate data from an ApsaraDB RDS for MySQL instance to a user-created Oracle database by using Data Transmission Service (DTS). DTS supports full data migration and incremental data migration. You can select these migration types to ensure service continuity of the source ApsaraDB RDS for MySQL instance during data migration.

Prerequisites

The destination Oracle database is created. The schema of the Oracle database is the same as the schema of the source database in the ApsaraDB RDS for MySQL instance. This is because DTS does not support schema migration from an ApsaraDB RDS for MySQL instance to a user-created Oracle database.

Permissions required for database accounts

Migration type	Full data migration	Incremental data migration
Source ApsaraDB RDS for MySQL instance	The read and write permissions	The read and write permissions
Destination Oracle database	The read and write permissions	The read and write permissions

Procedure

1. **Create a data migration instance.**
2. In the migration task list, find the instance that you created, and click **Configure Migration Task** in the Actions column.
3. **Optional.** Enter a name for the task.

DTS automatically generates a name for each task. Duplicate task names are allowed. You can edit the task name based on your needs. We recommend that you specify an informative name for easy identification.

4. **Configure the source and destination databases.** The following table describes the parameters.

* Task Name:

Source Database

* Instance Type:

* Instance Region:

* RDS Instance ID:

* Database Account:

* Database Password:

* Encryption: Non-encrypted SSL-encrypted

Destination Database

* Instance Type:

* Instance Region:

* Database Type:

* Hostname or IP Address:

* Port Number:

* Instance Type: Non-RAC Instance RAC or PDB Instance

* SID:

* Database Account:

* Database Password:

Section	Parameter	Description
Source Database	Instance Type	Select RDS Instance as the type of the source instance.
	Instance Region	The region where the source instance resides.
	RDS Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter an account that has the read and write permissions on the source database.
	Database Password	Enter the password of the source database account.
	Instance Type	Select User-Created Database with Public IP Address as the type of the destination database.
	Instance Region	The region where the destination instance resides.
	Database Type	Select Oracle.

Section	Parameter	Description
Destination Database	Hostname or IP Address	Enter the endpoint that is used to connect to the Oracle database.
	Port Number	The default port number is 1521.
	Instance Type	<ul style="list-style-type: none"> ◦ Non-RAC Instance: If you select this option, you must specify the SID. ◦ RAC Instance: If you select this option, you must specify the Service Name.
	Database Account	Enter an account that has the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

5. Click **Test Connectivity** and confirm that the test results for both the source and destination databases are **Passed**.
6. In the lower-right corner of the page, click **Set Whitelist and Next**.
7. Select the migration types based on your needs. Select objects from the **Available** section and click the  icon to move the objects to the **Selected** section.
 - To ensure service continuity during data migration, select **Full Data Migration and Incremental Data Migration**.
 - To perform only full data migration, select **Full Data Migration**.
8. Click **Precheck** and wait until the precheck is complete.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Click **Next** to start the migration task.

 **Note** If **Incremental Data Migration** is not selected, wait until the migration task is completed. If **Incremental Data Migration** is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.6. Migrate data between user-created Oracle databases

This topic describes how to migrate data between user-created Oracle databases by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the source Oracle database is 9i, 10g, 11g, or 12c.

 **Note** To ensure compatibility, make sure the versions of the source and destination databases are the same.

- Supplemental logging, including `SUPPLEMENTAL_LOG_DATA_PK` and `SUPPLEMENTAL_LOG_DATA_UI`, is enabled for the source Oracle database. For more information, see [Supplemental Logging](#).
- The `ARCHIVELOG` mode is enabled for the source Oracle database. Archived log files are accessible and a

suitable retention period is set for archived log files. For more information, see [Managing Archived Redo Log Files](#).

- The available storage space of the destination Oracle database is larger than the total size of the data in the source Oracle database.

Migration types

Migration type	Description
Schema migration	<p>DTS migrates the schemas of the required objects to the destination Oracle database. DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.</p> <p>Note If an object contains triggers, the data between the source and destination databases will become inconsistent.</p>
Full data migration	<p>DTS migrates historical data of the required objects from the source Oracle database to the destination Oracle database.</p> <p>Note During schema migration and full data migration, do not perform DDL operations on the objects to be migrated. Otherwise, the objects may fail to be migrated.</p>
Incremental data migration	<p>After full data migration, DTS retrieves redo log files from the source Oracle database. Then, DTS synchronizes incremental data from the source Oracle database to the destination Oracle database. Incremental data migration allows you to ensure service continuity when you migrate data between Oracle databases.</p>

SQL operations that can be synchronized during incremental data migration

- INSERT, UPDATE, and DELETE operations
- CREATE TABLE operations

Note The CREATE TABLE operations to create partition tables or tables that contain functions cannot be synchronized.

- ALTER TABLE, DROP TABLE, RENAME TABLE, CREATE INDEX, and ADD INDEX operations

Permissions required for database accounts

Database	Schema migration	Full data migration	Incremental data migration
Source Oracle database	The owner permission on schemas	The owner permission on schemas	SYSDBA
Destination Oracle database	The owner permission on schemas	The owner permission on schemas	The owner permission on schemas

Note For more information about how to create and authorize an Oracle database account, see [CREATE USER](#) and [GRANT](#).

Procedure

1. [Create a data migration instance](#).
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the

Actions column.

3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select Oracle .
	Hostname or IP Address	Enter the endpoint that is used to connect to the source Oracle database.
	Port Number	Enter the port number of the source Oracle database. The default port number is 1521.
	Instance Type	Select Non-RAC Instance or RAC Instance based on the architecture of the source Oracle database.
	SID	Enter the system ID (SID) of the destination Oracle database. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> ? Note This parameter is required if you select Non-RAC Instance as the instance type. </div>
	Service Name	Enter the server name of the instance. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> ? Note This parameter is required if you select RAC Instance as the instance type. </div>
	Database Account	Enter the account that is used to connect to the source Oracle database.
	Database Password	Enter the password of the source database account.
	Instance Type	Select User-Created Database with Public IP Address .
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select Oracle .
	Hostname or IP Address	Enter the endpoint that is used to connect to the destination Oracle database.
	Port Number	Enter the port number of the destination Oracle database. The default port number is 1521.
	Instance Type	Select Non-RAC Instance or RAC Instance based on the architecture of the destination Oracle database.

Destination Section Database	Parameter	Description
	SID	Enter the SID of the destination Oracle database. Note This parameter is required if you select Non-RAC Instance as the instance type.
	Service Name	Enter the server name of the instance. Note This parameter is required if you select RAC Instance as the instance type.
	Database Account	Enter the account that is used to connect to the destination Oracle database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the migration types and the objects to be migrated.

1. Configure Source and Destination | **2. Configure Migration Types and Objects** | 3. Map name modification | 4. Precheck

* Migration Types: Schema Migration Full Data Migration Incremental Data Migration **Note:** Incremental migration does not support trigger synchronization. For details, please [Reference Document](#)

Available

If you search globally, please expand the

- EOA_USER
- DTSTEST
 - Tables
 - Views
- SCOTT
- OWBSYS_AUDIT
- OWBSYS
- APEX_030200
- APEX_PUBLIC_USER
- SPATIAL_CSW_ADMIN_USR
- SPATIAL_WFS_ADMIN_USR
- ORDDATA
- XS\$NULL
- APPQOSSYS

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) [Learn more.](#)

- DTSTEST (10 objects)
 - ORACLETESTTABLE

Remove All

*Name batch change: No Yes

Information:

- Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
- DDL operations are not supported during data migration because this can cause migration failures.

Cancel Previous Save **Precheck**

Parameter	Description
-----------	-------------

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Schema Migration and Full Data Migration. To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <p>Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p>Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

Note If **Incremental Data Migration** is not selected, wait until the migration task is completed. If **Incremental Data Migration** is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.7. Migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. When you migrate data from a user-created Oracle database, you can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, or 12c.
- Supplemental logging, including `SUPPLEMENTAL_LOG_DATA_PK` and `SUPPLEMENTAL_LOG_DATA_UI`, is enabled for the user-created Oracle database. For more information, see [Supplemental Logging](#).

- The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see [Managing Archived Redo Log Files](#).

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Table names in the ApsaraDB RDS for MySQL instance are case-insensitive. If a table name in the user-created Oracle database contains uppercase letters, ApsaraDB RDS for MySQL converts all uppercase letters to lowercase letters before creating the table.

If the source Oracle database contains identical table names that differ only in capitalization, these table names are identified as duplicate. During schema migration, the following message is returned: The object already exists. To avoid name conflicts in the destination database, you can change the names of the migrated objects by using the object name mapping feature. For more information, see [Object name mapping](#).

Migration types

- Schema migration

DTS supports schema migration for tables and indexes. DTS does not support schema migration for the following types of objects: view, synonym, trigger, stored procedure, function, package, and user-defined type. DTS has the following limits on schema migration for tables and indexes:

- Schema migration of nested tables is not supported. Clustered tables and index-organized tables (IOTs) are converted into common tables in the destination database.
- Schema migration of function-based indexes, domain indexes, bitmap indexes, and reverse indexes is not supported.

- Full data migration

DTS migrates historical data of the required objects from the user-created Oracle database to the destination ApsaraDB RDS for MySQL instance.

- Incremental data migration

DTS retrieves redo log files from the user-created Oracle database. Then, DTS synchronizes incremental data from the user-created Oracle database to the destination database in the ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from the user-created Oracle database to the destination database.

SQL operations that can be synchronized during incremental data migration

- INSERT, DELETE, and UPDATE
- CREATE TABLE

 **Note** If a CREATE TABLE operation creates a table that contains functions, DTS does not synchronize the CREATE TABLE operation.

- ALTER TABLE, ADD COLUMN, DROP COLUMN, RENAME COLUMN, and ADD INDEX
- DROP TABLE
- RENAME TABLE, TRUNCATE TABLE, and CREATE INDEX

Data type mappings

For more information, see [Data type mappings between heterogeneous databases](#).

Procedure

1. Create a data migration instance.
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on where the source database is deployed. In this example, select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select Oracle .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is 1521.
	Instance Type	<ul style="list-style-type: none"> ◦ Non-RAC Instance: If you select this option, you must specify the SID. ◦ RAC Instance: If you select this option, you must specify the Service Name.
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select RDS Instance .
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination RDS instance.
	Database Account	Enter the database account of the destination RDS instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Select the migration types and the objects to be migrated.

Parameter	Description
-----------	-------------

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Schema Migration and Full Data Migration. To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <p> Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p> Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

 **Note** If **Incremental Data Migration** is not selected, wait until the migration task is completed. If **Incremental Data Migration** is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.8. Migrate data from a user-created Oracle database to a PolarDB cluster

This topic describes how to migrate data from a user-created Oracle database to a PolarDB cluster by using Data Transmission Service (DTS). PolarDB is formerly known as ApsaraDB RDS for PPAS. DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, or 12c.
- Supplemental logging, including `SUPPLEMENTAL_LOG_DATA_PK` and `SUPPLEMENTAL_LOG_DATA_UI`, is enabled for the user-created Oracle database. For more information, see [Supplemental Logging](#).
- The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see [Managing Archived Redo Log](#)

Files.**Precautions**

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- DTS supports schema migration for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.

Limits

- During schema migration, the reverse indexes and bitmap indexes of the source database are stored as common indexes in the PolarDB cluster.
- During schema migration, partitioned indexes are converted into independent indexes on each partition in the PolarDB cluster.
- Incremental data migration supports only tables that have primary keys or UNIQUE NOT NULL indexes.
- Incremental data migration does not support the LONG data type.
- Data definition language (DDL) operations that are performed during incremental data migration cannot be synchronized to the destination database.
- Materialized views cannot be migrated.

Data type mappings

For more information, see [Data type mappings between heterogeneous databases](#).

Procedure

1. [Create a data migration instance](#).
2. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the Actions column.
3. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on where the source database is deployed. In this example, select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select Oracle .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is 1521.
	Instance Type	<ul style="list-style-type: none"> ◦ Non-RAC Instance: If you select this option, you must specify the SID. ◦ RAC Instance: If you select this option, you must specify the Service Name.

Section	Parameter	Description
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select RDS Instance.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the destination RDS instance.
	Database Name	Enter the name of the destination database.
	Database Account	Enter the database account of the destination RDS instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the migration types and the objects to be migrated.

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Schema Migration and Full Data Migration. To ensure service continuity during data migration, select Schema Migration, Full Data Migration, and Incremental Data Migration. <p> Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p> Note</p> <ul style="list-style-type: none"> You can select columns, tables, or databases as the objects to be migrated. After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

6. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. After the task passes the precheck, click **Next**.

 **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until **The migration task is not delayed** appears and manually stop the task.

19.3.3.9. Migrate data from a user-created Oracle database to a Cloud Native Distributed Database PolarDB-X instance

This topic describes how to migrate data from a user-created Oracle database to a Cloud Native Distributed Database PolarDB-X instance by using Data Transmission Service (DTS). Cloud Native Distributed Database PolarDB-X is formerly known as Distributed Relational Database Service (DRDS). DTS allows you to ensure service continuity when you migrate both historical data and incremental data.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, or 12c.
- Supplemental logging, including `SUPPLEMENTAL_LOG_DATA_PK` and `SUPPLEMENTAL_LOG_DATA_UI`, is enabled for the user-created Oracle database. For more information, see [Supplemental Logging](#).
- The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see [Managing Archived Redo Log Files](#).

Precautions

- DTS cannot migrate schemas from a user-created Oracle database to a Cloud Native Distributed Database PolarDB-X instance.

 **Note** During schema migration, DTS migrates the schemas of the required objects, such as tables, from the source database to the destination database.

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

Migration types

- Full data migration

DTS migrates historical data of the required objects from the source Oracle database to the destination database.

- Incremental data migration

After full data migration is complete, DTS retrieves redo log files from the source Oracle database. Then, DTS synchronizes incremental data from the source Oracle database to the destination database.

Note The following SQL operations can be synchronized during incremental data migration: INSERT, DELETE, and UPDATE operations. Data definition language (DDL) operations cannot be synchronized during incremental data migration.

Procedure

1. Create databases and tables in the destination PolarDB-X instance based on the schemas of the source tables.

Note The data types of Oracle databases and PolarDB-X instances do not have one-to-one correspondence. You must create the corresponding data types in PolarDB-X instances. For more information, see [Data type mappings between heterogeneous databases](#).

2. Create a data migration instance.
3. In the migration task list, find the migration task that you created, and click **Configure Migration Task** in the Actions column.
4. Configure the source and destination databases.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select User-Created Database with Public IP Address .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	Database Type	Select Oracle .
	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created Oracle database. In this example, enter the public IP address.
	Port Number	Enter the service port number of the user-created Oracle database. The default port number is 1521.
	Instance Type	<ul style="list-style-type: none"> ◦ Non-RAC Instance: If you select this option, you must specify the SID. ◦ RAC Instance: If you select this option, you must specify the Service Name.
	Database Account	Enter the account of the user-created Oracle database. To perform incremental data migration, the account must have the database administrator (DBA) permission. To perform schema migration or full data migration, the account must have the owner permission on schemas.
	Database Password	Enter the password of the source database account.
Destination Database	Instance Type	Select DRDS Instance .
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data migration instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the destination Cloud Native Distributed Database PolarDB-X instance.

Section	Parameter	Description
	Database Account	Enter the database account of the destination Cloud Native Distributed Database PolarDB-X instance. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

5. In the lower-right corner of the page, click **Set Whitelist and Next**.

6. Select the migration types and the objects to be migrated.

The screenshot shows the '2. Configure Migration Types and Objects' step in the DTS console. At the top, there are four progress indicators: '1. Configure Source and Destination', '2. Configure Migration Types and Objects' (active), '3. Map name modification', and '4. Precheck'. Below the progress indicators, there are two checked options for 'Migration Types': 'Full Data Migration' and 'Incremental Data Migration'. A note explains that data migration applies to short-term scenarios and that long-term synchronization should use data synchronization. The main area is split into two panels: 'Available' on the left and 'Selected' on the right. The 'Available' panel shows a search bar and a list of databases: 'data123', 'mysqltest', 'mysqltestnew', and 'sys'. The 'Selected' panel shows a search bar and a list of objects: 'mysqltest (20Objects)', 'customer', and 'vipinfo'. There are navigation arrows between the panels and 'Select All'/'Remove All' buttons at the bottom of each. At the bottom of the interface, there are buttons for 'Cancel', 'Previous', 'Save', and 'Precheck'. A 'Name batch change' section has 'No' selected. An 'Information' section contains two points: 1. Data migration only copies data and schema. 2. DDL operations are not supported during data migration.

Parameter	Description
Migration Types	<ul style="list-style-type: none"> To perform only full data migration, select Full Data Migration. To ensure service continuity during data migration, select Full Data Migration and Incremental Data Migration. <p>Note If Incremental Data Migration is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.</p>

Parameter	Description
Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <div style="background-color: #e0f2f7; padding: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ○ You can select columns, tables, or databases as the objects to be migrated. ○ After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the destination database. For more information, see Object name mapping. ○ If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated. </div>

7. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. After the task passes the precheck, click **Next**.

 **Note** If Incremental Data Migration is not selected, wait until the migration task is completed. If Incremental Data Migration is selected, the migration task does not automatically stop. In this case, you must wait until The migration task is not delayed appears and manually stop the task.

19.3.4. Manage data migration tasks

19.3.4.1. Object name mapping

Data Transmission Service (DTS) provides the object name mapping feature. You can use this feature to change the names of one or more objects that are migrated to the destination instance. This topic describes how to use the object name mapping feature when you configure a data migration task.

Limits

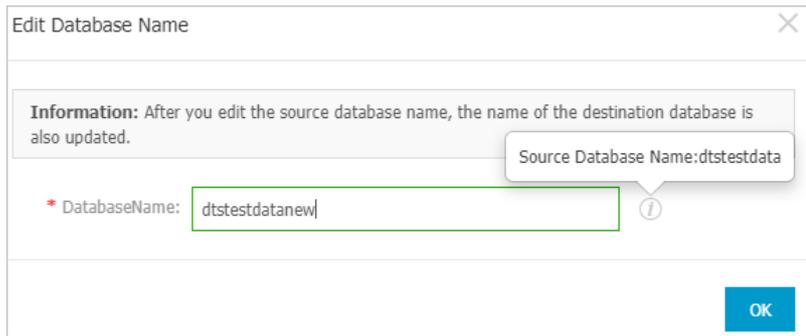
You can use the object name mapping feature only when a data migration task is configured and the current step is **Configure Migration Types and Objects**.

 **Note** Do not use the object name mapping feature after a data migration task is started. Otherwise, data may fail to be migrated.

Procedure

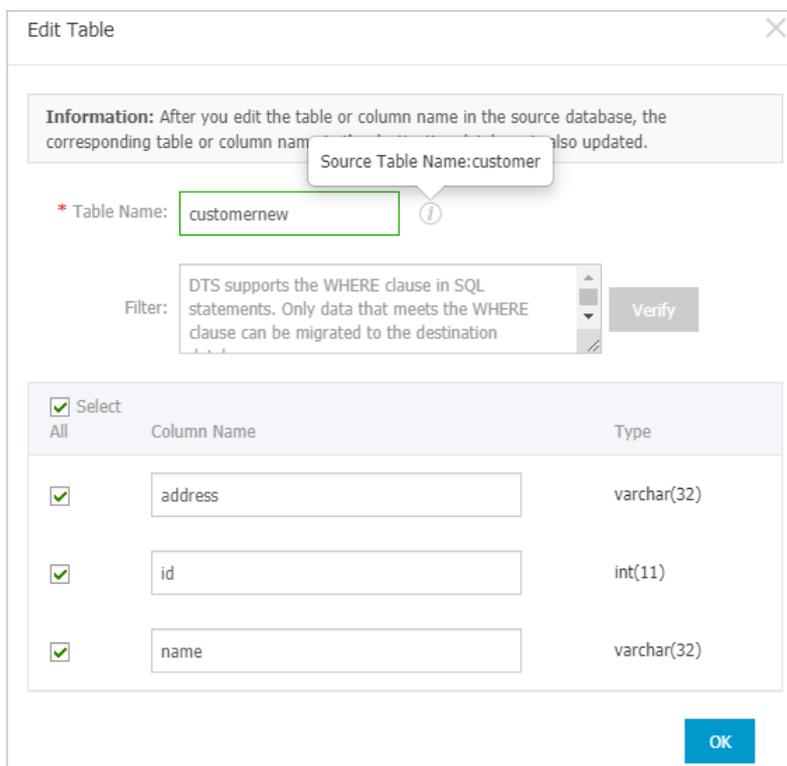
1. In the **Configure Migration Types and Objects** step, move the required objects to the **Selected** section, move the pointer over a database or table, and then click **Edit**.
2. In the dialog box that appears, specify a name for the object in the destination instance.
 - Database name mapping

In the **Edit Database Name** dialog box that appears, enter the database name that you want to use in the destination instance.



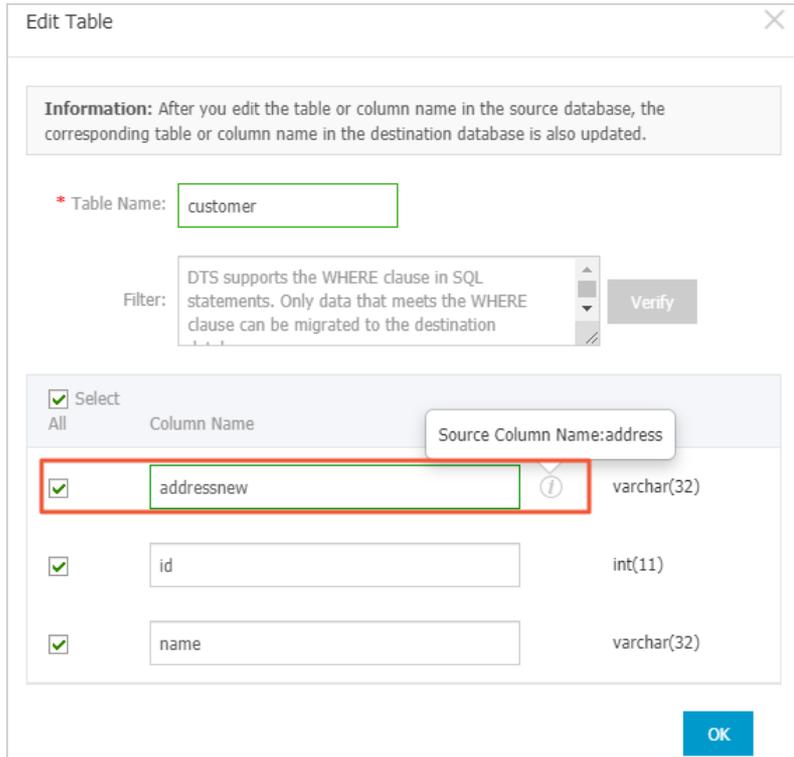
- **Table name mapping**

In the **Edit Table** dialog box that appears, enter the table name that you want to use in the destination instance.



- **Column name mapping**

In the **Edit Table** dialog box that appears, enter a new name for each column.



Note In this step, you can clear the options of columns that do not need to be synchronized.

3. Click **OK**.
4. Configure other parameters that are required for the data migration task.

19.3.4.2. Specify an SQL condition to filter data

This topic describes how to specify an SQL condition to filter the data of a specific table when you configure a data migration task.

The SQL condition takes effect only within the table that you select. DTS migrates only the data that meets the SQL condition to the destination database. This feature is applicable to scenarios such as regular data migration and table partitioning.

Limits

An SQL condition applies only to full data migration. If you select incremental data migration as the migration type, the SQL condition does not filter incremental data.

Specify an SQL condition

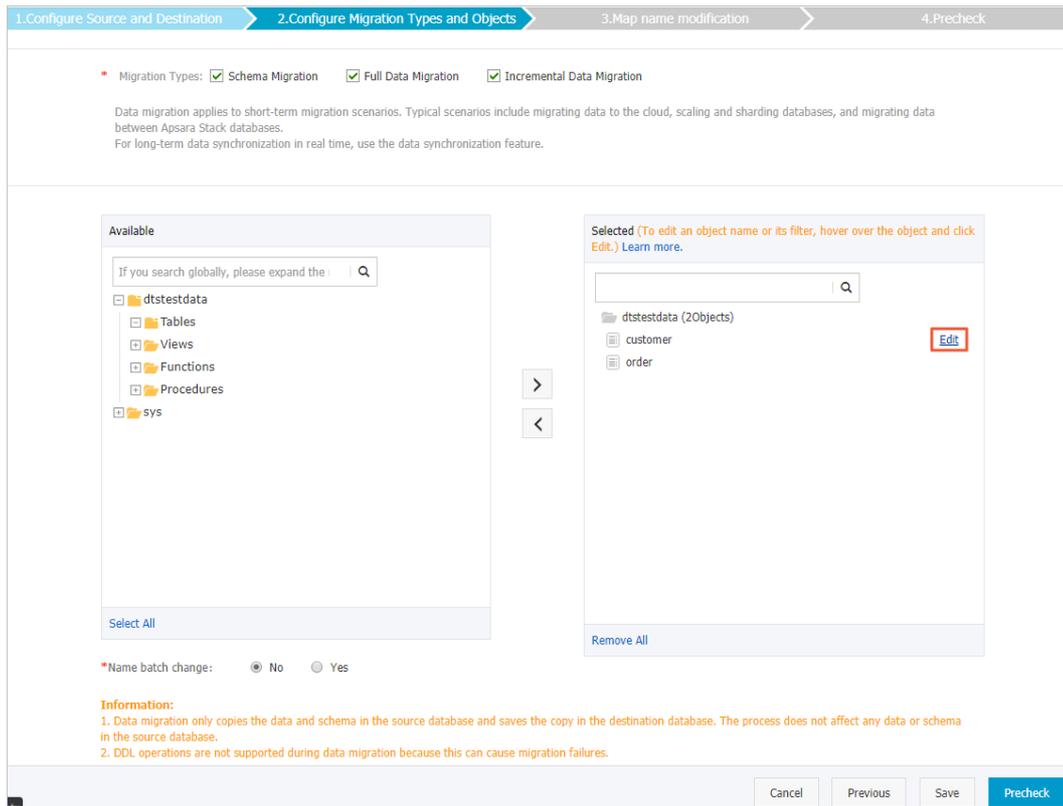
You can specify an SQL condition in the **Configure Migration Types and Objects** step when you configure a data migration task.

To filter the data of a specific table by using an SQL condition, you must select the table as the object that you want to migrate. You cannot select a database as the object. To specify an SQL condition, perform the following steps.

Procedure

1. In the **Configure Migration Types and Objects** step, move the pointer over a table in the **Selected** section. The **Edit** button appears, as shown in **Edit button**.

Edit button



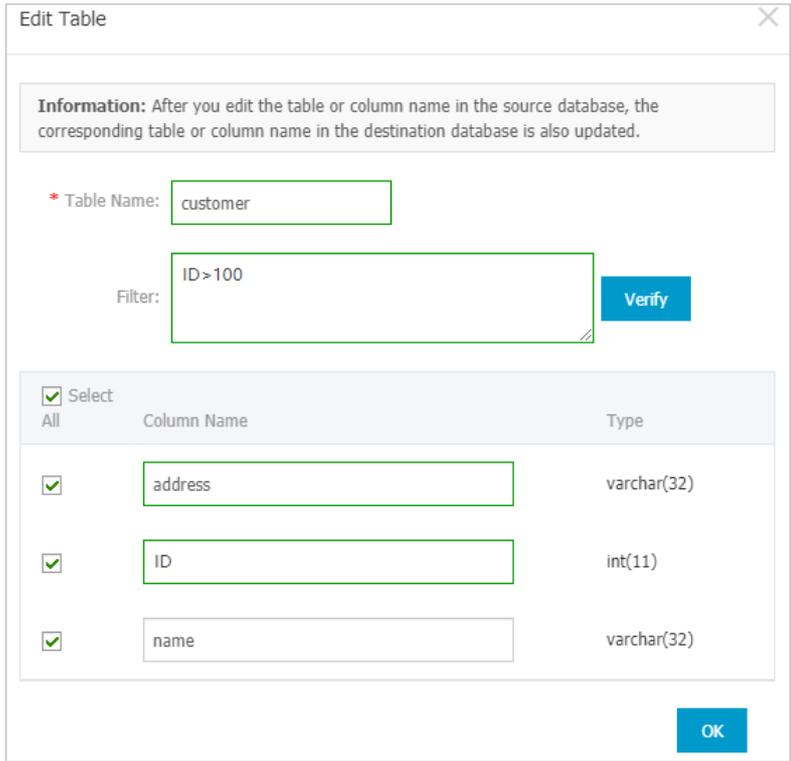
2. Click Edit. The Edit Table dialog box appears.

Modify an SQL condition

The SQL conditions in DTS are the same as the standard SQL WHERE conditions for databases. You can use SQL conditions to perform operations and run basic functions.

Enter an SQL condition in the text box. For example, you can enter `id>1000` to migrate the records whose IDs are greater than 1,000 to the destination instance, as shown in [Modify an SQL condition](#).

Modify an SQL condition



After the SQL condition is specified, click OK.

19.3.4.3. Troubleshoot a failed data migration task

This topic describes how to troubleshoot a failed data migration task. You can use this feature if your data migration task is in the Migration Failed state during schema migration or full data migration.

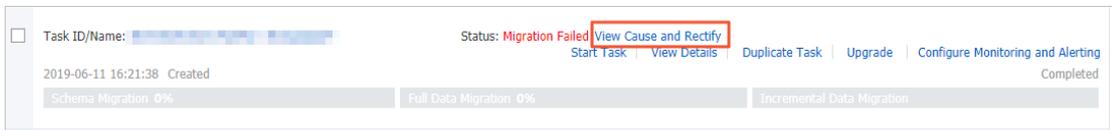
Troubleshoot a failed task during schema migration

DTS supports data migration between heterogeneous data sources. However, if you migrate data of unsupported types to the destination instance during schema migration, the task may fail.

1. [Log on to the DTS console.](#)
2. In the left-side navigation pane, click **Data Migration**.
3. Use one of the following methods to troubleshoot the failed task:

- o **Method 1**

- a. Find the task and click **View Cause and Rectify**.



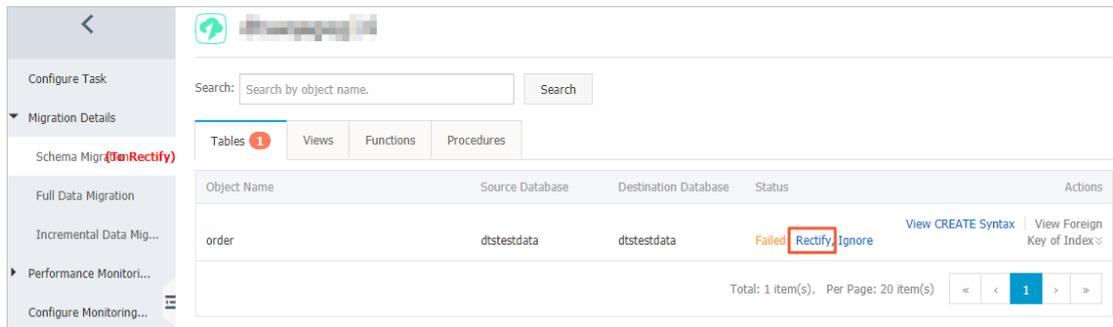
- b. Troubleshoot the issue based on the cause that is displayed in the View Cause and Rectify message. For example, you can troubleshoot an issue by modifying the schema syntax.

- c. Click **Restart Task**.

- o **Method 2**

- a. Click the instance ID or task name.
- b. In the left-side navigation pane, choose **Migration Details > Schema Migration**.

- c. On the **Schema Migration** page, find the object that causes the migration failure and click **Rectify** in the **Status** column.



- d. Troubleshoot the issue based on the cause that is displayed in the **Rectify** dialog box. For example, you can troubleshoot an issue by modifying the schema syntax.

- e. Click **Rectify**.

Note

- If the failure persists, the **Rectify** dialog box does not close and shows the failure cause. You must continue troubleshooting based on the failure cause until the troubleshooting is successful.
- If the troubleshooting is successful, the **Schema Migration** page appears and the status of the object changes to **Finished**.

4. If no objects are in the **Failed** state, DTS proceeds with the data migration task, for example, entering the full data migration process.

Troubleshoot a failed task during full data migration

1. **Log on to the DTS console.**
2. In the left-side navigation pane, click **Data Migration**.
3. Find the task and click **View Cause and Rectify**.

DTS allows you to troubleshoot a task that fails during full data migration due to the following reasons.

Note If a task fails during full data migration due to other reasons, DTS provides only the **Ignore** option. The object that causes the failure is not migrated to the destination database.

- The connection to the source or destination database failed or timed out.
Troubleshoot the issue, make sure that the connection is successful, and then click **Restart Task**.
 - The storage space of the destination instance is insufficient or the instance is locked.
Upgrade the specification of the destination instance or clear the log space, and then click **Restart Task**.
 - MyISAM tables in the source database are corrupted.
Troubleshoot the issue in the source database, and then click **Restart Task**.
4. In the dialog box that appears, troubleshoot the issue based on the failure cause.
 5. Click **Restart Task**.

19.3.5. Precheck items

19.3.5.1. Source database connectivity

DTS checks whether DTS servers can connect to the source database. DTS creates a connection to the source database by using the JDBC protocol. If the connection fails, the data migration task fails to pass the connectivity check.

The migration task may fail to pass the connectivity check because of the following reasons:

- The database account or password that is specified in the data migration task is invalid.

Troubleshooting:

Find a server that can connect to the source database. On the server, enter the database account and password that are specified in the data migration task to check whether the account and password are valid. If the database account or password is invalid, the following error message is displayed: Access deny.

Solution:

Log on to the DTS console, modify the database account and password, and then run a precheck again.

- The DTS servers are disallowed to access the source database.

Troubleshooting:

- Find a server that can connect to the source database. On the server, enter the database account and password that are specified in the data migration task to check whether the connection is successful. Only authorized DTS servers can connect to the source database. If the CIDR block of a DTS server is not included in the whitelist of the source database, the DTS server cannot connect to the source database.
- If the source database is a MySQL database, use a MySQL client to connect to the database and run the `SELECT HOST FROM mysql.user WHERE user='Account', password='Password';` command. If the CIDR blocks of DTS servers are not included in the whitelist of the source database, the query result of the preceding command is not %.

Solution:

- If the source database is a MySQL database, run the `GRANT ALL ON . TO 'Account'@'%' IDENTIFIED BY 'Password';` command to authorize the database account. Replace Account and Password in the preceding command with your database account and password. After the account is authorized, run a precheck again.
- A firewall is configured on the source database server.

Troubleshooting: If the server where the source database resides runs Linux, run the `iptables -L` command in the shell to check whether a firewall is configured for the server. If the server where the source database resides runs Windows, find Windows Defender Firewall from the Control Panel and check whether a firewall is configured for the server.

Solution:

Disable the firewall and run a precheck again.

- The network between DTS servers and the source database is unavailable.

If the failure persists, you can check whether the network between DTS servers and the source database is available. In this case, we recommend that you contact Alibaba Cloud engineers by submitting a ticket.

19.3.5.2. Check the destination database connectivity

This check item checks whether the DTS server can connect to the destination database for migration. DTS creates a connection to the destination database by using the JDBC protocol. If the connection fails, the check item fails.

The destination database connectivity precheck may fail for the following reasons:

- An incorrect account or password is provided when a migration task is created.

Diagnostics:

On any network-ready server that can connect to the destination database, use the account and password specified for creating the migration task to connect to the destination database through client software. Check whether the connection succeeds. If an error is reported for the connection and the error message contains Access deny, the account or password is incorrect.

Troubleshooting:

Modify the migration task in the DTS console, correct the account and password, and perform the precheck again.

- There is no connectivity between the DTS server and destination database.

If you check that the password and account are correct, the check item may fail because there is no connectivity between the DTS server and the destination database. In this case, contact the DTS engineers on duty.

19.3.5.3. Binary logging configurations of the source database

Whether binary logging is enabled in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether binary logging is enabled in the source database. If binary logging is disabled in the source database, the check result is Failed.

Troubleshooting: Run the `log_bin=mysql_bin` command to modify the configuration file of the source database. Restart the source database and run a precheck again.

Binary log format of the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the binary log format is set to ROW in the source database. If the binary log format is not set to ROW in the source database, the check result is Failed.

Troubleshooting: Run the `set global binlog_format=ROW` command in the source database and run a precheck again. We recommend that you restart the MySQL process. Otherwise, data loss may occur because sessions will continue to be written in a non-ROW mode.

Binary log files in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether specific binary log files are removed from the source database. If binary log files in the source database are incomplete, the check result is Failed.

Troubleshooting: Run the `PURGE BINARY LOGS TO 'The name of the first binary log file that is not deleted'` command in the source database and run a precheck again.

To find the binary log files that are removed from the source database, click the info icon next to the failed item. In the View Details dialog box, the names of deleted binary log files are displayed.

Parameter binlog_row_image of the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the value of the binlog_row_image parameter in the source database is set to FULL. This parameter indicates whether the full image is recorded. If the full image is not recorded in binary log files of the source database, the check result is Failed.

Troubleshooting: Run the `set global binlog_row_image=FULL` command in the source database and run a precheck again.

19.3.5.4. Integrity of the FOREIGN KEY constraints

DTS checks whether the parent table on which a child table depends is included in the selected objects. The precheck allows DTS to protect the integrity of the FOREIGN KEY constraints.

If the parent table on which a child table depends is not included in the selected objects, the check result is Failed.

Troubleshooting:

- Do not migrate the child tables that cause the check failure. To do this, remove these child tables from the selected objects and run a precheck again.

- Migrate the parent tables rather than the child tables. To do this, add the parent tables to the selected objects and run a precheck again.
- Delete the foreign key dependencies between the parent and child tables in the source database and run a precheck again.

19.3.5.5. Existence of FEDERATED tables

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the source database contains storage engines that are not supported by incremental data migration. Incremental data migration does not support the FEDERATED and MRG_MyISAM storage engines.

If the FEDERATED storage engine is used by specific tables in the source database, the check result is Failed.

If the MRG_MyISAM storage engine is used by specific tables in the source database, the check result is Failed.

Solution:

Remove the tables that use the FEDERATED or MRG_MyISAM storage engine from the selected objects. Then, create a separate migration task to perform schema migration and full data migration for these tables.

19.3.5.6. Permissions

Source database permissions

DTS checks whether the account of the source database has the required permissions to perform data migration. For information about the permissions that are required by each type of database, see the topics about how to configure data migration tasks.

Destination database permissions

DTS checks whether the account of the destination database has the required permissions to perform data migration. For information about the permissions that are required by each type of database, see the topics about how to configure data migration tasks.

19.3.5.7. Object name conflict

This check item checks for duplicate object names in the destination and source database. If this check item fails, an object in the destination RDS instance has the same name as an object to be migrated. This causes the migration to fail.

When this check item fails, an error message is displayed indicating that an object in the destination database has the same name as an object to be migrated from the source database.

Troubleshooting:

- Use the database and table name mapping feature provided by DTS to migrate the object to be migrated to another object with a different name in the destination database.
- In the destination database, delete or rename the object that has the same name as the object to be migrated.
- Modify the migration task and delete that object to be migrated from the list of objects to be migrated. Do not migrate this object.

19.3.5.8. Schema existence

This check item checks whether the database to be migrated exists in the destination RDS instance. If no, DTS creates one automatically. However, under the following circumstances, the automatic database creation fails, and this check item prompts a failure:

- The database name contains characters other than lowercase letters, digits, underscores (_), and hyphens (-).

The cause of the precheck failure is that the name of the source database does not comply with the requirements of RDS.

Troubleshooting: On the database management page of the RDS console, create a database that complies with the requirements of RDS and grant the migration account the read and write permissions on the new database. Use the database name mapping feature provided by DTS to map the source database to the new database. Then, perform the precheck again.

- The character set of the database is not UTF8, GBK, Latin1, or UTF-8MB4.

The cause of the precheck failure is that the character set of the source database does not comply with the requirements of RDS.

Troubleshooting: On the database management page of the RDS console, create a database that complies with the requirements of RDS and grant the migration account the read and write permissions on the new database. If the new database and the database to be migrated have different names, you can use the database name mapping feature of DTS to map the database to be migrated to the new database. Then re-run the precheck.

- The migration account of the destination database has no read and write permissions on the database to be migrated.

The cause of the precheck failure is that you are not authorized to operate on the source database.

Troubleshooting: On the database management page of the RDS console, click the Account Management tab. Grant the migration account the read and write permissions on the source database. Then, perform the precheck again.

19.3.5.9. Value of server_id in the source database

This item is checked only when you migrate incremental data between MySQL databases. DTS checks whether the value of server-id in the source database is set to an integer greater than 1.

If the check result is Failed, run the `set global server_id='An integer greater than 1'` command in the source database and perform a precheck again.

19.3.5.10. Source database version

DTS checks whether the version of the source database is supported. The table [Source database types and versions](#) lists the source database versions that are supported by DTS.

Source database types and versions

Source database type	Supported version
MySQL	5.0, 5.1, 5.5, 5.6, and 5.7. Only 5.1, 5.5, 5.6, and 5.7 are supported for incremental data migration.

If the check result is Failed, you must upgrade or downgrade the source database to a supported version before you perform a precheck again.

19.3.6. Data type mappings between heterogeneous databases

Heterogeneous databases support different data types. During schema migration, Data Transmission Service (DTS) converts the data types of the source database into those of the destination database. This topic lists the data type mappings for you to evaluate the impact of data migration on your business.

Data migration from a user-created Oracle database to a user-created MySQL database or an ApsaraDB RDS for MySQL instance

Data type in the Oracle database	Data type in the MySQL database	Supported by DTS
varchar2(n [char/byte])	varchar(n)	Yes
nvarchar2[(n)]	national varchar[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	national char[(n)]	Yes
number[(p[,s])]	decimal[(p[,s])]	Yes
float(p)	double	Yes
long	longtext	Yes
date	datetime	Yes
binary_float	decimal(65,8)	Yes
binary_double	double	Yes
timestamp[(fractional_seconds_precision)]	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
clob	longtext	Yes
nclob	longtext	Yes
blob	longblob	Yes
raw	varbinary(2000)	Yes
long raw	longblob	Yes
bfile	N/A	No
interval year(year_precision) to month	N/A	No
interval day(day_precision)to second[(fractional_seconds_precision)]	N/A	No

 Note

- A char column with a length greater than 255 bytes is converted to the varchar(n) type.
- Data types such as bfile, interval year to month, and interval day to second in Oracle databases are not supported in MySQL databases. They cannot be converted to data types supported by the destination database during schema migration.

The schema migration fails if the table to be migrated contains these three data types. You must make sure that columns with these three data types are excluded from the objects to be migrated.

- The timestamp data type of MySQL databases does not contain the time zone information. However, the timestamp with time zone and timestamp with local time zone data types in Oracle databases provide time zone information. Therefore, DTS converts the values of these data types based on the time zone to UTC time for storage in the destination instance.

Data migration from a user-created Oracle database to a PolarDB-X instance

Oracle data type	PolarDB-X data type	Supported by DTS
varchar2(n [char/byte])	varchar(n)	Yes
nvarchar2[(n)]	national varchar[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	national char[(n)]	Yes
number[(p,s)]	decimal[(p,s)]	Yes
float(p)	double	Yes
long	longtext	Yes
date	datetime	Yes
binary_float	decimal(65,8)	Yes
binary_double	double	Yes
timestamp[(fractional_seconds_precision)]	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with localtimezone	datetime[(fractional_seconds_precision)]	Yes
clob	longtext	Yes
nclob	longtext	Yes
blob	longblob	Yes
raw	varbinary(2000)	Yes
long raw	longblob	Yes
bfile	None	No
interval year(year_precision) to month	None	No

Oracle data type	PolarDB-X data type	Supported by DTS
interval day(day_precision)to second[(fractional_seconds_precision)]	None	No

 Note

- If a char field in the Oracle database is greater than 255 bytes in length, DTS converts this field to the varchar(n) type in the PolarDB-X instance.
- The timestamp data type of PolarDB-X does not contain the time zone information. However, the timestamp with time zone and timestamp with local time zone data types in Oracle databases provide the time zone information. Therefore, DTS converts the values of these data types into UTC time in the destination PolarDB-X instance.

Data migration from a user-created Oracle database to a PolarDB cluster

Oracle data type	PolarDB data type	Supported by DTS
varchar2(n [char/byte])	varchar2[(n)]	Yes
nvarchar2[(n)]	nvarchar2[(n)]	Yes
char[(n [byte/char])]	char[(n)]	Yes
nchar[(n)]	nchar[(n)]	Yes
number[(p,s)]	number[(p,s)]	Yes
float(p)	double precision	Yes
long	long	Yes
date	date	Yes
binary_float	real	Yes
binary_double	double precision	Yes
timestamp[(fractional_seconds_precision)]	timestamp[(fractional_seconds_precision)]	Yes
timestamp[(fractional_seconds_precision)]with time zone	timestamp[(fractional_seconds_precision)]with time zone	Yes
timestamp[(fractional_seconds_precision)]with local time zone	timestamp[(fractional_seconds_precision)]with time zone	Yes
clob	clob	Yes
nclob	nclob	Yes
blob	blob	Yes
raw	raw(size)	Yes
long raw	long raw	Yes
bfile	None	No

Oracle data type	PolarDB data type	Supported by DTS
interval year(year_precision) to month	interval year to month	No
interval day(day_precision) to second[(fractional_seconds_precision)]	interval day to second[(fractional_seconds_precision)]	No

 **Note** PolarDB does not support the timestamp[(fractional_seconds_precision)]with local time zone data type. DTS converts the data of this type into UTC time and then stores the data in the destination PolarDB cluster by using the timestamp[(fractional_seconds_precision)]with time zone data type.

19.4. Data synchronization

19.4.1. Database types, initial synchronization types, and synchronization topologies

You can use Data Transmission Service (DTS) to synchronize data between various data sources. This topic describes the database types, initial synchronization types, and synchronization topologies that are supported by DTS.

Source database	Destination database	Initial synchronization type	Synchronization topology
<ul style="list-style-type: none"> • User-created MySQL database 5.1, 5.5, 5.6, and 5.7 • RDS MySQL 5.6 and 5.7 	User-created MySQL database 5.1, 5.5, 5.6, and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	RDS MySQL 5.6 and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
	AnalyticDB for PostgreSQL 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization

Source database	Destination database	Initial synchronization type	Synchronization topology
	MaxCompute	Initial schema synchronization Initial full data synchronization	One-way synchronization
Cloud Native Distributed Database PolarDB-X (formerly known as DRDS)	Cloud Native Distributed Database PolarDB-X	Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

19.4.2. Create a data synchronization instance

Before you configure a task to synchronize data, you must create a data synchronization instance. This topic describes how to create a data synchronization instance in the Data Transmission Service (DTS) console.

Procedure

1. [Log on to the DTS console.](#)
2. In the left-side navigation pane, click **Data Synchronization**.
3. In the upper-right corner of the page, click **Create Synchronization Task**.
4. In the Create DTS Instances dialog box, set the required parameters.

Parameter	Description
Source Instance Region	Select the region where the source instance resides.
Source Instance Type	Select the type of the source instance. <ul style="list-style-type: none"> ◦ MySQL: a user-created MySQL database or an ApsaraDB RDS for MySQL instance ◦ Drds: a Cloud Native Distributed Database PolarDB-X instance (formerly known as DRDS)
Destination Instance Region	Select the region where the destination instance resides.
Destination Instance Type	Select the type of the destination instance. <ul style="list-style-type: none"> ◦ MySQL: a user-created MySQL database or an ApsaraDB RDS for MySQL instance ◦ AnalyticDB: AnalyticDB for MySQL ◦ MaxCompute ◦ DataHub ◦ Drds: a Cloud Native Distributed Database PolarDB-X instance (formerly known as DRDS) ◦ AnalyticDB for PostgreSQL: AnalyticDB for PostgreSQL

Parameter	Description
Synchronization Mode	Two-way synchronization is available only when you select MySQL as the type of both the source and destination instances.
Instances to Create	Set the number of data synchronization instances that you want to create at a time. The default value is 1.

Note In the Create DTS Instances dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

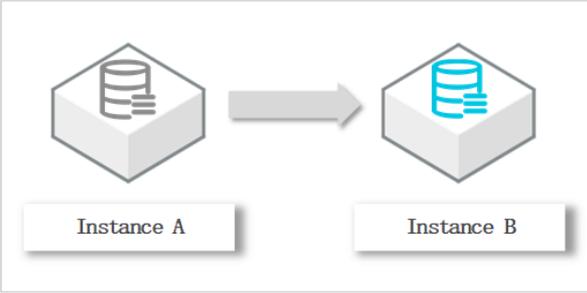
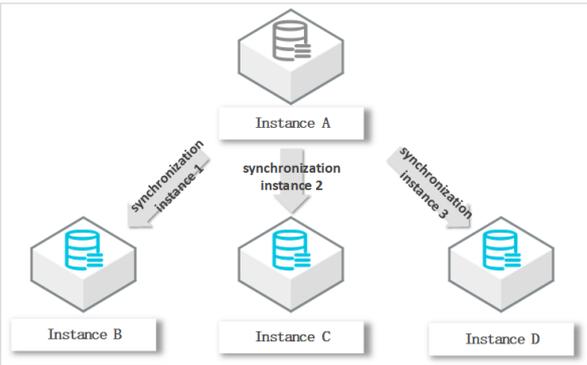
5. Click **Create**.

19.4.3. Synchronization topologies

The data synchronization feature supports multiple types of synchronization topologies. You can select a topology for your data synchronization instances based on your business requirements. This topic describes the synchronization topologies that are supported by DTS and how to use these topologies.

One-way synchronization

To ensure data consistency for one-way synchronization, we recommend that you perform only read operations on the objects in the destination instance. Do not modify the objects.

Topology type	Topology	Description
One-way one-to-one synchronization		None
One-way one-to-many synchronization		<p>You must purchase multiple synchronization instances to implement one-way one-to-many synchronization.</p> <p>For example, if you want to synchronize data from Instance A to Instance B, C, and D, you must purchase three synchronization instances.</p>

Topology type	Topology	Description
One-way cascade synchronization		<p>You must purchase multiple synchronization instances to implement one-way cascade synchronization.</p> <p>For example, if you want to synchronize data from Instance A to Instance B and then from Instance B to Instance C, you must purchase two synchronization instances.</p>
One-way many-to-one synchronization		<p>You must purchase multiple synchronization instances to implement one-way many-to-one synchronization.</p> <p>For example, if you want to synchronize data from Instance B, C, and D to Instance A, you must purchase three synchronization instances.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p>Note To ensure data consistency, you must select different objects for these synchronization instances.</p> </div>

Two-way synchronization

DTS supports two-way data synchronization only between two MySQL databases. DTS does not support two-way data synchronization between multiple MySQL databases.

Topology type	Topology	Description
Two-way one-to-one synchronization		<p>To ensure data consistency, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the instances.</p>

19.4.4. Configure data synchronization tasks

19.4.4.1. Configure data synchronization between ApsaraDB RDS for MySQL instances

This topic describes how to configure one-way data synchronization between ApsaraDB RDS for MySQL instances.

Prerequisites

The source and destination ApsaraDB RDS for MySQL instances are created.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If you select one or more tables (not a database) as the required objects, do not use `gh-ost` or `pt-online-schema-change` to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The tables to be migrated in the source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

DTS supports one-way synchronization and two-way synchronization. For more information, see [Synchronization topologies](#).

SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE
DDL	<ul style="list-style-type: none"> • ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE

Limits

- Incompatibility with triggers

If you select a database as the object and the database contains a trigger that updates a table, data inconsistency may occur. To solve this issue, you must delete the trigger in the destination database.

- Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if you select a table as the object and rename the table during data synchronization, the data of this table is not synchronized to the destination database. To avoid this situation, you can select the database to which this table belongs as the object when you configure the data synchronization task.

Procedure

1. Create a data synchronization instance.

Note When you create the data synchronization instance, set both Source Instance Type and Destination Instance Type to MySQL, and set Synchronization Mode to One-Way Synchronization.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.

3. Configure the source and destination instances.

1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Synchronization Task Name:

Source Instance Details

Instance Type:

Instance Region:

* Instance ID:

* Database Account:

* Database Password:

* Encryption: Non-encrypted SSL-encrypted

Destination Instance Details

Instance Type:

Instance Region:

* Instance ID:

* Database Account:

* Database Password:

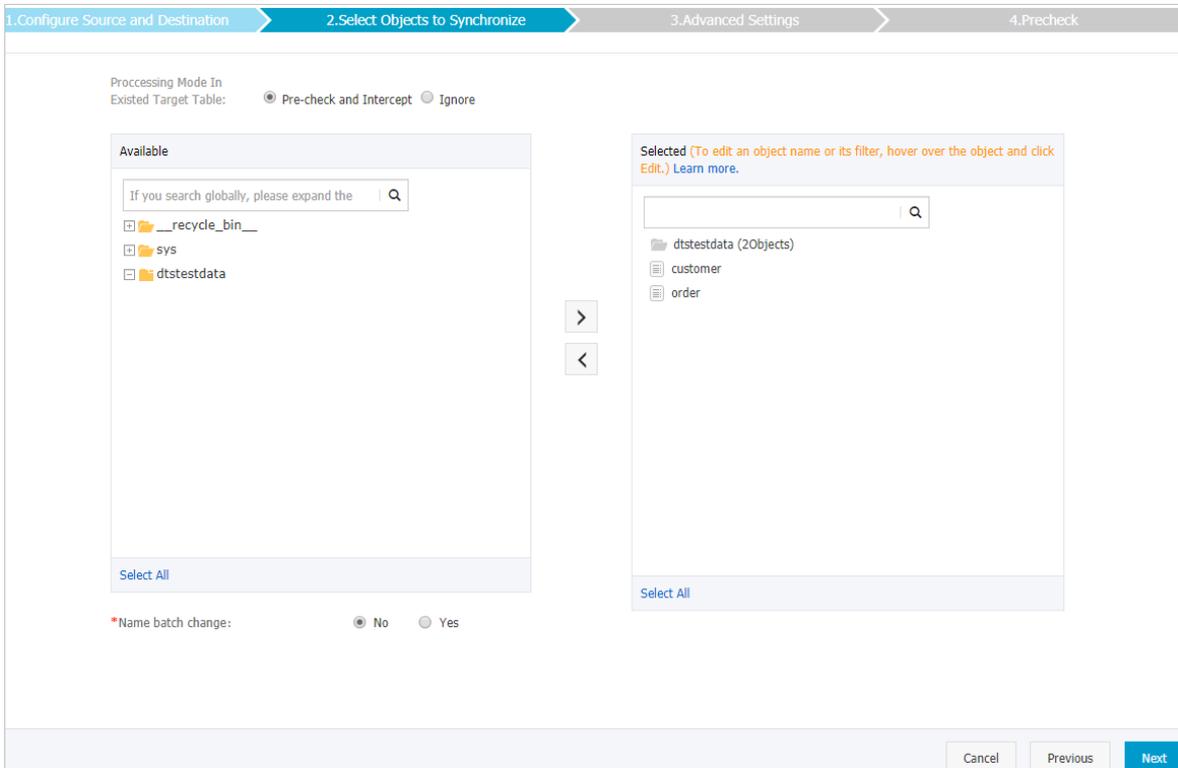
* Encryption: Non-encrypted SSL-encrypted

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select RDS Instance .
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc; margin-top: 5px;"> <p>Note If the database engine of the source RDS instance is MySQL 5.6, you do not need to configure the database account or database password.</p> </div>
	Database Password	Enter the password of the source database account.

Section	Parameter	Description
	Encryption	Select an encryption method. If you select SSL-encrypted , you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance Details	Instance Type	Select RDS Instance .
	Instance Region	The region of the source instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the destination RDS instance.
	Database Account	Enter the database account of the destination RDS instance. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p>? Note If the database engine of the destination RDS instance is MySQL 5.6, you do not need to configure the database account or database password.</p> </div>
	Database Password	Enter the password of the destination database account.
	Encryption	Select an encryption method. If you select SSL-encrypted , you must enable SSL encryption for the RDS instance before you configure the data synchronization task.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the processing mode of conflicting tables, and the objects that you want to synchronize.



Parameter	Description
Processing Mode In Existed Target Table	<ul style="list-style-type: none"> ◦ Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. ◦ Ignore: skips the precheck for identical table names in the source and destination databases. <div style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> <p> Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</p> <ul style="list-style-type: none"> ▪ DTS does not synchronize the data records that have the same primary keys as the data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. ▪ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails. </div>
Select Objects	<p>Select objects (tables or a database) from the Available section and click the  icon to move the objects to the Selected section.</p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #ccc;"> <p> Note</p> <ul style="list-style-type: none"> ◦ If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. ◦ After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance. </div>

6. In the lower-right corner of the page, click **Next**.

7. Configure initial synchronization.

1.Configure Source and Destination
2.Select Objects to Synchronize
3.Advanced Settings
4.Precheck

Initial Synchronization: Initial Schema Synchronization Initial Full Data Synchronization

 **Note** Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

8. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, DTS performs initial synchronization.

19.4.4.2. Synchronize data from an ApsaraDB RDS for MySQL instance to a MaxCompute project

MaxCompute (formerly known as ODPS) is a fast and fully managed computing platform for large-scale data warehousing. MaxCompute can process exabytes of data. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a MaxCompute project by using Data Transmission Service (DTS).

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If you select one or more tables (not a database) as the required objects, do not use `gh-ost` or `pt-online-schema-change` to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- You can select only tables as the objects to be synchronized.
- MaxCompute does not support the PRIMARY KEY constraint. If network errors occur, DTS may synchronize duplicate data records to MaxCompute.

SQL operations that can be synchronized

- Data definition language (DDL) operation: ADD COLUMN
- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

Synchronization process

1. Initial schema synchronization DTS synchronizes the schemas of the required objects from the source database to MaxCompute. During initial schema synchronization, DTS adds the `_base` suffix to the end of the source table name. For example, if the name of the source table is `customer`, the name of the table in MaxCompute is `customer_base`.
2. Initial full data synchronization DTS synchronizes the historical data of the table from the source database to the destination table in MaxCompute. For example, the `customer` table in the source database is synchronized to the `customer_base` table in MaxCompute. The data is the basis for subsequent incremental synchronization.

 **Note** The destination table that is suffixed with `_base` is known as a full baseline table.

3. Incremental data synchronization DTS creates an incremental data table in MaxCompute. The name of the incremental data table is suffixed with `_log`, for example, `customer_log`. Then, DTS synchronizes the incremental data that was generated in the source database to the incremental data table.

 **Note** For more information, see [Schema of an incremental data table](#).

Procedure

1. [Create a data synchronization instance](#).

Note When you create the data synchronization instance, set Source Instance Type to MySQL, set Destination Instance Type to MaxCompute, and set Synchronization Mode to One-Way Synchronization.

- Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
- Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select RDS Instance.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. Note If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.
	Database Password	Enter the password of the source database account.
Destination Instance Details	Encryption	Select Non-encrypted or SSL-encrypted. If you select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
	Instance Type	This parameter is set to MaxCompute and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Project	The name of the MaxCompute project.

- In the lower-right corner of the page, click **Set Whitelist and Next**.
- In the lower-right corner of the page, click **Next**. In this step, the permissions on the MaxCompute project are granted to the synchronization account.

1.Configure Source and Destination 2.Authorize MaxCompute Account 3.Select Objects to Synchronize 4.Precheck

To synchronize data to a MaxCompute instance, you must grant the following permissions of project dtstest to the synchronization account.

- CreateTable
- CreateInstance
- CreateResource
- CreateJob
- List

Cancel Previous **Next**

- Configure the synchronization policy and objects.

7. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**. Then, DTS performs initial synchronization.

Schema of an incremental data table

DTS synchronizes incremental data that is generated in the source MySQL database to the incremental data table in MaxCompute. The incremental data table stores incremental data and specific metadata. The following figure shows the schema of an incremental data table.

	A	B	C	D	E	F	G	H	I	J	K	L
	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
1	10000	2018-02-03 01:38:01		1560000333	U	156000655	Y	N	2019	08	16	16
2	10000	2018-02-03 01:38:01		1560000333	U	156000655	N	Y	2019	08	16	16
3	9999	2016-11-18 11:44:54		1560000419	D	156000845	Y	N	2019	08	16	16
4	10001	2018-12-23 05:11:59		1560000435	I	156000878	N	Y	2019	08	16	16

Note In the example, the `modifytime_year`, `modifytime_month`, `modifytime_day`, `modifytime_hour`, and `modifytime_minute` fields form the partition key.

The following table describes the schema of an incremental data table.

Field	Description
<code>record_id</code>	<p>The ID of the incremental log entry.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p>Note</p> <ul style="list-style-type: none"> The ID auto-increments for each new log entry. If an UPDATE operation is performed, DTS generates two incremental log entries for the operation. The two incremental log entries have the same record ID. </div>
<code>operation_flag</code>	<p>The operation type. Valid values:</p> <ul style="list-style-type: none"> I: an INSERT operation. D: a DELETE operation. U: an UPDATE operation.
<code>utc_timestamp</code>	The operation timestamp. It is also the timestamp of the binary log file. The timestamp is in the UTC format.
<code>before_flag</code>	Indicates whether the column values are pre-update values. Valid values: Y and N.
<code>after_flag</code>	Indicates whether the column values are post-update values. Valid values: Y and N.

Additional information about the `before_flag` and `after_flag` fields

For different operation types, the `before_flag` and `after_flag` fields of an incremental log entry are defined as follows:

- INSERT**

For an INSERT operation, the column values are the newly inserted record values (post-update values). The value of the `before_flag` field is N and the value of the `after_flag` field is Y.

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
5	10001	2018-12-23 05:11:59		156944878	I	15694878	N	Y	2019	08	16	16

• UPDATE

DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the record_id, operation_flag, and dts_utc_timestamp fields.

The second log entry records the pre-update values, so the value of the before_flag field is Y and the value of the after_flag field is N. The second log entry records the post-update values, so the value of the before_flag field is N and the value of the after_flag field is Y.

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
2	10000	2018-02-03 01:38:01		156940333	U	15694655	Y	N	2019	08	16	16
3	10000	2018-02-03 01:38:01		156940333	U	15694655	N	Y	2019	08	16	16

• DELETE

For a DELETE operation, the column values are the deleted record values (pre-update values). The value of the before_flag field is Y and the value of the after_flag field is N.

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
4	9999	2016-11-18 11:44:54		1569400419	D	15694845	Y	N	2019	08	16	16

Merge a full baseline table and incremental data table

After a data synchronization task is started, DTS creates a full baseline table and an incremental data table in MaxCompute. You can use SQL statements to merge the two tables. This allows you to obtain the full data at a specific time point.

This section describes how to merge data for the customer table. The following figure shows the schema of the customer table.

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	null	
2	register_time	timestamp	YES		null	
3	address	varchar(32)	YES		null	

1. Create a table in MaxCompute based on the schema of the source table. The table is used to store the merged data.

For example, you can obtain full data of the customer table at the 1565944878 time point. Run the following SQL statements to create the required table:

```
CREATE TABLE `customer_1565944878` (
  `id` bigint NULL,
  `register_time` datetime NULL,
  `address` string);
```

Note For more information about the data types that are supported by MaxCompute, see MaxCompute User Guide.

2. Run the following SQL statements in MaxCompute to merge the full baseline table and incremental data table and obtain full data at a specific time point:

```
set odps.sql.allow.fullscan=true;
insert overwrite table <result_storage_table>
select <col1>,
       <col2>,
       <colN>
from(
select row_number() over(partition by t.<primary_key_column>
order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag, <col1>, <col2>, <colN>
from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.<col1>, incr.<col2>,incr.<colN>
from <table_log> incr
where utc_timestamp< <timestamp>
union all
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.<col1>, base.<col2>,base.<colN>
from <table_base> base) t) gt
where record_num=1
and after_flag='Y'
```

 Note

- <result_storage_table>: the name of the table that stores the merged data.
- <col1>/<col2>/<colN>: the names of the columns in the table to be merged.
- <primary_key_column>: the name of the primary key column in the table to be merged.
- <table_log>: the name of the incremental data table.
- <table_base>: the name of the full baseline table.
- <timestamp>: the timestamp that is generated when full data is obtained.

Run the following SQL statements to obtain full data of the customer table at the 1565944878 time point:

```

set odps.sql.allow.fullscan=true;
insert overwrite table customer_1565944878
select id,
       register_time,
       address
from(
select row_number() over(partition by t.id
order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag, id, register_time, address
from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.id, incr.register_time, incr.address
from customer_log incr
where utc_timestamp< 1565944878
union all
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.id, base.register_time, base.address
from customer_base base) t) gt
where gt.row_number= 1
and gt.after_flag= 'Y';
    
```

3. Query the merged data from the customer_1565944878 table.

	A	B	C
1	id	register_time	address
2	1	2017-12-09 14:00:12	
3	2	2017-11-16 21:17:39	
4	3	2019-01-29 07:56:20	

19.4.4.3. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service that is developed by Alibaba Cloud for online data analysis with high concurrency. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). After you synchronize data, you can use AnalyticDB for MySQL to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- The tables that you want to synchronize from the ApsaraDB RDS for MySQL instance contain primary keys.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- If the disk space usage of nodes in an AnalyticDB for MySQL cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects to be synchronized. You must ensure that the destination cluster has sufficient storage space.

SQL operations that can be synchronized

- Data definition language (DDL) operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN
- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

 **Note** We recommend that you do not change the data type of fields in the source table during data synchronization. Otherwise, DTS generates an error message and stops the data synchronization task.

Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. The following table lists the data types that DTS can convert.

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
BIGINT UNSIGNED	DECIMAL(20,0)
BINARY	VARBINARY
BIT	VARCHAR
BLOB	VARBINARY
CHAR	VARCHAR
DATE	DATE
DATETIME	DATETIME
DECIMAL	DECIMAL
DOUBLE	DOUBLE
ENUM	VARCHAR
FLOAT	FLOAT
GEOMETRY	VARBINARY
GEOMETRYCOLLECTION	VARBINARY
INT UNSIGNED	BIGINT
INTEGER	INT
JSON	JSON
LINestring	VARBINARY
LOBLOB	VARBINARY
LONGTEXT	VARCHAR
MEDIUMBLOB	VARBINARY
MEDIUMINT	INT
MEDIUMINT UNSIGNED	INT

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
MEDIUMTEXT	VARCHAR
MULTILINESTRING	VARBINARY
MULTIPOINT	VARBINARY
MULTIPOLYGON	VARBINARY
NUMERIC	DECIMAL
POINT	VARBINARY
POLYGON	VARBINARY
SET	VARCHAR
SMALLINT UNSIGNED	INT
TEXT	VARCHAR
TIME	TIME
TIMESTAMP	TIMESTAMP
TINYBLOB	VARBINARY
TINYINT UNSIGNED	SMALLINT
TINYTEXT	VARCHAR
VARBINARY	VARBINARY
VARCHAR	VARCHAR
YEAR	INT

Procedure

1. **Create a data synchronization instance.**

 **Note** When you create the data synchronization instance, set Source Instance Type to **MySQL**, set Destination Instance Type to **AnalyticDB**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

1. Select Source and Destination Instances for
2. Authorize AnalyticDB Account
3. Select Object to Be Synchronized
4. Precheck

Synchronization Task Name:

Source Instance Details

Instance Type:

Instance Region:

* Instance ID:

* Database Account:

* Database Password:

* Encryption: Non-encrypted SSL-encrypted

Destination Instance Details

Instance Type:

Instance Region:

* Version: 2.0 3.0

* Database:

* Database Account:

* Database Password:

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select RDS Instance.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS instance. The account must have the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. 🔔 Note If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.
	Database Password	Enter the password of the source database account.
	Encryption	Select an encryption method. If you select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
	Instance Type	This parameter is set to AnalyticDB and cannot be changed.

Section	Parameter	Description
Destination Instance Details	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Version	Select 3.0.
	Database	Select the ID of the destination AnalyticDB for MySQL cluster.
	Database Account	Enter the database account of the destination AnalyticDB for MySQL cluster. The account must have the read and write permissions on the destination database.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Configure the synchronization policy and objects.

Parameter	Description
Initial Synchronization	You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.

Parameter	Description
Processing Mode In Existed Target Table	<ul style="list-style-type: none"> ◦ Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. ◦ Ignore: skips the precheck for identical table names in the source and destination databases. <div style="background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p> Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</p> <ul style="list-style-type: none"> ▪ DTS does not synchronize the data records that have the same primary keys as the data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. ▪ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails. </div>
Merge Multi Tables	<ul style="list-style-type: none"> ◦ If you select Yes, DTS adds the <code>__dts_data_source</code> column to each table to record data sources. In this case, DDL operations cannot be synchronized. ◦ No is selected by default. In this case, DDL operations can be synchronized. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of specific tables, you can create two data synchronization tasks.</p> </div>
Synchronization Type	<p>Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default.</p> <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note Only the INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized.</p> </div>
Select Objects	<p>Select objects (tables or a database) from the Available section and click the  icon to move the objects to the Selected section.</p> <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ◦ If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. ◦ After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance. </div>

6. In the lower-right corner of the page, click **Next**.

7. Specify a type for the tables that you want to synchronize to the destination database.

1. Configure Source and Destination Instances		2. Authorize AnalyticDB Account		3. Select Objects to Synchronize		4. Precheck	
AnalyticDB Table Group	AnalyticDB Table Name	Type(All) ▾	Primary Key Column	Distribution Column	Definition Status(All) ▾		
dstestdata	customer	Partitioned 1 ▾	id	id ▾	Defined		
dstestdata	order	Partitioned 1 ▾	orderid	orderid ▾	Defined		

[Set All to Partitioned Table](#)
[Set All to Dimension Table](#)

Total: 2 item(s), Per Page: 20 ▾ item(s) << < 1 > >>

Note After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to AnalyticDB for MySQL.

8. In the lower-right corner of the page, click **Precheck**.

Note You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the Precheck dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

19.4.4.4. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature allows you to transfer and analyze data with ease.

Prerequisites

- The tables that you want to synchronize contain primary keys.
- An AnalyticDB for PostgreSQL instance is created.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the schemas of the required objects from the source database to the destination database.
- DTS does not synchronize the following types of data: JSON, GEOMETRY, CURVE, SURFACE, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION, and BYTEA.

SQL operations that can be synchronized

- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE
- Data definition language (DDL) operations: ALTER TABLE, ADD COLUMN, DROP COLUMN, and RENAME COLUMN

 **Note** The CREATE TABLE and DROP TABLE operations are not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see [Add objects to be synchronized](#).

Term mappings

Term in ApsaraDB RDS for MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Create a data structure in the destination instance

Create a database, schema, and table in the destination AnalyticDB for PostgreSQL instance based on the data structure of the source RDS instance.

Configure a data synchronization task

1. [Create a data synchronization instance](#).

 **Note** When you create the data synchronization instance, set Source Instance Type to **MySQL**, set Destination Instance Type to **AnalyticDB for PostgreSQL**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select RDS Instance .
	Instance Region	The region where the source RDS instance resides.
	Instance ID	Select the ID of the source RDS instance.
	Encryption	Select Non-encrypted or SSL-encrypted .  Note If you select SSL-encrypted , you must enable SSL encryption for the RDS instance before you configure the data synchronization task.
Destination Instance	Instance Type	This parameter is set to AnalyticDB for PostgreSQL and cannot be changed.
	Instance Region	The region where the destination instance resides.
	Instance ID	Select the ID of the destination AnalyticDB for PostgreSQL instance.
	Database Name	Enter the name of the destination database.

Destination Instance Details	Parameter	Description
	Database Account	<p>Enter the database account of the destination AnalyticDB for PostgreSQL instance.</p> <p>Note The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions.</p>
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.
5. Wait until the synchronization account is created. Then, click **Next**.
6. Configure the synchronization policy and objects.

Section	Parameter	Description
Synchronization policy	Initial Synchronization	<p>Select Initial Full Data Synchronization.</p> <p>Note DTS synchronizes the historical data of the required objects from the source instance to the destination instance. The historical data is the basis for subsequent incremental synchronization.</p>
	Processing Mode In Existed Target Table	<ul style="list-style-type: none"> ◦ Pre-Check and intercept (Selected by default) Checks the Schema Name Conflict item and generates an error message if the destination table contains data. ◦ Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. ◦ Ignore Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode.
	Synchronization Type	<ul style="list-style-type: none"> ◦ Insert ◦ Update ◦ Delete ◦ AlterTable <p>Note Select the types of operations that you want to synchronize based on your business requirements.</p>

Section	Parameter	Description
Select Objects	N/A	<p>You can select only tables as the objects to be synchronized. You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add objects to be synchronized.</p> </div>

7. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

8. Close the Precheck dialog box after the following message is displayed: **The precheck is passed**. Then, DTS performs initial synchronization.

19.4.4.5. Synchronize data between Cloud Native Distributed Database PolarDB-X instances

Cloud Native Distributed Database PolarDB-X is formerly known as Distributed Relational Database Service (DRDS). It is compatible with the MySQL protocol and syntax, and supports automatic sharding, online smooth scaling, auto scaling, and transparent read/write splitting. This topic describes how to synchronize data between Cloud Native Distributed Database PolarDB-X instances by using Data Transmission Service (DTS).

Prerequisites

The tables that you want to synchronize contain primary keys.

Precautions

- DTS uses the read and write resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- We recommend that you do not change the network type of the Cloud Native Distributed Database PolarDB-X instances during data synchronization.

 **Note** After you change the network type of a Cloud Native Distributed Database PolarDB-X instance during data synchronization, you must submit a ticket to resume the data synchronization instance.

- We recommend that you do not scale up or down the databases in the Cloud Native Distributed Database PolarDB-X instances. Otherwise, data may fail to be synchronized.

Supported synchronization topologies

DTS supports the following synchronization topologies: one-way one-to-one synchronization, one-way one-to-many synchronization, one-way cascade synchronization, and one-way many-to-one synchronization. For more information, see [Synchronization topologies](#).

SQL operations that can be synchronized

The INSERT, UPDATE, and DELETE operations can be synchronized.

Before you begin

Create a database and tables in the destination instance based on the schemas of the objects in the source instance. This is because DTS does not support initial schema synchronization between Cloud Native Distributed Database PolarDB-X instances.

Note During initial schema synchronization, DTS synchronizes the schemas of the required objects from the source database to the destination database.

Procedure

1. **Create a data synchronization instance.**

Note When you create the data synchronization instance, set both Source Instance Type and Destination Instance Type to **Drds**, and set Synchronization Mode to **One-Way Synchronization**.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

1. Select Source and Destination
2. Select Object to Be Synchronized
3. Advanced Settings
4. Precheck

Synchronization Task Name:

Source Instance Details

Instance Type: DRDS Instance

Instance Region:

* DRDS Instance ID:

Destination Instance Details

Instance Type: DRDS Instance

Instance Region:

* DRDS Instance ID:

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	This parameter is set to DRDS Instance and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.

Section	Parameter	Description
Destination Instance Details	Instance Type	This parameter is set to DRDS Instance and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the destination Cloud Native Distributed Database PolarDB-X instance.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Configure the synchronization policy and objects.

Parameter	Description
-----------	-------------

Parameter	Description
Processing Mode In Existed Target Table	<ul style="list-style-type: none"> ◦ Pre-check and Intercept: checks whether the destination tables are empty. If the destination tables are empty, the precheck is passed. If the tables are not empty, an error is returned during the precheck and the data synchronization task cannot be started. ◦ Ignore: skips the check for empty destination tables. <div style="background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p> Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</p> <ul style="list-style-type: none"> ▪ If the source and destination databases have the same schema, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. ▪ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails. </div>
Select Objects	<p>Select tables from the Available section and click the  icon to move the tables to the Selected section.</p> <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ◦ You can select only tables as the objects to be synchronized. ◦ After an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination instance. For more information, see Specify the name of an object in the destination instance. </div>

6. Click **Next**.

7. Specify whether you want to perform initial full data synchronization.

 **Note** During initial full data synchronization, DTS synchronizes the historical data of the required objects from the source database to the destination database. If you do not select Initial Full Data Synchronization, DTS does not synchronize the historical data.

8. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**. Then, DTS

performs initial synchronization.

19.4.4.6. Synchronize data from a Cloud Native Distributed Database PolarDB-X instance to an AnalyticDB for MySQL cluster

This topic describes how to synchronize data from a Cloud Native Distributed Database PolarDB-X instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

The tables that you want to synchronize contain primary keys.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Supported synchronization topologies

DTS supports the following synchronization topologies: one-way one-to-one synchronization, one-way one-to-many synchronization, and one-way many-to-one synchronization. For more information, see [Synchronization topologies](#).

SQL operations that can be synchronized

The INSERT, UPDATE, and DELETE operations can be synchronized.

Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. The following table lists the data types that DTS can convert.

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
BIGINT UNSIGNED	DECIMAL(20,0)
BINARY	VARBINARY
BIT	VARCHAR
BLOB	VARBINARY
CHAR	VARCHAR
DATE	DATE
DATETIME	DATETIME
DECIMAL	DECIMAL
DOUBLE	DOUBLE
ENUM	VARCHAR
FLOAT	FLOAT

Data type of ApsaraDB RDS for MySQL	Data type of AnalyticDB for MySQL
GEOMETRY	VARBINARY
GEOMETRYCOLLECTION	VARBINARY
INT UNSIGNED	BIGINT
INTEGER	INT
JSON	JSON
LINESTRING	VARBINARY
LOB	VARBINARY
LONGTEXT	VARCHAR
MEDIUMBLOB	VARBINARY
MEDIUMINT	INT
MEDIUMINT UNSIGNED	INT
MEDIUMTEXT	VARCHAR
MULTILINESTRING	VARBINARY
MULTIPOINT	VARBINARY
MULTIPOLYGON	VARBINARY
NUMERIC	DECIMAL
POINT	VARBINARY
POLYGON	VARBINARY
SET	VARCHAR
SMALLINT UNSIGNED	INT
TEXT	VARCHAR
TIME	TIME
TIMESTAMP	TIMESTAMP
TINYBLOB	VARBINARY
TINYINT UNSIGNED	SMALLINT
TINYTEXT	VARCHAR
VARBINARY	VARBINARY
VARCHAR	VARCHAR
YEAR	INT

Procedure

1. Create a data synchronization instance.

Note When you create the data synchronization instance, set Source Instance Type to Drds, set Destination Instance Type to AnalyticDB, and set Synchronization Mode to One-Way Synchronization.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.

3. Configure the source and destination instances.

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	This parameter is set to DRDS Instance and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.
Destination Instance Details	Instance Type	This parameter is set to AnalyticDB and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Version	Select 3.0 .
	Database	Select the ID of the destination AnalyticDB for MySQL cluster.
	Database Account	Enter the database account of the destination AnalyticDB for MySQL cluster.
	Database Password	Enter the password of the destination database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Configure the synchronization policy and objects

Section	Parameter	Description
	Initial Synchronization	You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.

Section	Parameter	Description
Synchronization policy	Processing Mode In Existed Target Table	<ul style="list-style-type: none"> ◦ Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. ◦ Ignore Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode.
	Synchronization Type	<p>Select the types of operations that you want to synchronize based on your business requirements.</p> <ul style="list-style-type: none"> ◦ Insert ◦ Update ◦ Delete
Select Objects	N/A	<p>Select tables from the Available section and click the  icon to move the tables to the Selected section.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p>Note</p> <ul style="list-style-type: none"> ◦ You can select only tables as the objects to be synchronized. ◦ You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance. </div>

6. Click **Next**.

7. Specify a type for the tables that you want to synchronize to the destination database.

1. Configure Source and Destination Instances
2. Authorize AnalyticDB Account
3. Select Objects to Synchronize
4. Precheck

AnalyticDB Table Group	AnalyticDB Table Name	Type(All) ▼	Primary Key Column	Distribution Column	Definition Status(All) ▼
dtstestdata	customer	Partitioned 1 ▼	id	id ▼	Defined
dtstestdata	order	Partitioned 1 ▼	orderid	orderid ▼	Defined

[Set All to Partitioned Table](#)
[Set All to Dimension Table](#)

Total: 2 item(s), Per Page: 20 item(s) « < 1 > »

Note After you select **Initial Schema Synchronization**, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to AnalyticDB for MySQL.

8. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

9. Close the Precheck dialog box after the following message is displayed: **The precheck is passed.** Then, DTS performs initial synchronization.

19.4.4.7. Synchronize data from a Cloud Native Distributed Database PolarDB-X instance to a DataHub instance

This topic describes how to synchronize data from a Cloud Native Distributed Database PolarDB-X instance to a DataHub instance by using Data Transmission Service (DTS). After you synchronize data, you can use big data services such as Realtime Compute to analyze data in real time.

Prerequisites

- The tables that you want to synchronize have PRIMARY KEY or UNIQUE constraints.
- A DataHub project is created to receive the synchronized data.

Precautions

- If you select one or more tables (not a database) as the required objects, do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- Only one-way synchronization is supported.

Limits

- You can select only tables as the objects to be synchronized.
- Initial full data synchronization is not supported. DTS does not synchronize the historical data of the required objects from the source PolarDB-X instance to the destination DataHub instance.
- DTS does not synchronize data definition language (DDL) operations to the destination database. If you perform a DDL operation on the source PolarDB-X instance during data synchronization, data fails to be synchronized. To solve this issue, you must modify the related topic in the destination DataHub instance and then restart the data synchronization task.

SQL operations that can be synchronized

The INSERT, UPDATE, and DELETE operations can be synchronized.

Procedure

1. [Create a data synchronization instance.](#)

 **Note** When you create the data synchronization instance, set Source Instance Type to Drds, set Destination Instance Type to Datahub, and set Synchronization Mode to One-Way Synchronization.

2. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column.
3. Configure the source and destination instances.

Section	Parameter	Description
---------	-----------	-------------

Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	This parameter is set to DRDS Instance and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the source Cloud Native Distributed Database PolarDB-X instance.
Destination Instance Details	Instance Type	This parameter is set to DataHub and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you created the data synchronization instance. You cannot change the value of this parameter.
	Project	Select the name of the DataHub project.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Configure the synchronization policy and objects.

Parameter	Description
Initial Synchronization	<p>Select Initial Schema Synchronization.</p> <p> Note After you select Initial Schema Synchronization, DTS synchronizes the schemas of the required objects (such as tables) to the destination DataHub instance.</p>
Select Objects	<p>Select objects from the Available section and click the  icon to move the objects to the Selected section.</p> <p> Note</p> <ul style="list-style-type: none"> ◦ You can select only tables as the objects to be synchronized. ◦ You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Specify the name of an object in the destination instance.

6. In the lower-right corner of the page, click **Precheck**.

 **Note** You can start the data migration task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**. Then, DTS performs initial synchronization.

19.4.4.8. Configure two-way data synchronization between RDS instances

19.4.4.8.1. Overview

DTS supports two-way real-time data synchronization between RDS instances on any two clouds. This section describes how to use DTS to create a two-way synchronization task between two ApsaraDB RDS for MySQL instances for active geo-redundancy, geo-disaster recovery, and other scenarios.

19.4.4.8.2. Supported synchronization statements

Two-way synchronization between ApsaraDB RDS for MySQL instances supports all DML updates (including INSERT, UPDATE, and DELETE) and the following DDL updates:

- ALTER TABLE, ALTER VIEW, ALTER FUNCTION, and ALTER PROCEDURE
- CREATE DATABASE, CREATE SCHEMA, CREATE INDEX, CREATE TABLE, CREATE PROCEDURE, CREATE FUNCTION, CREATE TRIGGER, CREATE VIEW, and CREATE EVENT
- DROP FUNCTION, DROP EVENT, DROP INDEX, DROP PROCEDURE, DROP TABLE, DROP TRIGGER, and DROP VIEW
- RENAME TABLE and TRUNCATE TABLE

 **Note** To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates on the same table in only one direction.

For example, for two-way synchronization, you must enable DDL synchronization in either the A-to-B or B-to-A direction. If DDL synchronization is configured in one direction, it is not supported in the reverse direction. You can only perform DML synchronization.

19.4.4.8.3. Detect and resolve conflicts

To ensure data consistency, for two-way synchronized instances, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the instances. If you unexpectedly update a record with the same primary key, business primary key, or unique key on both instances that are two-way synchronized, a synchronization conflict occurs. To maximize the stability of two-way synchronized instances, DTS supports detecting and resolving data conflicts.

Considerations

During two-way synchronization, the system time of the source and destination instances may not be the same. Additionally, synchronization delays may occur. For these reasons, DTS cannot guarantee that its conflict detection mechanism can completely prevent data conflicts. You must refactor certain business logic to ensure that records of the same primary key, business primary key, or unique key are updated only on one of the instances that are two-way synchronized.

Supported conflict types

Currently, DTS supports detecting the following conflict types:

- Uniqueness conflicts caused by INSERT operations

A uniqueness conflict occurs when the synchronization of an inserted row violates the unique constraint. For example, if two instances in two-way synchronization insert a record with the same primary key value at almost the same time, one of the inserted records fails to be synchronized because a record with the same primary key value already exists in the destination instance.

- Inconsistent records caused by UPDATE operations

Update conflicts occur in the following scenarios:

- The records to be updated do not exist in the destination instance. If the records to be updated do not exist, DTS automatically changes the UPDATE operation to the INSERT operation and inserts these records to the destination instance. In this case, duplicate unique key values may occur.
- The primary keys or unique keys of the records to be updated conflict with each other.
- A DELETE operation is made on non-existent records
A delete conflict occurs when the records to be deleted do not exist in the destination instance.
In this case, DTS automatically ignores the DELETE operation regardless of the conflict resolution policy that you have configured.

Supported conflict resolution policies

For the preceding synchronization conflicts, DTS provides the following resolution policies. You can select a conflict resolution policy as required when configuring two-way synchronization.

- **TaskFailed:** The synchronization task reports an error and automatically exits the process in case of a conflict.
When the synchronization encounters a conflict of the preceding types, the synchronization task reports an error and automatically exits the process. The task enters a failed state and you must manually resolve the conflict. This method is the default conflict resolution policy.
- **Ignore:** The records in the destination instance are used in case of a conflict.
When the synchronization encounters a conflict of the preceding types, the synchronization task skips the current synchronization statement and continues the process. The records in the destination instance are used.
- **Overwrite:** The conflict records in the destination instance are overwritten in case of a conflict.
When the synchronization encounters a conflict of the preceding types, the conflict records in the destination instance are overwritten.

19.4.4.8.4. Synchronization restrictions

This section describes the restrictions in cross-cloud data synchronization using DTS.

Restrictions in data sources

Currently, only ApsaraDB RDS for MySQL instances support two-way synchronization. Other heterogeneous data sources do not support two-way synchronization.

The destination instance cannot be an RDS instance that runs in standard access mode and has only a public network address.

Restrictions in synchronization architecture

Currently, DTS only supports two-way synchronization between two ApsaraDB RDS for MySQL instances. Two-way synchronization between more than two instances is not supported.

Feature restrictions

- Incompatible with triggers

When you synchronize an entire database and the database contains a trigger that updates the synchronization table, the synchronized data may be inconsistent.

For example, the object to be synchronized is database A that contains table a and table b. Table a has a trigger that inserts a row to table b after the row is inserted to table a. In this case, if an INSERT operation is performed on table a in the source instance during synchronization, the data in table b is inconsistent between the source and destination instances.

To resolve this problem, you must delete the trigger in the destination instance, so that the data in table b is only synchronized from the source instance.

- Restrictions in the RENAME TABLE operation

The RENAME TABLE operation may result in inconsistent synchronization data. For example, if the object to be synchronized only includes table a and the rename a to b command is executed in the source instance during synchronization, subsequent operations to the renamed table b are not synchronized to the destination database. To solve this problem, you can synchronize the entire database where table a and table b are stored.

- Restrictions in DDL synchronization direction

To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates on the same table in only one direction. For example, in A-to-B and B-to-A synchronization, you can implement DDL synchronization in either the A-to-B or B-to-A direction. If DDL synchronization is configured in one direction, it is not supported in the reverse direction.

19.4.4.8.5. Configure two-way data synchronization between ApsaraDB RDS for MySQL instances across regions

This topic describes how to configure two-way data synchronization between ApsaraDB RDS for MySQL instances across regions.

Prerequisites

The source and destination ApsaraDB RDS for MySQL instances are created.

Procedure

1. [Log on to the DTS console](#).
2. In the left-side navigation pane, click **Data Synchronization**.
3. On the **Synchronization Tasks** page, click **Create Synchronization Task** in the upper-right corner.

 **Note**

Source Instance Region: Select the region where the source RDS instance resides.

Source Instance Type: Select the type of the source instance. In this example, select MySQL.

Destination Instance Region: Select the region where the destination RDS instance resides.

Destination Instance Type: Select the type of the destination instance. In this example, select MySQL.

Synchronization Mode: Select the synchronization mode. In this example, select Two-Way Synchronization.

Instances to Create: Set the number of instances that you want to create.

4. After you configure the preceding information, click **Create**.

After you create a synchronization instance, go back to the **Synchronization Tasks** page. The new synchronization instance is in the Not Configured state and contains two synchronization tasks. You can configure two-way synchronization for the tasks.

5. Find one of the created synchronization tasks and click **Configure Synchronization Channel** in the Actions column.
6. Configure the parameters for the data synchronization task.
 - Synchronization Task Name

We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
 - RDS instance ID

You must specify the ID of the Apsara Stack tenant account to which the destination RDS instance belongs. You can then select an RDS instance ID from the Instance ID drop-down list.

After you complete the preceding configurations, click **Set Whitelist** and **Next** to configure the RDS instance whitelists.

7. Configure the RDS instance whitelists.

In this step, DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination RDS instances. This ensures that DTS servers can connect to the RDS instances and the data synchronization task can be created.

We recommend that you do not remove the CIDR blocks of DTS servers from the whitelists of the RDS instances. This ensures the stability of the data synchronization task.

Click **Next** to create a data synchronization account.

8. Create a data synchronization account in the destination RDS instance.

Create a data synchronization account named `dtssyncwriter` in the destination RDS instance. Do not delete the account during data synchronization. Otherwise, an interruption occurs.

9. Configure the synchronization policies and select the objects that you want to synchronize.

After you create a data synchronization account, you must configure the synchronization policies and select the objects that you want to synchronize.

- **Exclude DDL Statements**

Specify whether to synchronize DDL statements in a specific direction. To include DDL statements, select **No**. To exclude DDL statements, select **Yes**. If you select **No**, DTS does not synchronize the DDL operations that are performed on a table in the opposite direction.

- **DML Statements for Synchronization**

Select the types of DML operations that you want to synchronize. By default, **Insert**, **Update**, and **Delete** are selected.

- **Conflict Resolution Policy**

Select the resolution policy for synchronization conflicts. By default, **TaskFailed** is selected.

For example, if Node A is the primary business center and Node B is a secondary business center, you must give the priority to Node A. You must set the conflict resolution policy in the A-to-B direction to **Overwrite** and that in the B-to-A direction to **Ignore**.

- **Select Objects**

You can select databases and tables as the objects to be synchronized.

If you select an entire database, all schema changes such as the **CREATE TABLE** and **DROP VIEW** operations that performed on the objects in the database are synchronized to the destination database.

If you select a table, only the **DROP TABLE**, **ALTER TABLE**, **TRUNCATE TABLE**, **RENAME TABLE**, **CREATE INDEX**, and **DROP INDEX** operations that performed on this table are synchronized to the destination database.

10. Configure initial synchronization.

Initial synchronization is the first step to start the synchronization task. During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

Initial synchronization includes **initial schema synchronization** and **initial full data synchronization**. You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases.

If the tables to be synchronized in one direction are also included in the objects to be synchronized in the opposite direction, DTS does not synchronize these tables during initial synchronization.

11. Run a precheck.

After the data synchronization task is configured, DTS performs a precheck. Close the **Precheck** dialog box after the task passes the precheck.

After the task is started, the task list appears. The task is in the **Performing Initial Sync** state. The duration of the initial synchronization depends on the data volume of the objects that you want to synchronize. After initial synchronization, the task status changes to **Synchronizing**. This indicates that the data synchronization task is created.

After the task is configured in one direction, the source and destination RDS instances of the task in the opposite direction cannot be changed.

12. Repeat steps 5 to 11 to configure the data synchronization task in the opposite direction.

19.4.5. Manage data synchronization instances

19.4.5.1. Specify the name of an object in the destination instance

After an object, such as a database or table, is synchronized from the source instance to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature provided by DTS to specify a different name for the object in the destination instance.

Notes

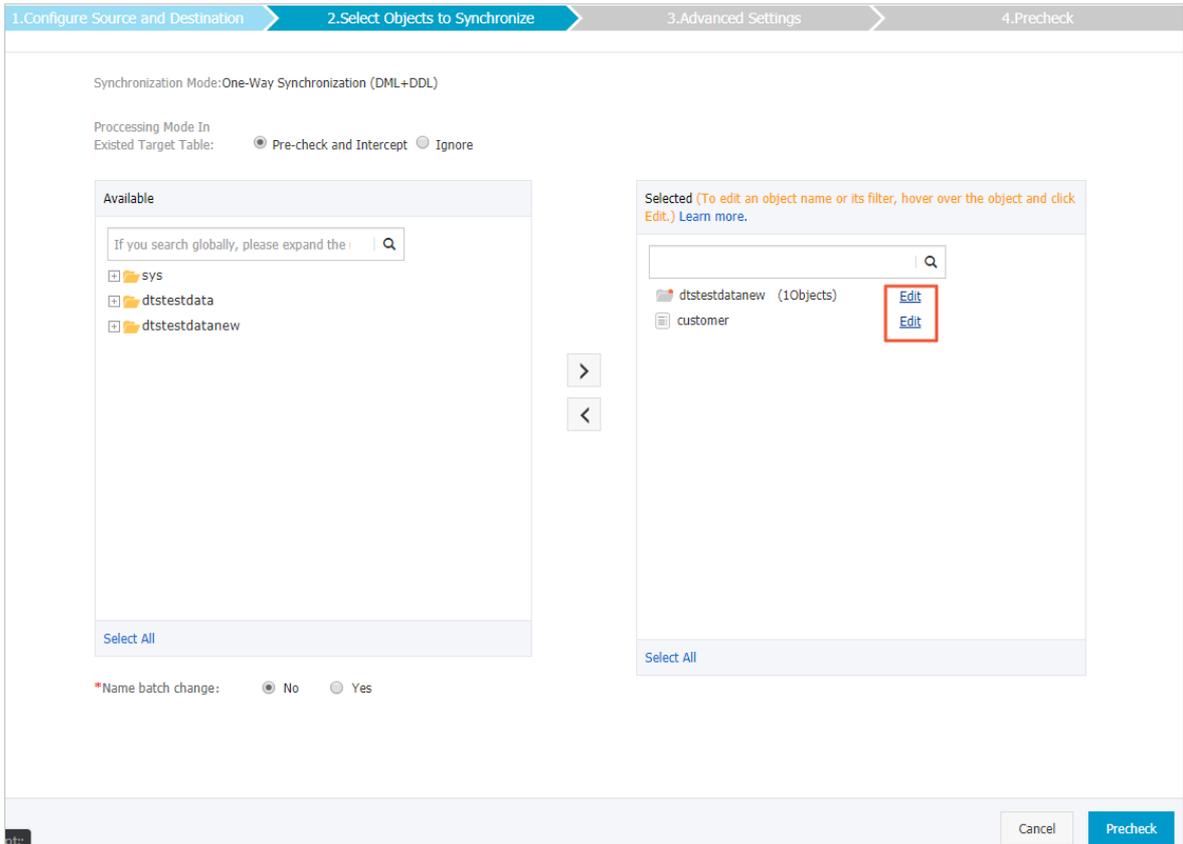
You can perform this operation only when a data synchronization task is configured and the current process is **Select Objects to Synchronize**.

 **Note** Do not perform this operation after the data synchronization task is started. Otherwise, the synchronization may fail.

Procedure

1. On the **Select Objects to Synchronize** page, move the required objects to the **Selected** section, move the pointer over a database or table, and then click **Edit**.

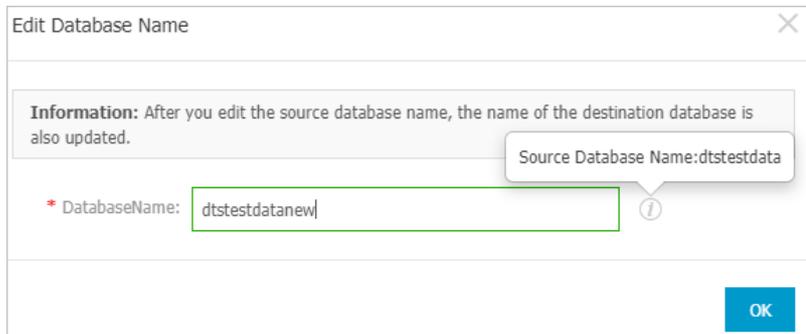
 **Note** Different database types support different objects. If **Edit** appears when you move the pointer over the target object, the operation is supported.



2. In the dialog box that appears, specify a name for the object in the destination instance.

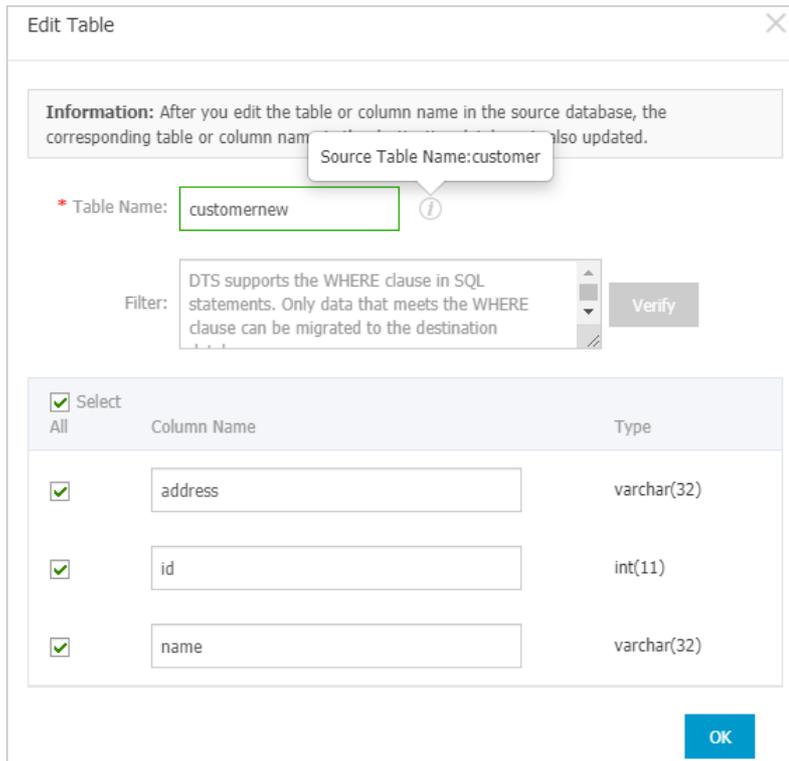
- o Database name mapping

In the **Edit Database Name** dialog box that appears, enter the database name that you want to use in the destination instance.



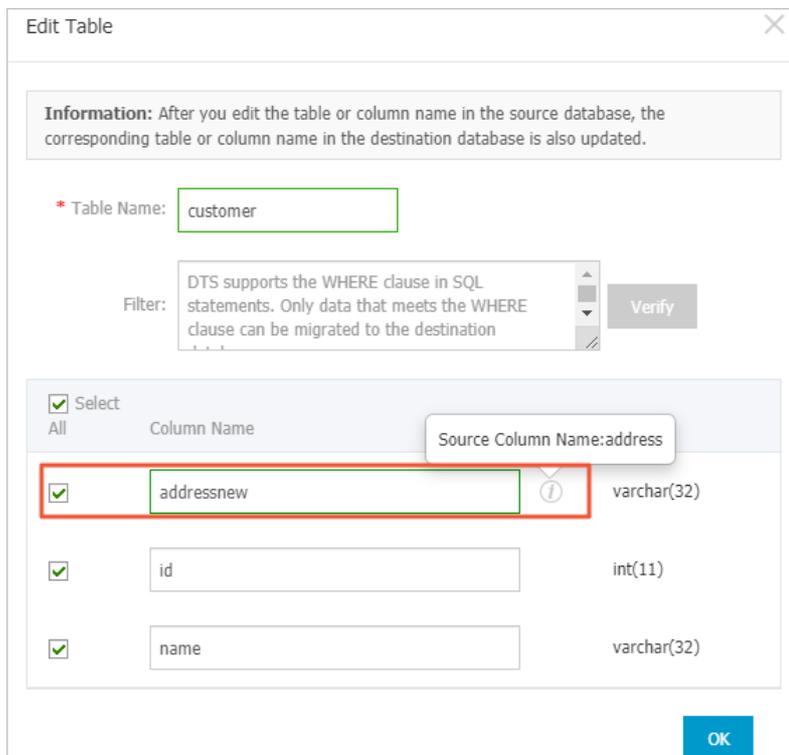
- o Table name mapping

In the **Edit Table** dialog box that appears, enter the table name that you want to use in the destination instance.



o Column name mapping

In the Edit Table dialog box that appears, enter a new name for each column.



Note In this step, you can deselect columns that do not need to be synchronized.

3. Click **OK**.

4. Configure other parameters that are required for the data synchronization task.

19.4.5.2. Check the synchronization performance

DTS provides the trend charts of data synchronization tasks based on three performance metrics: bandwidth, synchronization speed (TPS), and synchronization delay. You can view the running status of data synchronization tasks in the DTS console.

1. [Log on to the DTS console](#).
2. In the left-side navigation pane, click **Data Synchronization**.
3. On the Synchronization Tasks page, click the ID of the data synchronization task that you want to check.
The task details page appears.
4. On the task details page, click **Synchronization Performance** in the left-side navigation pane.
5. View the trend charts of synchronization performance.

DTS provides the trend charts of data synchronization tasks based on three performance metrics: bandwidth, synchronization speed (TPS), and synchronization delay.

- **Bandwidth:** the bandwidth of data that the data writing module pulls from the data pulling module per second. Unit: MB/s.
- **Synchronization speed (TPS):** the number of transactions that DTS synchronizes to the destination instance per second.
- **Synchronization delay:** the difference between the timestamp of the latest synchronized data in the destination instance and the current timestamp in the source instance. Unit: milliseconds.

19.4.5.3. Add objects to a data synchronization task

When a data synchronization task is running, you can add objects to the task or remove objects from the task. This topic describes how to add objects to a data synchronization task in the DTS console.

Limits

You can modify the required objects only when the data synchronization task is in the **Synchronizing** or **Synchronization Failed** state.

Start time of data synchronization

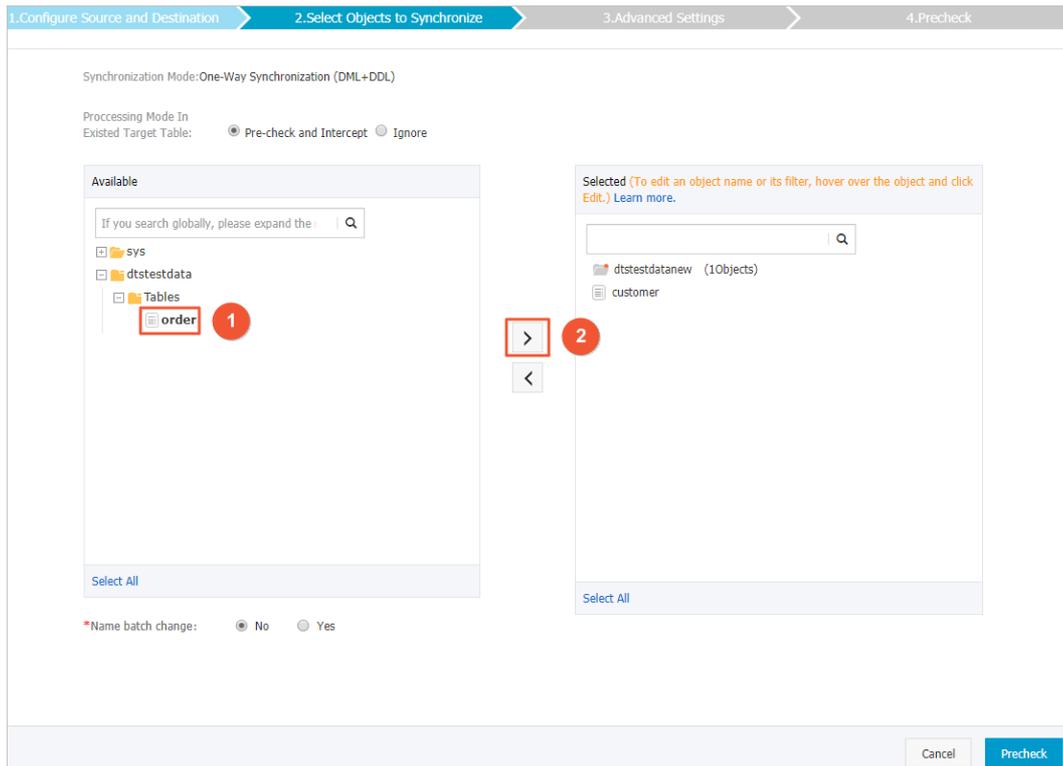
The time when DTS synchronizes data of new objects depends on whether initial synchronization is specified for the data synchronization task.

- If initial synchronization is specified, DTS synchronizes schemas and historical data, and then synchronizes incremental data.
- If initial synchronization is not specified, DTS synchronizes data after incremental data is generated on the source instance.

Procedure

1. [Log on to the DTS console](#).
2. In the left-side navigation pane, click **Data Synchronization**.
3. Find the data synchronization task and choose **More > Modify Objects to Synchronize** in the Actions column.
4. On the **Select Objects to Synchronize** tab, add objects based on your needs, as shown in [Add objects to a data synchronization task](#).

Add objects to a data synchronization task



5. Click Precheck.

After the task passes the precheck, the objects are added to the data synchronization task.

After the objects are added, if initial synchronization is specified for the data synchronization task, the task status changes from **Synchronizing** to **Synchronizing (The initial synchronization of the new objects is being performed.)**.

Note You can click **View More** to view the initial synchronization progress of the new objects. After the initial synchronization on the new objects is complete, the task status returns to **Synchronizing**.

19.4.5.4. Remove objects from a data synchronization task

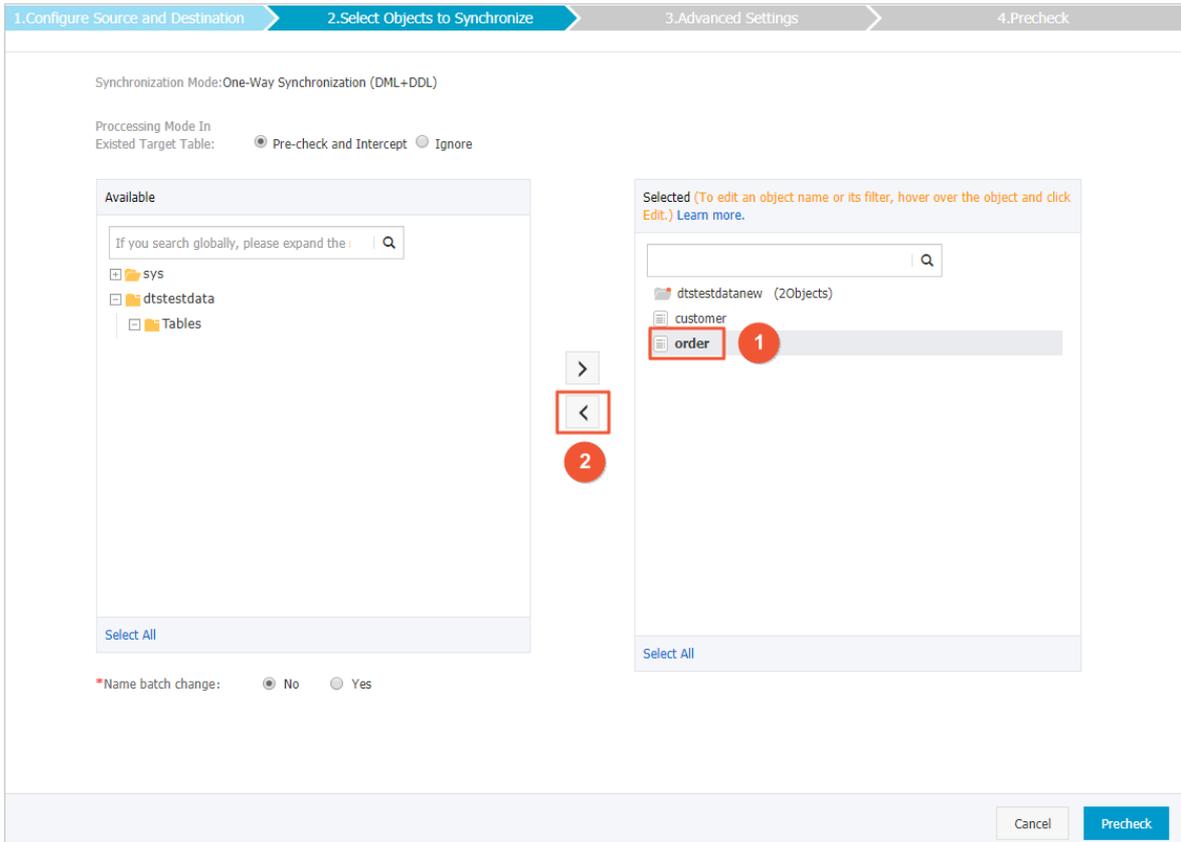
When a data synchronization task is running, you can add objects to the task or remove objects from the task. This topic describes how to remove objects from a data synchronization task in the DTS console.

Limits

You can modify the required objects only when the data synchronization task is in the **Synchronizing** or **Synchronization Failed** state.

Procedure

1. **Log on to the DTS console.**
2. In the left-side navigation pane, click **Data Synchronization**.
3. Find the data synchronization task and choose **More > Modify Objects to Synchronize** in the Actions column.
4. On the **Select Objects to Synchronize** tab, remove objects based on your needs.



5. Click **Precheck** to run a precheck.

19.4.5.5. Troubleshoot precheck failures

Before Data Transmission Service (DTS) runs a data synchronization task, DTS performs a precheck. This topic describes the precheck items and how to troubleshoot precheck failures.

Source database connectivity

- **Description**
DTS checks whether DTS servers can connect to the source RDS instance. DTS creates a connection to the source RDS instance by using the JDBC protocol. If the connection fails, the task fails to pass the precheck.
- **Cause of failure**
 - DTS does not support data synchronization between RDS instances in the region where the source instance resides.
 - The database account or password of the source instance is invalid.
- **Solution**
Submit a ticket and contact Alibaba Cloud technical support.

Destination database connectivity

- **Description**
DTS checks whether DTS servers can connect to the destination RDS instance. DTS creates a connection to the destination RDS instance by using the JDBC protocol. If the connection fails, the task fails to pass the precheck.
- **Cause of failure**
 - DTS does not support data synchronization between RDS instances in the region where the destination instance resides.

- The database account or password of the destination instance is invalid.

- **Solution**

Submit a ticket and contact Alibaba Cloud technical support.

Source database version

- **Description**

DTS checks whether:

- The database version of the source RDS instance is supported by the data synchronization feature.
- The database version of the destination RDS instance is the same as the database version of the source RDS instance.

- **Cause of failure**

- The database version of the source RDS instance is earlier than the supported database versions. The data synchronization feature supports the following database versions: MySQL 5.1, 5.5, 5.6, and 5.7.
- The database version of the destination RDS instance is earlier than the database version of the source RDS instance.

- **Solution**

- If the database version of the source RDS instance is earlier than the supported database versions, upgrade the source RDS instance to MySQL 5.6 or 5.7 in the RDS console. Then, create a data synchronization task again.
- If the database version of the destination RDS instance is earlier than the database version of the source RDS instance, upgrade the destination RDS instance to MySQL 5.6 or 5.7 in the RDS console. Then, create a data synchronization task again.

Database existence

DTS checks whether the destination database already exists in the destination instance. If the destination database does not exist in the destination instance, DTS automatically creates a database. However, DTS fails to create the database and reports a failure under the following circumstances:

- The database name contains characters other than lowercase letters, digits, underscores (_), and hyphens (-).
- The character set of the database is not UTF-8, GBK, Latin1, or UTF-8MB4.
- The account of the destination database does not have the read/write permissions on the source database.

If the data source is an RDS instance, the task passes the precheck.

Source database permissions

DTS checks whether the account of the source database has the required permissions. If the account does not have the required permissions, the task fails to pass the precheck. If the source database is an RDS instance, the task passes the precheck.

Destination database permissions

- **Description**

DTS checks whether the account of the destination database has the required permissions. If the account does not have the required permissions, the task fails to pass the precheck.

- **Cause of failure**

- DTS fails to create a database account in the destination RDS instance.
- DTS fails to grant the read/write permissions to the database account of the destination RDS instance.

- **Solution**

Submit a ticket and contact Alibaba Cloud technical support.

Object name conflict

- Description

DTS checks object names only if you select initial synchronization for a data synchronization task. DTS checks whether an object that you want to synchronize has the same name as an object in the destination RDS instance.

- Cause of failure

If an object in the destination RDS instance has the same name as the object that you want to synchronize, the task fails to pass the precheck.

- Solution

- Remove the conflicting object from the destination database.
- Then, create a data synchronization task again. Select both Initial Schema Synchronization and Initial Full Data Synchronization.

Value of server_id in the source database

DTS checks whether the value of the server_id parameter in the source database is set to an integer that is greater than or equal to 2. If the data source is an RDS instance, the task passes the precheck.

Whether binary logging is enabled for the source database

DTS checks whether the binary logging feature is enabled for the source database. If the binary logging feature is disabled for the source database, the task fails to pass the precheck. If the data source is an RDS instance, the task passes the precheck.

Binary log format of the source database

DTS checks whether the binary log format of the source database is set to ROW. If the binary log format of the source database is not set to ROW, the task fails to pass the precheck. If the data source is an RDS instance, the task passes the precheck.

Integrity of the FOREIGN KEY constraints

- Description

DTS checks whether the parent tables and child tables that have referential relationships with each other are all included in the required objects. The precheck allows DTS to protect the integrity of the FOREIGN KEY constraints.

- Cause of failure

One or more child tables are included in the required objects. However, the parent tables that are referenced by the child tables are not included in the required objects. This impairs the integrity of the FOREIGN KEY constraints.

- Solution

The following solutions are available:

- Create a data synchronization task again and do not synchronize the child tables that fail to pass the precheck.
- Create a data synchronization task again and add the parent tables to the required objects.
- Remove the FOREIGN KEY constraints from the child tables that fail to pass the precheck. Then, create a data synchronization task again.

Storage engine

- Description

DTS checks whether the required objects use the storage engines that are not supported by the data synchronization feature, such as FEDERATED, MRG_MyISAM, and TokuDB.

- Cause of failure

If the storage engine of a source table is FEDERATED, MRG_MyISAM, or TokuDB, the task fails to pass the precheck.

- Solution

Change the unsupported storage engine to InnoDB and create a data synchronization task again.

Character set

- Description

DTS checks whether the required objects use the character sets that are not supported by the data synchronization feature, such as the UCS-2 character set.

- Cause of failure

If the character sets used by the required objects are not supported by the data synchronization feature, the task fails to pass the precheck.

- Solution

Change the unsupported character sets to UTF-8, GBK, or Latin1. Then, create a data synchronization task again.

Complicated topologies

- Description

DTS checks whether the topology that you specify for the source and destination RDS instances is supported.

- Cause of failure

- The source RDS instance in the current task is being used as the destination instance of another task.
- The destination RDS instance in the current task is being used as the source or destination instance of another task.
- The objects that you want to synchronize in the current task are being synchronized by an existing task. The two tasks have the same source and destination RDS instances.

- Solution

- If the task that you want to create has the same source and destination RDS instances as an existing task, you can add the required objects to the existing task. You do not need to create another task to synchronize these objects.
- If the task that you want to create conflicts with an existing task, wait until the existing task is completed before you create a data synchronization task again.

Format of the MySQL database password

DTS checks whether the format of the password that is used to access the source database is no longer valid. If the data source is an RDS instance, the task passes the precheck.

19.5. Change tracking

19.5.1. Overview

You can use Data Transmission Service (DTS) to track data changes from ApsaraDB RDS for MySQL instances in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations.

Supported databases

- User-created MySQL databases or ApsaraDB RDS for MySQL
- PolarDB-X (formerly known as DRDS)
- User-created Oracle database

Objects for change tracking

The objects for change tracking include tables and databases.

In change tracking, data changes include data manipulation language (DML) operations and data definition language (DDL) operations. When you configure change tracking, you must select operation types.

Change tracking tasks

A change tracking task is the basic unit of change tracking and data consumption. To track data changes from an RDS instance, you must create a change tracking task in the DTS console for the RDS instance. The change tracking task pulls data changes from the RDS instance in real time and locally stores the data changes. You can use the DTS SDK to consume the tracked data. You can also create, manage, or delete change tracking tasks in the DTS console.

19.5.2. Create a change tracking instance

Before you configure a task to track data changes, you must create a change tracking instance. This topic describes how to create a change tracking instance in the Data Transmission Service (DTS) console.

Procedure

1. [Log on to the DTS console](#).
2. In the left-side navigation pane, click **Change Tracking**.
3. In the upper-right corner, click **Create Change Tracking Task**.
4. In the **Create DTS Instances** dialog box, select a region, and enter the number of change tracking instances that you want to create.

 **Note** In the **Create DTS Instances** dialog box, you can view the total number of instances, the number of existing instances, and the number of instances that can be created.

5. Click **Create**.

19.5.3. Configure change tracking tasks

19.5.3.1. Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance

You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance.

Prerequisites

The version of the MySQL database is 5.1, 5.5, 5.6, or 5.7.

Precautions

- DTS does not track data definition language (DDL) operations that are performed by `gh-ost` or `pt-online-schema-change`. Therefore, the change tracking client may fail to write the consumed data to the destination tables due to schema conflict.
- If the source database is used in another task, for example, it is used in a running data migration task, DTS may track data changes of other objects. In this case, you must use the change tracking client to filter the tracked data.

Procedure

1. [Create a change tracking instance](#).
2. Find the change tracking instance that you created, and click **Configure Channel** in the **Actions** column.

3. Configure the source database.

1. Select Instance
2. Select Required Objects
3. Precheck

Task Name:

Source Database

* Version: Old New

* Instance Type:

Database Type: MySQL

Instance Region:

* RDS Instance ID:

Information: Currently, DTS does not support change tracking of read-only instances or temporary instances.

* Database Account:
The account must have the following permissions: REPLICATION SLAVE, REPLICATION CLIENT, and SELECT for all objects to synchronize.

* Database Password:

Consumer network type

* Network Type: Classic

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Version	Select a version based on your business requirements: <ul style="list-style-type: none"> ◦ If the source database is a user-created MySQL database, select Old. ◦ If the source database is ApsaraDB RDS for MySQL, select New. <div style="background-color: #e1f5fe; padding: 5px; border: 1px solid #cfe2f3; margin-top: 5px;"> Note You can follow the same procedure to configure change tracking tasks when you select Old or New. In this example, select New. </div>
	Instance Type	Select RDS Instance .
	Database Type	MySQL is selected by default.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you created the change tracking task. You cannot change the value of this parameter.
	RDS Instance ID	Select the ID of the RDS instance from which you want to track data changes. <div style="background-color: #e1f5fe; padding: 5px; border: 1px solid #cfe2f3; margin-top: 5px;"> Note A read-only instance or temporary instance cannot be used as the source instance for change tracking. </div>

Section	Parameter	Description
	Database Account	<p>Enter the database account of the source RDS instance.</p> <p>Note</p> <ul style="list-style-type: none"> The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, and the permission to perform SELECT operations on the required objects. If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.
	Database Password	<p>Enter the password of the source database account.</p>
Consumer Network Type	Network Type	<p>Classic is selected by default.</p> <p>Note</p> <ul style="list-style-type: none"> This parameter is available only if the Version parameter is set to New. If you track data changes over internal networks, the network latency is minimal.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the data change types and objects.

1.Select Instance 2.Select Required Objects 3.Precheck

Information: If you select an entire database, DTS tracks all the data added to the database. If you only select some tables, you must modify the objects in the Available section to include other required objects in the task.

* Required Data Types: Data Updates Schema Updates ?

Required Objects

If you search globally, please expand the

- [-] dts
 - [-] dtstestdata
 - [-] Tables
 - [-] dtstestdata0925
 - [-] dtstestdatanew
 - [-] sys

Select All

Selected

- [-] dtstestdata(20objects)
 - [-] customer
 - [-] order

Select All

Cancel Previous Save and Precheck

Parameter	Description
Required Data Types	<ul style="list-style-type: none"> ○ Data Updates If you select Data Updates, DTS tracks data updates of the selected objects, including the INSERT, DELETE, and UPDATE operations. ○ Schema Updates If you select Schema Updates, DTS tracks the create, delete, and modify operations that are performed on all object schemas of the source instance. You must use the change tracking client to filter the tracked data. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p>Note</p> <ul style="list-style-type: none"> ○ If you select a database as the object, DTS tracks data changes of all objects, including new objects in the database. ○ If you select a table as the object, DTS tracks only data changes of this table. In this case, if you want to track data changes of another table, you must add the table to the required objects. For more information, see Modify the objects for change tracking. </div>
Required Objects	Select objects from the Required Objects section and click the icon to move the objects to the Selected section.

6. In the lower-right corner of the page, click **Save and Precheck**.

 **Note** You can start a change tracking task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the Precheck dialog box after the following message is displayed: **The precheck is passed.** After the change tracking task is configured, DTS performs initial change tracking, which takes about 1 minute. After the initial change tracking is complete, you can consume the tracked data.

What's next

- Previous change tracking feature: [Run the SDK demo code](#)
- New change tracking feature: [Use a Kafka client to consume tracked data](#)

19.5.3.2. Track data changes from a PolarDB-X instance

You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a PolarDB-X instance. PolarDB-X is formerly known as Distributed Relational Database Service (DRDS).

Precautions

- DTS does not track data definition language (DDL) operations that are performed by gh-ost or pt-online-schema-change. Therefore, the change tracking client may fail to write the consumed data to the destination tables due to schema conflict.
- If the source database is used in another task, for example, it is used in a running data migration task, DTS may track data changes of other objects. In this case, you must use the change tracking client to filter the tracked data.

Procedure

1. [Create a change tracking instance.](#)
2. Find the change tracking instance that you created, and click **Configure Channel** in the **Actions** column.
3. Configure the source database.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	This parameter is set to DRDS Instance and cannot be changed.
	Database Type	This parameter is set to DRDS and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you created the change tracking task. You cannot change the value of this parameter.
	DRDS Instance ID	Select the ID of the PolarDB-X instance.
	Database Name	Select the ID of the source database in the PolarDB-X instance.
	Database Account	Enter the database account of the PolarDB-X instance.
	Database Password	Enter the password of the source database account.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the data change types and objects.

1.Select Instance
2.Select Required Objects
3.Precheck

Information: If you select an entire database, DTS tracks all the data added to the database. If you only select some tables, you must modify the objects in the Available section to include other required objects in the task.

* Required Data Types: Data Updates Schema Updates ⌵

Required Objects

If you search globally, please expand the

- dts
- dtstestdata
 - Tables
- dtstestdata0925
- dtstestdatanew
- sys

Select All

>

<

Selected

dtstestdata(20objects)

- customer
- order

Select All

Cancel
Previous
Save and Precheck

Parameter	Description
Required Data Types	<ul style="list-style-type: none"> ◦ Data Updates If you select Data Updates, DTS tracks data updates of the selected objects, including the INSERT, DELETE, and UPDATE operations. ◦ Schema Updates If you select Schema Updates, DTS tracks the create, delete, and modify operations that are performed on all object schemas of the source instance. You must use the change tracking client to filter the tracked data. <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ◦ If you select a database as the object, DTS tracks data changes of all objects, including new objects in the database. ◦ If you select a table as the object, DTS tracks only data changes of this table. In this case, if you want to track data changes of another table, you must add the table to the required objects. For more information, see Modify the objects for change tracking. </div>
Required Objects	Select objects from the Required Objects section and click the  icon to move the objects to the Selected section.

6. In the lower-right corner of the page, click **Save and Precheck**.

 **Note** You can start a change tracking task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**
After the change tracking task is configured, DTS performs initial change tracking, which takes about 1 minute. After the initial change tracking is complete, you can consume the tracked data.

19.5.3.3. Track data changes from a user-created Oracle database

You can use Data Transmission Service (DTS) to track data changes in real time. This feature applies to the following scenarios: lightweight cache updates, business decoupling, asynchronous data processing, and synchronization of extract, transform, and load (ETL) operations. This topic describes how to track data changes from a user-created Oracle database.

Prerequisites

- The version of the user-created Oracle database is 9i, 10g, 11g, or 12c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see [Supplemental Logging](#).
- The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see [Managing Archived Redo Log Files](#).

Precautions

- DTS does not track data definition language (DDL) operations that are performed by gh-ost or pt-online-schema-change. Therefore, the change tracking client may fail to write the consumed data to the destination tables due to schema conflict.
- If the source database is used in another task, for example, it is used in a running data migration task, DTS

may track data changes of other objects. In this case, you must use the change tracking client to filter the tracked data.

Procedure

1. **Create a change tracking instance.**
2. Find the change tracking instance that you created, and click **Configure Channel** in the **Actions** column.
3. Configure the source database and network type.

The screenshot shows a configuration wizard with three steps: 1. Select Instance, 2. Select Required Objects, and 3. Precheck. The 'Task Name' field is set to 'Oracle'. The 'Source Database' section includes a dropdown for 'Instance Type' (User-Created Database with Public IP Address), 'Database Type' (Oracle), 'Instance Region', and input fields for 'Hostname or IP Address', 'Port Number' (1521), 'SID' (testsid), 'Database Account' (dtstest), and 'Database Password' (masked). The 'Consumer network type' section has a radio button for 'Network Type' set to 'Classic'. At the bottom right are 'Cancel' and 'Set Whitelist and Next' buttons.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select an instance type based on where the source database is deployed. In this example, select User-Created Database with Public IP Address .
	Database Type	This parameter is set to Oracle and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you created the change tracking task. You cannot change the value of this parameter.
	Hostname or IP Address	Enter the hostname or IP address of the user-created Oracle database.
	Port Number	Enter the service port number of the user-created Oracle database.
	SID	Enter the system ID (SID) of the user-created Oracle database.

Section	Parameter	Description
	Database Account	Enter the account of the user-created Oracle database. Note The account must have the database administrator (DBA) permission.
	Database Password	Enter the password of the source database account.
Consumer Network Type	N/A	Classic is selected by default. Note If you track data changes over internal networks, the network latency is minimal.

4. In the lower-right corner of the page, click **Set Whitelist and Next**.

5. Select the data change types and objects.

Information: If you select an entire database, DTS tracks all the data added to the database. If you only select some tables, you must modify the objects in the Available section to include other required objects in the task.

* Required Data Types: Data Updates Schema Updates

Required Objects

- EOA_USER
- DTSTEST
 - Tables
 - GOOD_SALE
 - ORACLESTTABLE1216
- SCOTT
- OWBSYS_AUDIT
- OWBSYS
- APEX_030200
- APEX_PUBLIC_USER
- SPATIAL_CSW_ADMIN_USR
- SPATIAL_WFS_ADMIN_USR
- ORDDATA
- XS\$NULL

Selected

- DTSTEST(10objects)
 - ORACLESTTABLE

Buttons: Cancel, Previous, Save and Precheck

Parameter	Description
-----------	-------------

Parameter	Description
Required Data Types	<ul style="list-style-type: none"> ◦ Data Updates If you select Data Updates, DTS tracks data updates of the selected objects, including the INSERT, DELETE, and UPDATE operations. ◦ Schema Updates If you select Schema Updates, DTS tracks the create, delete, and modify operations that are performed on all object schemas of the source instance. You must use the change tracking client to filter the tracked data. <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ◦ If you select a database as the object, DTS tracks data changes of all objects, including new objects in the database. ◦ If you select a table as the object, DTS tracks only data changes of this table. In this case, if you want to track data changes of another table, you must add the table to the required objects. For more information, see Modify the objects for change tracking. </div>
Required Objects	Select objects from the Required Objects section and click the  icon to move the objects to the Selected section.

6. In the lower-right corner of the page, click **Save and Precheck**.

 **Note** You can start a change tracking task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

7. Close the Precheck dialog box after the following message is displayed: **The precheck is passed.** After the change tracking task is configured, DTS performs initial change tracking, which takes about 1 minute. After the initial change tracking is complete, you can consume the tracked data.

What's next

[Use a Kafka client to consume tracked data](#)

19.5.4. Manage change tracking tasks

19.5.4.1. Modify the consumption checkpoint

During data consumption, you can modify the consumption checkpoint of a change tracking task based on your business requirements. After you modify the consumption checkpoint, the downstream SDK client will consume the data that is generated after the specified time.

Prerequisites

A change tracking task is created. For more information, see [Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance \(previous version\)](#) or [Track data changes from a PolarDB-X instance](#).

Procedure

1. Stop all downstream SDK consumption processes.

Note We recommend that you perform this operation during off-peak hours to avoid service interruption.

2. Log on to the DTS console.
3. In the left-side navigation pane, click Change Tracking.
4. Find the change tracking task, move the pointer over the Consumption Checkpoint column, and then click the  icon.

<input type="checkbox"/>	Task ID/Name	Status	Consumption Checkpoint	Data Range	Billing Method	Actions
<input type="checkbox"/>		Normal	2019-07-18 10:02:32 	2019-09-19 10:33:14 2019-09-26 11:13:56	Pay-As-You-Go	Modify Required Objects More
<input type="checkbox"/>	Delete					Total: 1 item(s) Per Page: 20 item(s) << < 1 > >>

5. In the Modify Consumption Checkpoint dialog box, specify a new consumption checkpoint.

Information: The time you select must be within the range[2019-09-19 10:33:14 - 2019-09-26 11:10:34]that is specified for the channel.

Consumption Checkpoint: 2019-07-18 

10 : 02 : 32

Close Edit

Note The selected time range must be within the time range of the tracked data. For more information, see the prompt in the dialog box.

6. Click Modify.
7. Restart the downstream SDK consumption processes.
The downstream SDK client tracks data changes from the new consumption checkpoint.

19.5.4.2. Modify the objects for change tracking

DTS allows you to add or remove the objects for change tracking in the consumption process. This topic describes how to modify the objects for change tracking.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click Change Tracking.
3. Find the change tracking task, and click Modify Required Objects in the Actions column.
4. In the Select Required Objects step, add or remove the objects for change tracking.
 - o Add the objects for change tracking
In the Required Objects section, select one or more objects and click the  icon to add the objects to the Selected section.
 - o Remove the objects for change tracking
In the Selected section, select one or more objects and click the  icon to move the objects to the Required Objects section.

- In the lower-right corner of the page, click **Save and Precheck**.

 **Note** You can start a change tracking task only after the task passes the precheck. If the task fails to pass the precheck, click the  icon next to each failed item to view details. Troubleshoot the issues based on the causes and run a precheck again.

19.5.4.3. Create a consumer group

You can manage consumer groups of a change tracking task in the DTS console. This topic describes how to create a consumer group.

Prerequisites

A change tracking task is created. For more information, see [Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance \(new version\)](#) or [Track data changes from a PolarDB-X instance](#).

Note

- You can create multiple consumer groups (up to 20) in a change tracking instance to repeatedly consume data.
- A consumer group consumes each message only once, and only one consumer can consume data.

Procedure

- [Log on to the DTS console](#).
- In the left-side navigation pane, click **Change Tracking**.
- Find the change tracking task and click the task ID.
- In the left-side navigation pane, click **Consume Data**.
- On the **Consume Data** page, click **Add Consumer Group** in the upper-right corner.
- In the dialog box that appears, set the parameters for the consumer group.

Parameter	Description
Consumer Group Name	Enter a new name for the consumer group. We recommend that you use an informative name for easy identification.
Username	Enter the username of the consumer group. <ul style="list-style-type: none"> A username must contain one or more of the following character types: uppercase letters, lowercase letters, digits, and underscores (_). The username must be 1 to 16 characters in length.
Password	Enter the password that corresponds to the username of the consumer group. <ul style="list-style-type: none"> A password must contain two or more of the following character types: uppercase letters, lowercase letters, digits, and special characters. The password must be 8 to 32 characters in length.
Confirm Password	Enter the new password again.

- Click **Create**.

19.5.4.4. Manage consumer groups

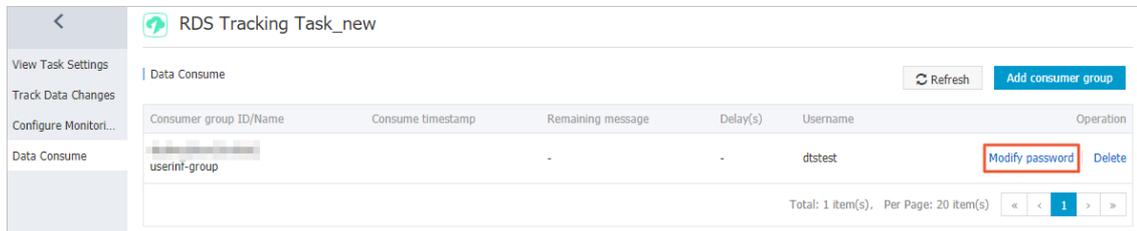
You can manage consumer groups of a change tracking task in the DTS console. This topic describes how to modify the password of a consumer group and how to delete a consumer group.

Prerequisites

[Create a consumer group](#)

Procedure

1. **Log on to the DTS console.**
2. In the left-side navigation pane, click **Change Tracking**.
3. Find the change tracking task and click the task ID.
4. In the left-side navigation pane, click **Consume Data**.
5. Modify the password of a consumer group or delete a consumer group. Modify the password of a consumer group
 - i. On the **Consume Data** page, find the target consumer group and click **Modify Password** in the **Actions** column.



- ii. In the **Modify Password** dialog box that appears, enter the **old password** and **new password**, and enter the **new password** again in the **Confirm Password** field.

Note

- A password must contain two or more of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- The password must be 8 to 32 characters in length.

- iii. Click **Modify**.

Delete a consumer group

Note After a consumer group is deleted, the data in the group will be cleared and cannot be recovered. We recommend that you use caution when performing this operation.

- i. On the **Consume Data** page, find the target consumer group and click **Delete** in the **Actions** column.
 - ii. In the **Delete Consumer Group** message that appears, click **OK**.

19.5.5. Use the SDK to consume tracked data

19.5.5.1. Methods provided by SDK

You can use the SDK demo code that is provided by DTS to consume tracked data. This topic describes the methods that are available for the SDK classes.

Methods of the RegionContext class

Method	Description
--------	-------------

Method	Description
<code>setAccessKey(accessKey)</code>	Specifies the AccessKey ID of the Alibaba Cloud account to which the source instance belongs.
<code>setSecret(AccessKeySecret)</code>	Specifies the AccessKey secret of the Alibaba Cloud account to which the source instance belongs.
<code>setUsePublicIp(usePublicIp)</code>	Specifies whether to track data changes over the Internet. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p>? Note DTS can track data changes only over the Internet. Therefore, set the usePublicIp parameter to <code>true</code>.</p> </div>
<code>context.setUseBinary(boolean useBinary)</code>	Specifies whether to enable the binary packaging feature. Valid values: True and False. We recommend that you enable this feature to improve consumption performance.
<code>context.setUseDrcNet(boolean useDrcNet)</code>	Specifies whether to enable the network optimization feature. Valid values: True and False. We recommend that you enable this feature to improve consumption performance.

Methods of the ClusterClient class

Method	Description
<code>void addConcurrentListener(ClusterListener arg0)</code>	Adds a downstream listener to retrieve data changes from a change tracking instance. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p>? Note The <code>ClusterListener arg0</code> parameter specifies an object of the <code>ClusterListener</code> class.</p> </div>
<code>void askForGUID(String arg0)</code>	Retrieves data changes from a change tracking instance. Set the String arg0 parameter to the ID of the change tracking instance.
<code>List<ClusterListener> getConcurrentListeners()</code>	Queries the list of listeners in a ClusterClient object. The return type is <code>List<ClusterListener></code> .
<code>void start()</code>	Starts the SDK client to start change tracking.
<code>void stop()</code>	Stops the SDK client to stop change tracking. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p>? Note Data pulling and notification callback are performed in the same thread of the SDK client. If the consumption code of the notify() method contains a function that prevents signal interruptions, the stop() function may fail to terminate the SDK client.</p> </div>

Methods of the ClusterListener class

The `void notify(List<ClusterMessage> arg0)` method specifies the consumption mode of tracked data. When the DTS SDK receives the data, it uses the notify() function to notify a ClusterListener object to consume the data. Then, the SDK displays the data on the screen.

Methods of the ClusterMessage class

Note Each ClusterMessage object stores the data record of a transaction. Each data record in the transaction is stored by using a Record object.

Method	Description
<code>Record getRecord()</code>	Retrieves a change record from a ClusterMessage object. The change record contains an entry in the binary log file, such as a BEGIN, COMMIT, UPDATE, or INSERT operation.
<code>void ackAsConsumed</code>	<p>After the data consumption is complete, you must call this method to send an ACK packet to instruct the DTS server to update the consumer offset. This ensures the integrity of the consumed data after an abnormal SDK client restarts.</p> <p>Note If a downstream SDK client restarts after a breakdown, the client resumes change tracking from the last consumer offset.</p>

Methods of the Record class

The `String getAttribute(String key)` method retrieves the attribute values in a Record object. The following table describes the parameters that are available when you call this method.

Parameter	Description
<code>record_id</code>	<p>The ID of the record.</p> <p>Note The record ID may not increment during the change tracking process.</p>
<code>instance</code>	The endpoint that is used to connect to the database instance. The format is <IP address>:<Port number>.
<code>source_type</code>	The engine type of the database instance. The value is set to MySQL.
<code>source_category</code>	The type of the record. The value is set to full_recorded.
<code>timestamp</code>	The binlog timestamp that is generated when the SQL statement is executed in the source database.
<code>checkpoint</code>	<p>The checkpoint of the binary log file. The format is <code>binlog_offset@binlog_file</code> .</p> <p>Note The <code>binlog_offset</code> parameter indicates the offset of a record in the binary log file. The <code>binlog_file</code> parameter indicates the numerical suffix of the binary log file. For example, if the name of a binary log file is <code>mysql-bin.0008</code>, the value of the <code>binlog_file</code> parameter is 8.</p>

Parameter	Description
<code>record_type</code>	The operation type. Valid values: insert, update, delete, replace, ddl, begin, commit, and heartbeat. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ? Note A heartbeat record indicates the heartbeat table that is defined by DTS. The system generates one heartbeat record per second to detect whether the change tracking instance is running as expected. </div>
<code>db</code>	The name of the database.
<code>table_name</code>	The name of the table.
<code>record_recording</code>	The encoding format.
<code>primary</code>	The name of the primary key column. If the primary key is a composite key, separate column names with commas (,).
<code>fields_enc</code>	The encoding of each field value. Separate fields with commas (,). <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ? Note If a field value is not of the character type, the encoding of this field value is null. </div>

The following table lists the methods that are preset in the SDK demo code. You can call these methods to retrieve the attribute values in a Record object.

Method	Description
Type <code>getOpt()</code>	Queries the operation type.
String <code>getCheckpoint()</code>	Queries the checkpoint of the binary log file.
String <code>getTimestamp()</code>	Queries the timestamp of the binary log file.
String <code>getDbname()</code>	Queries the database name.
String <code>getTablename()</code>	Queries the table name.
String <code>getPrimaryKeys()</code>	Queries the name of the primary key column.
DBType <code>getDbType()</code>	Queries the database type.
String <code>getServerId()</code>	Queries the endpoint that is used to connect to the database instance.
int <code>getFieldCount()</code>	Queries the number of fields.
List<Field> <code>getFieldList()</code>	Queries the definitions of all fields, the pre-change image values, and the post-change image values. For more information, see Methods of the Field class .
Boolean <code>isFirstInLogevent()</code>	Checks whether the record is the first transaction log in a large volume of data changes. The return value is True or False.

Methods of the Field class

Method	Description
<code>String getEncoding()</code>	Obtains the encoding format of the field value.
<code>String getFieldname()</code>	Queries the name of the field.
<code>Type getType()</code>	Queries the data type of the field.
<code>ByteString getValue()</code>	Queries the value of the field. The return type is <code>ByteString</code> . If the field is not specified, the method returns <code>NULL</code> .
<code>Boolean isPrimary()</code>	Checks whether the field is a primary key column. The return value is <code>True</code> or <code>False</code> .

19.5.5.2. Quick start

This section describes how to use the DTS Java SDK to perform some basic operations.

Initialize a RegionContext object

A `RegionContext` object stores the settings of authentication credentials and network access mode. The following code shows how to initialize a `RegionContext` object.

```
import java.util.List;
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
public class MainClass
{
    public static void main(String[] args) throws Exception {
        // Create a RegionContext object.
        RegionContext context = new RegionContext();
        context.setAccessKey("<AccessKey>");
        context.setSecret("<AccessKeySecret>");
        context.setUsePublicIp(true);
        // Create a ClusterClient object.
        final ClusterClient client = new DefaultClusterClient(context);
        // Other invocation code.
        ...
    }
}
```

Initialize a Listener object

Data consumption is implemented by using an object of the `Listener` class. After you initialize the `ClusterClient` object, you must add a `Listener` object. The `Listener` object uses the `notify()` method to receive and consume the tracked data. The following code shows how to display the tracked data on the screen.

```
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.ClusterListener;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
import com.aliyun.drc.clusterclient.message.ClusterMessage;
public class MainClass
{
    public static void main(String[] args) throws Exception {
        // Initialize the RegionContext object.
        ...
        //Initialize the ClusterClient object.
        ...
        ClusterListener listener = new ClusterListener(){
            @Override
            public void notify(List<ClusterMessage> messages) throws Exception {
                for (ClusterMessage message : messages) {
                    // Display the tracked data on the screen.
                    System.out.println(message.getRecord() + ":" + message.getRecord().getTablename() + ":"
                    + message.getRecord().getOpt());
                    // Call the following method to send an ACK packet to the DTS server.
                    message.ackAsConsumed();
                }
            }
        }
    }
}
```

DTS saves the consumption checkpoints of the SDK to the DTS server. This simplifies disaster recovery during the use of the SDK. The `ackAsConsumed()` method sends the checkpoint and timestamp of the latest data record that was consumed by the DTS SDK to the DTS server. If the SDK restarts due to an error, the SDK obtains the consumption checkpoint from the DTS server. The SDK resumes data consumption from the checkpoint. This ensures that the SDK does not consume duplicate data.

Start the ClusterClient object

Use the following code:

```

import java.util.List;
import com.aliyun.drc.clusterclient.ClusterClient;
import com.aliyun.drc.clusterclient.ClusterListener;
import com.aliyun.drc.clusterclient.DefaultClusterClient;
import com.aliyun.drc.clusterclient.RegionContext;
import com.aliyun.drc.clusterclient.message.ClusterMessage;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class MainClass
{
    public static void main(String[] args) throws Exception {
        // Initialize the RegionContext object.
        ...
        // Initialize the ClusterClient object.
        ...
        // Initialize the ClusterListener object.
        ...
        // Add a Listener class.
        client.addConcurrentListener(listener);
        // Specify the ID of the change tracking instance.
        client.askForGUID("dts_rdsrjiei2u2afnb_DSF");
        // Start a background thread. The main thread cannot exit.
        client.start();
    }
}

```

The `askForGUID()` method sets the ID of the change tracking instance. You can obtain the ID of the change tracking instance from the DTS console. After the ID of the change tracking instance is specified in the `askForGUID()` method, the SDK retrieves incremental data from this instance.

Before you can start a `ClusterClient` object, you must add a `Listener` class to the `ClusterClient` object. When the `ClusterClient` object pulls incremental data from the change tracking instance, it also calls the `notify()` method of the `Listener` class to consume data.

19.5.5.3. Parse tracked SQL statements

You can use the DTS SDK to track data changes. DTS records the tracked data changes in a custom format. This topic describes how to parse various types of SQL statements.

Parse a DDL statement

If a data definition language (DDL) operation is performed in the source database, the operation type of the data record is DDL. The DDL statement is stored in the value of the first column. You can use the following sample code to parse the DDL statement:

```

String ddl_string;
Record.Type type=record.getOpt();
if(type.equals(Record.Type.DDL)){
    List<DataMessage.Record.Field> fields = record.getFieldList();
    ddl_string = fields.get(0).getValue().toString();
}

```

Parse an INSERT statement

If an INSERT operation is performed in the source database, the operation type of the data record is INSERT. You can use the following sample code to parse the INSERT statement:

```
StringBuilder insert_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
StringBuilder fieldName=new StringBuilder();
StringBuilder fieldValue = new StringBuilder();
if(type.equals(Record.Type.INSERT)){
    int i=0;
    List<DataMessage.Record.Field> fields = record.getFieldList();
    for (; i < fields.size(); i++) {
        field = fields.get(i);           fieldName.append("`'+field.getFieldname().toLowerCase()+`'");
        fieldValue.append(""+field.getValue()+"");
        if (i != fields.size() - 1) {
            fieldName.append(',');
            fieldValue.append(',');
        }
    }
    insert_string.append("insert "+ record.getTablename()+"("+fieldName.toString()+") values("+fieldValue.toString()+");");
}
```

Parse an UPDATE statement

If an UPDATE operation is performed in the source database, the operation type of the data record is UPDATE. The field values prior to the UPDATE operation are stored in `Record.getFieldList()` entries with even indexes. The field values after the UPDATE operation are stored in `Record.getFieldList()` entries with odd indexes.

If the UPDATE operation is performed on a table that has a primary key, you can use the following sample code to parse the UPDATE statement:

```
StringBuilder update_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
StringBuilder SetValue = new StringBuilder();
StringBuilder WhereCondition = new StringBuilder();
String ConditionStr;
boolean hasPk=false;
boolean pkMode=false;
boolean hasSet=false;
if(type.equals(Record.Type.UPDATE)){
    int i=0;
    DataMessage.Record.Field OldField = null;
    DataMessage.Record.Field NewField = null;
    List<DataMessage.Record.Field> fields = record.getFieldList();
    for (; i < fields.size() ; i++) {
        if (i % 2 == 0) {
```

```

        OldField = fields.get(i);
        continue;
    }
    NewField = fields.get(i);
    field = NewField;
    if (field.isPrimary()) {
        if (hasPk) {
            WhereCondition.append(" and ");
        }
        //where old value
        ConditionStr = getFieldValue(OldField);
        if(ConditionStr==null){
            WhereCondition.append(""+field.getFieldname().toLowerCase()+" "+
+ " " + "is null");
        }else{
            WhereCondition.append(""+field.getFieldname().toLowerCase()+" "+" = " + ""+OldField.getValue()+"");
        }
        hasPk = true;
    }
    if (hasSet) {
        SetValue.append(",");
    }
    SetValue.append(""+field.getFieldname().toLowerCase()+" "+" = " + ""+field.getValue()+"");
    String setStr = getFieldValue(field);
    hasSet = true;
}
update_string.append("Update "+record.getTablename() +" Set " + SetValue + " Where "+WhereCondition +";");
}

protected String getFieldValue(Field field) throws Exception {
    ByteString byteString = field.getValue();
    if (byteString == null) {
        return null;
    }
    else {
        String value;
        if (field.getType() == com.aliyun.drc.client.message.DataMessage.Record.Field.Type.STRING && field.getEncoding()
!= null && field.getEncoding() != "ASCII") {
            value = field.getValue().toString(field.getEncoding());
        }
        else {
            value = byteString.toString();
        }
        return value;
    }
}
}

```

Parse a DELETE statement

If a DELETE operation is performed in the source database, the operation type of the data record is DELETE. If the DELETE operation is performed on a table that has a primary key, you can use the following sample code to parse the DELETE statement:

```
Stringbuilder delete_string=new StringBuilder();
Record.Type type=record.getOpt();
DataMessage.Record.Field field;
Stringbuilder FieldName=new StringBuilder();
Stringbuilder FieldValue = new StringBuilder();
Stringbuilder DeleteCondition = new StringBuilder();
boolean hasPk=false;
boolean pkMode=false;
if(type.equals(Record.Type.DELETE)){
    int i=0;
    List<DataMessage.Record.Field> fields = record.getFieldList();
    delete_string.append("Delete From" + record.getTablename() + "where");
    // Check whether the table has a primary key.
    if (record.getPrimaryKeys() != null) {
        pkMode = record.getPrimaryKeys().length() > 0 ? true : false;
    }
    for (; i < fields.size(); i++) {
        if ((pkMode && ! field.isPrimary())) {
            continue;
        }
        if (hasPk) {
            delete_string.append(" and ");
        }
        delete_string.append(field.getFieldname() + "=" + field.getValue());
        hasPk = true;
    }
    delete_string.append(";");
}
```

Parse a REPLACE statement

If a REPLACE operation is performed in the source database, the operation type of the data record is UPDATE or INSERT.

- If the value specified in the REPLACE statement does not exist, the operation type of the data record is INSERT.
- If the value specified in the REPLACE statement exists, the operation type of the data record is UPDATE.

Parse a BEGIN statement

If a BEGIN operation is performed in the source database, the operation type of the data record is BEGIN. You do not need to perform operations on fields because the BEGIN statement does not modify fields. You only need to check that the operation is a BEGIN operation. You can use the following sample code to parse the BEGIN statement:

```
StringBuilder sql_string = new StringBuilder();
Record.Type type = record.getOpt();
if(type.equals(Record.Type.BEGIN)){
    sql_string.append("Begin");
}
```

Parse a COMMIT statement

If a COMMIT operation is performed in the source database, the operation type for the data record is COMMIT. You do not need to perform operations on fields because the COMMIT statement does not modify fields. You only need to check that the operation is a COMMIT operation. You can use the following sample code to parse the COMMIT statement:

```
StringBuilder sql_string = new StringBuilder();
Record.Type type = record.getOpt();
if(type.equals(Record.Type.COMMIT)){
    sql_string.append("commit");
}
```

19.5.5.4. Run the SDK demo code

This section describes how to run the demo code provided by the DTS console.

1. Create an AccessKey.

Your account must pass the AccessKey authentication before you can use an SDK to connect to a subscription channel. Therefore, before using the SDK, you must obtain an AccessKey. For more information, see the "Obtain an AccessKey" section of the *DTS Developer Guide*.

2. Install the Java SDK.

The development environment supported by the DTS Java SDK is J2SE Development Kit (JDK) V1.5 or later.

For an Eclipse project, you can follow these steps to install the Java SDK:

- i. Click **View Example Code** and download the SDK package *consumer.jar*.
- ii. Import the JAR package to an Eclipse project as follows:

In Eclipse, right-click your project and choose **Properties > Java Build Path > Libraries > Add External JARs**. Select the path for storing the *consumer.jar* package *consumer.jar*.

- iii. Select the *consumer.jar* package and click **OK**.

Then you can use the DTS Java SDK in the project.

3. Run the demo code.

DTS provides the SDK demo code. You can copy the demo code by using the **View Demo Code** option in the DTS console. For an Eclipse project, you can follow these steps to run the demo code:

- i. Create a class named *MainClass* in the *src* directory of the Eclipse project.
- ii. Open the generated Java file *MainClass* and delete the code template.
- iii. Paste the demo code into the *MainClass* file.
- iv. Modify the *AccessKeyId*, *AccessKeySecret*, and subscription channel ID in the demo code.

Change the marked parts in the preceding demo code to the *AccessKeyId*, *AccessKeySecret*, and subscription channel ID of your account.

You can obtain the subscription channel ID from the [DTS console](#).

- v. In Eclipse, right-click the demo file and choose **Run as > Java Application** to run the demo code.

19.5.6. Use a Kafka client to consume tracked data

This topic describes how to use the demo code of a Kafka client to consume tracked data. The change tracking feature of the new version allows you to consume tracked data by using a Kafka client from V0.11 to V1.1.

Prerequisites

- A change tracking task is created. For more information, see [Track data changes from a user-created MySQL database or an ApsaraDB RDS for MySQL instance \(new version\)](#) or [Track data changes from a PolarDB-X instance](#).
- One or more consumer groups are created. For more information, see [Create a consumer group](#).

Precautions

- If you enable auto commit when you use the change tracking feature, some data may be committed before it is consumed. This results in data loss. We recommend that you manually commit data.

 **Note** If data fails to be committed due to a fault, you can restart the client to continue consuming data from the last recorded consumer offset. However, duplicate data may be generated during this period. You must manually filter out the duplicate data.

- Data is serialized and stored in the Avro format. For more information, see [Record.avsc](#).

 **Note** If the client that you use is not a Kafka client, you must parse the tracked data based on the Avro schema.

- Regarding the `offsetFotTimes` interface, the search unit of DTS is seconds, and the search unit of native Kafka is milliseconds.

Download and run the demo code of the Kafka client

Click [here](#) to download the demo code of the Kafka client. For more information about how to use the demo code, visit [Readme](#).

Download and run the demo code of the Kafka client

Step	File or directory
1. Use the native Kafka consumer to obtain incremental data from the change tracking instance.	subscribe_example-master/javaimpl/src/main/java/reco rdgenerator/
2. Deserialize the image of the incremental data, and obtain attributes such as the pre-image and post-image.	subscribe_example-master/javaimpl/src/main/java/boot /MysqlRecordPrinter.java
3. Convert the dataTypeNumber values in the deserialized data into MySQL or Oracle data types.	subscribe_example-master/javaimpl/src/main/java/reco rdprocessor/mysql/

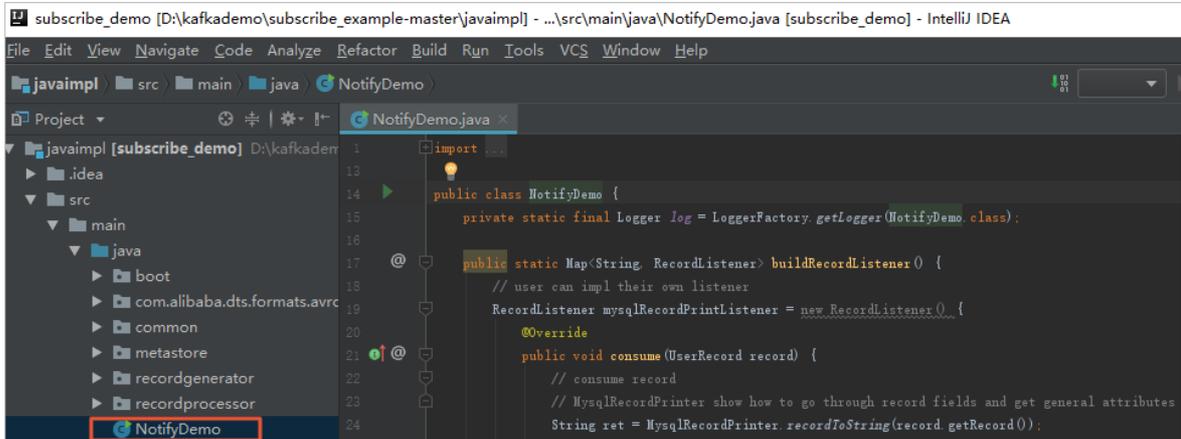
 **Note** For more information, see [Mappings between MySQL data types and dataTypeNumber values](#) and [Mappings between Oracle data types and dataTypeNumber values](#).

Procedure

This procedure uses IntelliJ IDEA (Community Edition 2018.1.4 Windows) as an example.

1. Download the [demo code of the Kafka client](#), and then decompress the package.
2. Open IntelliJ IDEA. In the window that appears, click **Open**.

- In the dialog box that appears, go to the directory in which the downloaded demo code resides. Find the `pom.xml` file.
- In the dialog box that appears, select **Open as Project**.
- On the IntelliJ IDEA page, expand folders to find the demo file of the Kafka client, and double-click the file. The file name is `NotifyDemo.java`.



- Set the parameters in the `NotifyDemo.java` file.

Parameter	Description	Method to obtain
USER_NAME	<p>The username of the consumer group.</p> <p>Warning If you are not using the Kafka client that is described in this topic, you must specify the username in the following format: <code><Consumer group account>-<Consumer group ID></code>, for example, <code>dtstes-t-dtsae*****bpv</code>. Otherwise, the connection fails.</p>	<p>In the DTS console, click the instance ID, and then click Data Consume. You can obtain the Consumer Group ID and the corresponding Account information.</p> <p>Note The password of the consumer group account is specified when you create a consumer group.</p>
PASSWORD_NAME	The password of the account.	
SID_NAME	The ID of the consumer group.	
GROUP_NAME	The name of the consumer group. Set this parameter to the consumer group ID.	
KAFKA_TOPIC	The topic of the change tracking task.	
KAFKA_BROKER_URL_NAME	<p>The network address and port number of the change tracking task.</p> <p>Note If you track data changes over internal networks, the network latency is minimal. This is applicable if the ECS instance where you deploy the Kafka client belongs to the same VPC or classic network as the change tracking instance.</p>	<p>In the DTS console, click the instance ID. On the Track Data Changes page, you can obtain the tracked topic, network address, and port number.</p>

Parameter	Description	Method to obtain
INITIAL_CHECKPOINT_NAME	<p>The consumer offset of consumed data. The value is a UNIX timestamp.</p> <p>Note You must save the consumer offset. If the consumption process is interrupted, you can specify the consumer offset on the change tracking client to resume data consumption. This allows you to prevent against data loss. When you start the change tracking client, you can specify the consumer offset to consume data on demand.</p>	When you use the Kafka client to track data changes for the first time, convert the required time point into a UNIX timestamp.
USE_CONFIG_CHECKPOINT_NAME	<p>Default value: <i>true</i>. The default value indicates that the client is forced to consume data from the specified consumer offset. This allows you to retain the data that is received but not processed.</p>	None.

7. On the top of the IntelliJ IDEA page, choose **Run > Run** to run the client.

Note When you run IntelliJ IDEA for the first time, it loads and installs the relevant dependency.

Execution result

The following figure shows that the Kafka client can track data changes in the source database.

```

[2020-03-09 10:41:52.408] INFO [Consumer clientId=consumer-1, groupId=dts-...] Discovered coordinator [redacted] rack: null (org.apache.kafka.clients.consumer.internals.AbstractCoordinator)
[2020-03-09 10:41:57.203] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721711, offset: 1732521, info: 1583721711} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:41:57.511] INFO BtlRecordProcessor haven't receive records from generator for 5s (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:02.203] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721721, offset: 1732539, info: 1583721721} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:07.204] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721726, offset: 1732544, info: 1583721726} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:12.205] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721731, offset: 1732548, info: 1583721731} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:17.205] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721736, offset: 1732554, info: 1583721736} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:22.205] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721741, offset: 1732559, info: 1583721741} (recordprocessor.BtlRecordProcessor)
[2020-03-09 10:42:27.206] INFO commit record with checkpoint Checkpoint{ topicPartition: cn_hangzhou_re..., dtstest-timestamp: 1583721746, offset: 1732569, info: 1583721746} (recordprocessor.BtlRecordProcessor)
    
```

You can also delete the `//` characters from the `//log.info(ret);` string in line 25 of the `NotifyDemo.java` file. Then, run the client again to view the data change information.

```

[2020-03-09 11:51:19.363] INFO recordID [1737005]source [{"sourceType": "MySQL", "version": "8.0.16"}]dbTable [dtstestdata.customer]recordType [UPDATE]recordTimestamp [1583725879]extra tags [{"pk_val_info="
Field [id]Before [10005]After [10005]
Field [name]Before [changma]After [changma]
Field [address]Before [hangzhou]After [beijing]
    
```

Mappings between MySQL data types and dataTypeNumber values

MySQL data type	Value of dataTypeNumber
MYSQL_TYPE_DECIMAL	0
MYSQL_TYPE_INT8	1
MYSQL_TYPE_INT16	2
MYSQL_TYPE_INT32	3
MYSQL_TYPE_FLOAT	4

MySQL data type	Value of dataTypeNumber
MYSQL_TYPE_DOUBLE	5
MYSQL_TYPE_NULL	6
MYSQL_TYPE_TIMESTAMP	7
MYSQL_TYPE_INT64	8
MYSQL_TYPE_INT24	9
MYSQL_TYPE_DATE	10
MYSQL_TYPE_TIME	11
MYSQL_TYPE_DATETIME	12
MYSQL_TYPE_YEAR	13
MYSQL_TYPE_DATE_NEW	14
MYSQL_TYPE_VARCHAR	15
MYSQL_TYPE_BIT	16
MYSQL_TYPE_TIMESTAMP_NEW	17
MYSQL_TYPE_DATETIME_NEW	18
MYSQL_TYPE_TIME_NEW	19
MYSQL_TYPE_JSON	245
MYSQL_TYPE_DECIMAL_NEW	246
MYSQL_TYPE_ENUM	247
MYSQL_TYPE_SET	248
MYSQL_TYPE_TINY_BLOB	249
MYSQL_TYPE_MEDIUM_BLOB	250
MYSQL_TYPE_LONG_BLOB	251
MYSQL_TYPE_BLOB	252
MYSQL_TYPE_VAR_STRING	253
MYSQL_TYPE_STRING	254
MYSQL_TYPE_GEOMETRY	255

Mappings between Oracle data types and dataTypeNumber values

Oracle data type	Value of dataTypeNumber
VARCHAR2/NVARCHAR2	1
NUMBER/FLOAT	2

Oracle data type	Value of dataTypeNumber
LONG	8
DATE	12
RAW	23
LONG_RAW	24
UNDEFINED	29
XMLTYPE	58
ROWID	69
CHAR and NCHAR	96
BINARY_FLOAT	100
BINARY_DOUBLE	101
CLOB/NCLOB	112
BLOB	113
BFILE	114
TIMESTAMP	180
TIMESTAMP_WITH_TIME_ZONE	181
INTERVAL_YEAR_TO_MONTH	182
INTERVAL_DAY_TO_SECOND	183
UROWID	208
TIMESTAMP_WITH_LOCAL_TIME_ZONE	231

20.Data Management (DMS)

20.1. What is Data Management?

Data Management (DMS) is an integrated database solution that provides data, schema, and server management, access control, business intelligence (BI) insights, data trend analysis, data tracking, and performance optimization.

Supported database types

- ApsaraDB RDS for MySQL
- ApsaraDB RDS for SQL Server
- PolarDB and ApsaraDB RDS for PostgreSQL

Supported database operations

- SQL operations
 - Use of SQL windows
 - Use of the command window
 - Saving of work environment settings
 - SQL statement execution
 - SQL statement optimization
 - SQL statement formatting and improvement
 - Viewing of execution plans
 - Smart SQL completion
- Operations on database objects
 - Operations on data tables
 - Operations on schemas: modify schemas by adding or deleting columns, indexes, foreign keys, and partitions
 - Changes to table data: insert, update, and delete
 - Table data query and visualized editing
- Operations on views and programmable objects such as functions, stored procedures, triggers, and events
 - Creation
 - Modification
 - Deletion
 - Enabling and disabling
- Data processing
 - Data import
 - Data export
- Performance and diagnosis
 - Lock wait analysis
- Use of data processing tools
 - Drawing of entity-relationship (ER) diagrams
 - Statistics on table data volume
 - Batch operations on tables

User-friendly interactions

DMS provides user-friendly tips. When an error occurs, DMS displays suggestions on how best to achieve your goal.

20.2. Log on to an ApsaraDB for RDS instance by using DMS

This topic describes how to log on to an ApsaraDB for RDS instance by using DMS.

Context

The ApsaraDB for RDS console allows you to log on to ApsaraDB for RDS instances by using DMS. You can use DMS to manage database data and schemas.

Procedure

1. Log on to the ApsaraDB for RDS console. For more information, see the *Log on to the ApsaraDB for RDS console* topic in *ApsaraDB for RDS User Guide*.
2. In the left-side navigation pane, click **Instances**. On the **Instances** page, find the target instance. Click the instance ID or click **Manage** in the **Actions** column.
3. On the page that appears, click **Log On to DB** to go to the logon page of the DMS console.
4. Enter the logon information.

Parameter	Description
IP address:Port	The internal or public endpoint and corresponding port number of the target ApsaraDB for RDS instance. Example: <code>rm-txxxxxxxxxxxxxxxxx.mysql.aliyun-inc.com:3306</code> . To obtain the internal or public endpoint and the port number, perform the following steps: <ol style="list-style-type: none"> Log on to the ApsaraDB for RDS console. Go to the Instances page, find the target instance, and then click the instance ID or click Manage in the Actions column. In the Basic Information section of the page that appears, obtain the endpoint and port number.
Database Username	The account used to connect to the ApsaraDB for RDS instance. <p> Note The account is created in the ApsaraDB for RDS instance.</p>
Enter your password	The password of the account used to connect to the ApsaraDB for RDS instance. <p> Note The password is that you specified when you create the account in the ApsaraDB for RDS instance.</p>

5. Click **Login**.

 **Note** If you want the browser to remember the password, select **Remember your password** and then click **Login**.

20.3. SQL operations

20.3.1. Use the command window

This topic describes how to use the command window in DMS.

Context

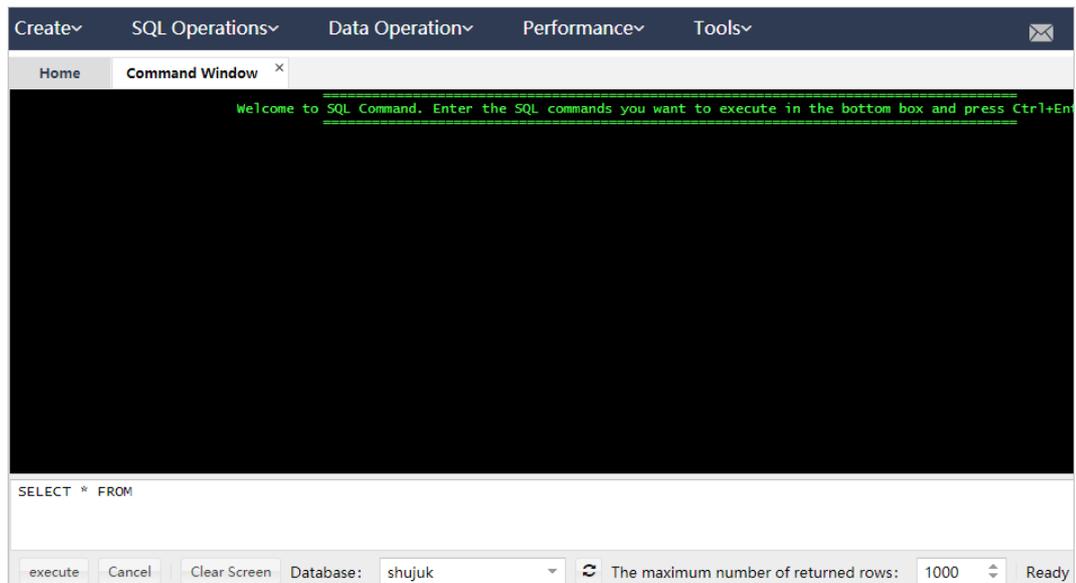
A MySQL database is used in this example.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the top navigation bar, choose **SQL Operations > Command Window**.

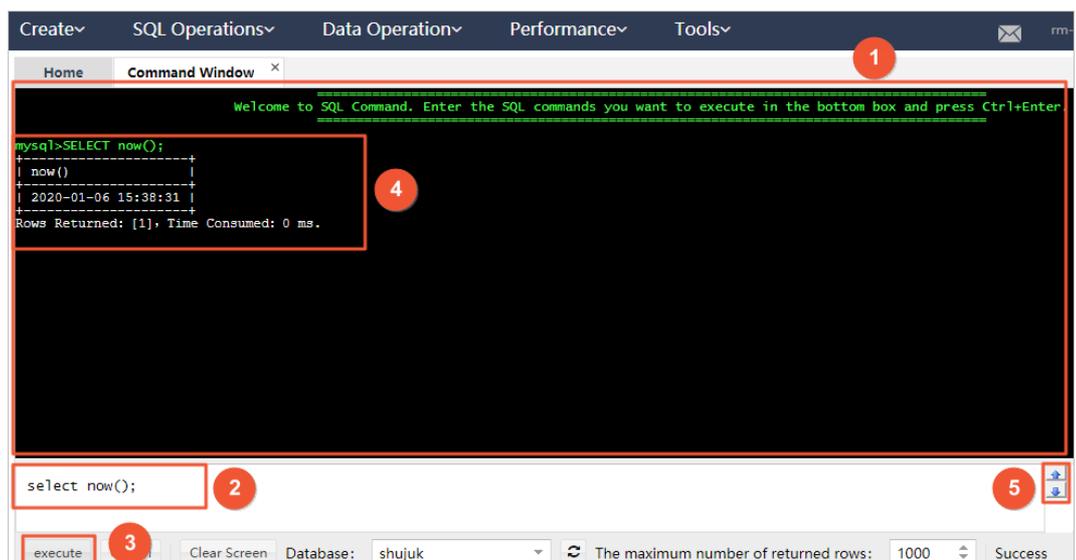
An empty command window appears, as shown in **Command window**.

Command window



3. Enter an SQL statement in the command window, and then click **execute**, as shown in **Execution of an SQL statement**.

Execution of an SQL statement



The following table describes the numbered items in the command window.

Numbered items in the command window

No.	Name	Description
1	Command window	The area displays execution results of SQL statements.
2	SQL statement input area	You can enter SQL statements in this area.
3	execute button	You can click this button to execute the entered SQL statements.
4	Result display area	DMS appends execution results to the result display area.
5	Up and down arrows	You can click the up or down arrow to view an executed SQL statement and execute it again.

- (Optional) If the execution process takes longer than expected, you can click **Cancel** to abort the execution.
- (Optional) Click **Clear Screen** to clear the results for proper display of subsequent results.

To switch to another database, select the new database from the Database drop-down list.

20.3.2. Use the SQL window

20.3.2.1. Open an empty SQL window

This topic describes how to use SQL windows.

Context

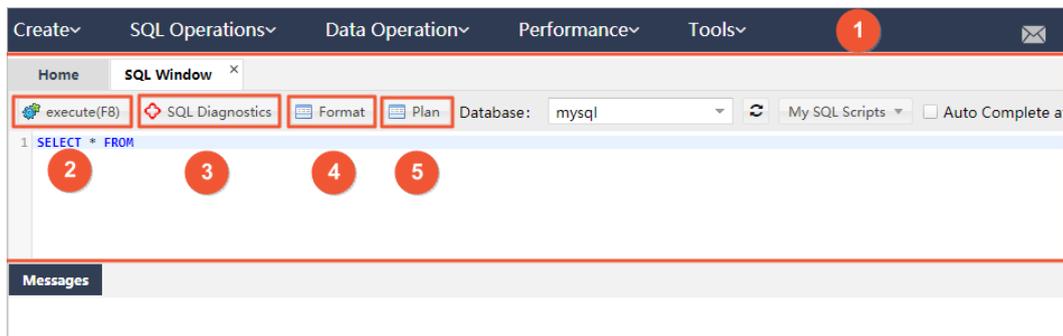
- A MySQL database is used in this example.
- A maximum of 20 SQL windows (including the homepage) can be opened in DMS. We recommend that you open no more than five SQL windows at a time.

Procedure

- Log on to an RDS instance through DMS.
- In the top navigation bar, choose **SQL Operations > SQL Window** to open the **SQL Window** tab.

Empty SQL window shows the empty SQL window you opened.

Empty SQL window



Numbered items in the SQL window describes the numbered items in the SQL window.

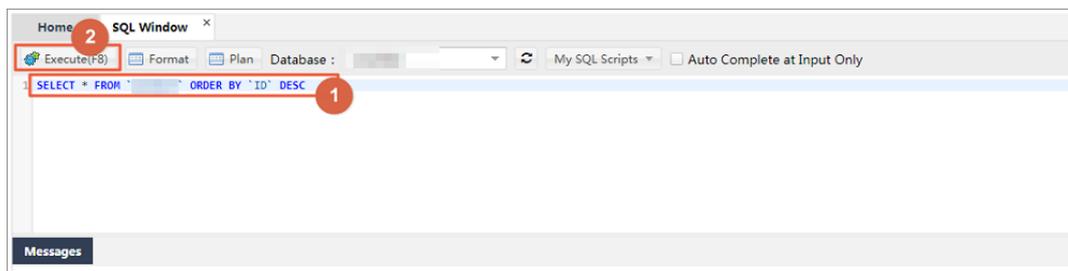
Numbered items in the SQL window

No.	Name	Description
-----	------	-------------

No.	Name	Description
1	SQL window	The green-framed area is the main body of the SQL window.
2	execute(F8) button	You can click this button to execute the entered SQL statements.
3	SQL Diagnostics button	You can click this button to diagnose the current SQL statements.
4	Format button	You can click this button to format the entered SQL statements for readability.
5	Plan button	You can click this button to display the execution plans of the selected SQL statements. You can optimize the SQL statements and improve SQL processing performance based on the execution plans.

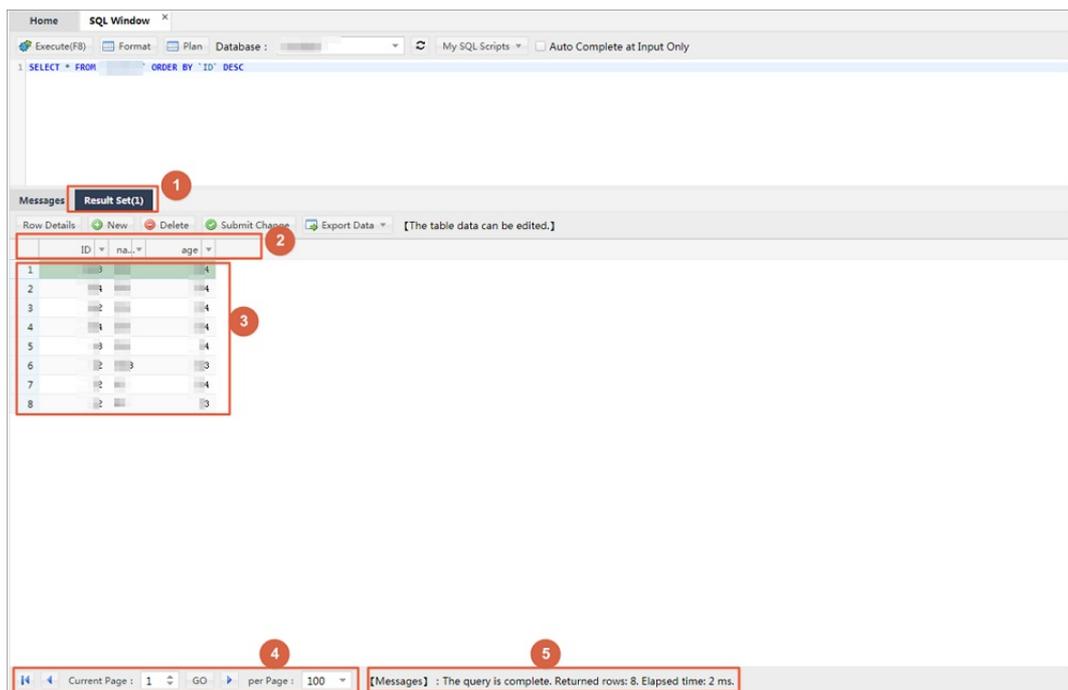
3. Enter the SQL statement that you want to execute and click execute(F8) to complete the SQL query or update, as shown in [Execute SQL statements.](#)

Execute SQL statements.



4. You can view the result set, as shown in [View the result set.](#)

View the result set



Numbered items in the result set

No.	Description
1	The Result Set tab displays the results returned by the SQL query statement.
2	The first row of the table shows the column names. If an alias has been specified for a column in the SQL statement, the alias is displayed in this table.
3	The data area of the table displays the queried data in rows. If the data area is not big enough to show all of the queried data, horizontal and vertical scroll bars will appear to help you navigate the result set.
4	<p>You can set Current Page or Per Page to view the result set.</p> <ul style="list-style-type: none"> Each page displays 100 data entries by default. Go to the next page to view more data. You can set the number of data entries displayed per page as needed. The obtained data entries on the next page are displayed in the data area numbered 3 in the figure.
5	You can view the progress of data acquisition and time elapsed.

5. View the SQL execution message.

Each time a data query (SELECT) or data correction (INSERT, UPDATE, or DELETE) statement is executed, DMS returns a message about the execution including the status and impact.

Data query shows the message returned for data query.

Data query



Data correction shows the message returned for data correction.

Data correction



Numbered items in the data correction window describes the numbered items in the data correction window.

Numbered items in the data correction window

No.	Description
1	After you execute an SQL statement, you can click the Messages tab to view the execution status. No result set is returned for data correction. DMS displays a message after data correction has been completed.
2	DMS executes the entered SQL statements step by step: <ul style="list-style-type: none"> ○ Analyzes the entered SQL statements. ○ Executes the SQL statements in the database. ○ Displays the queried data. ○ Displays statistics, including the number of data rows that are queried or affected.
3	DMS displays the SQL execution results: <ul style="list-style-type: none"> ○ Whether the execution is successful. ○ Number of queried rows, or number of rows affected by the Add, Delete, or Modify operation. ○ The amount of time consumed to execute the SQL statements.

6. Execute SQL statements in batches.

DMS supports batch execution of SQL statements, as shown in [Batch execution](#).

Batch execution

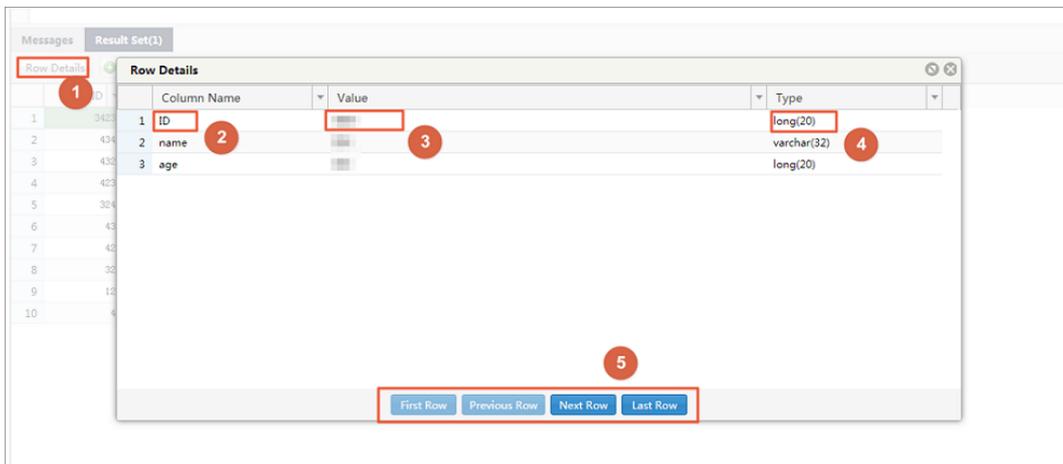


- o 1: Displays the execution result of the first SQL statement.
- o 2: Displays the execution result of the second SQL statement.
 - i. Separate each SQL statement with a semicolon (;) or another separator.
 - ii. To execute only some SQL statements, select the SQL statements that you want to execute. To execute all SQL statements, deselect or select all SQL statements, and click **execute(F8)**.
Wait until all SQL statements have been executed.
 - iii. View the execution results.

If you execute the SELECT statement, DMS displays the result set. If you execute other statements, DMS displays the execution results, such as the number of affected rows.

7. Click **Row Details** to view the details of a single entry in the result set, as shown in **Row details**.

Row details



The following table describes the numbered items in the **Row Details** dialog box.

Numbered items in the Row Details dialog box

No.	Description
1	On the Result Set tab, select a single data entry that you want to display, and click Row Details to view details of the entry. The Row Details dialog box displays the name, value, and type of each column of the data entry.

No.	Description
2	Column Name: If you have specified an alias for a column, the alias is displayed.
3	Value: DMS automatically parses and displays the column values. Data such as time and binary code is formatted as a string to ease reading.
4	Type: You can view the type and length of each column.
5	The navigation area. You can click Previous Row, Next Row, First Row, and Last Row to view the details of the corresponding data entry.

8. (Optional) Edit the queried data in the result set.

- Click **New** to add a row of data to the currently queried table.
- Click **Delete** to delete the selected row of data from the result set table.
- Select the row that you want to perform operations on.
- Update the column values in the selected row directly.

After you modify data, click **Submit Change** to save the modified data to the database.

After you click **Submit Change**, DMS displays the SQL statements required to save your changes. This allows you to confirm the changes and prevent misoperations that can result in data loss.

Click **OK** to apply the changes to the database as expected.

9. Click **Format** to format the selected SQL statements to make them easier to read and write.

- Only the selected SQL statements are formatted. If you do not select any SQL statements, all entered SQL statements will be formatted.
- This feature transforms your SQL statements into standard, easy-to-read statements without changing the SQL execution logic and syntax or affecting the execution.

Example:

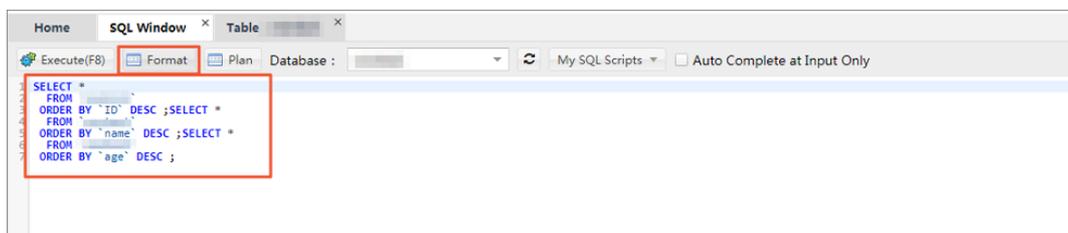
Original SQL statements shows the original SQL statements.

Original SQL statements



Formatted SQL statements shows the formatted SQL statements.

Formatted SQL statements



10. Click **Plan** to view the execution plan if you want to troubleshoot SQL-related problems or optimize SQL performance.

- After you click **Plan**, DMS displays the execution plan of the selected SQL statement. If no SQL statement is selected, DMS displays the execution plans of all SQL statements.

- DMS displays an execution plan in detail. You can view information such as the execution plan type and possible keys.
 - Different databases display execution plans in different ways and in varying levels of detail.
 - If you want to view the execution plans of several SQL statements, DMS displays the execution plan of each SQL statement in detail on different tabs, as shown in [Plan](#).

Plan

ID	SELECT_TYPE	TABLE	TYPE	POSSIBLE_KEYS	KEY	KEY_LEN	REF	ROWS
1	SIMPLE	index	PRIMARY			8		

- 1: Shows the execution plan of the first SQL statement in detail.
- 2: Shows the execution plan of the second SQL statement in detail.

20.3.2.2. Restore a saved SQL window

This topic describes how to restore a saved SQL window.

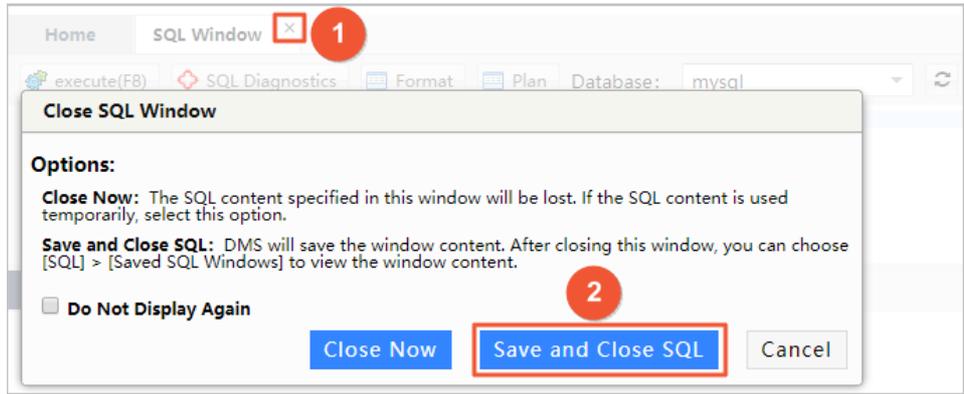
Context

- A MySQL database is used in this example.
- A maximum of 20 SQL windows (including the homepage) can be opened or stored in DMS. We recommend that you open no more than five SQL windows at a time.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **SQL Operations > SQL Window**.
3. Save the current SQL work environment.
 - DMS automatically saves the work environment when you close the operation page.
 - The next time that you log on to the DMS console, DMS automatically restores the last saved work environment, including the last used database, the open SQL windows, and the SQL statements entered in the SQL windows.
 - When you close a SQL window, DMS prompts you to confirm whether to save the content of the window, as shown in [Confirm to close a SQL window](#).

Confirm to close a SQL window

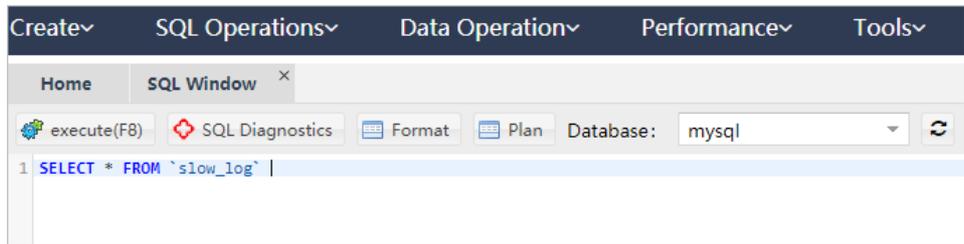


- 1: Click the Close icon in the upper-right corner of the SQL window to close the window.
- 2: DMS prompts you to confirm whether to save the content. Click **Save and Close SQL**. DMS saves the content of the SQL window before it is closed.

If you click **Close Now**, DMS does not save the content of the SQL window.

4. Restore a saved SQL window.
 - i. Choose **SQL Operations > Saved SQL Windows**.
DMS displays all saved SQL windows.
 - ii. Click **View SQL** next to a saved SQL window to restore the SQL window.
 - iii. When you log on to the database through DMS, DMS automatically restores the most recently saved content of the SQL window, as shown in [Restored SQL window](#).

Restored SQL window



20.3.2.3. Manage common SQL statements

This topic describes how to manage common SQL statements in DMS.

Context

A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS](#).
2. In the top navigation bar, choose **SQL Operations > SQL Window** to open the **SQL Window** tab.
3. Perform the following operations:
 - Add a common SQL statement.

Choose **My SQL Scripts > Add SQL** to add a common SQL statement.

The applicable scope is described as follows:

- **Current Instance:** You can only view the custom SQL statement through the instance (with the specified IP address and port number) that is currently connected to.

- **Current Database:** You can only view the custom SQL statement through the database that is currently connected to. If you switch the current database to another one, you cannot view the custom SQL statement by choosing **My SQL Scripts > Select SQL**.
- View saved SQL statements.
To view saved SQL statements, choose **My SQL Scripts > Select SQL**.
- Manage SQL statements.
To manage your SQL statements, choose **My SQL Scripts > Manage SQL**.
 - In the **Manage SQL** dialog box that appears, click **Edit** or **Delete** to edit or delete SQL statements.
 - In the **Manage SQL** dialog box, click **New** to add an SQL statement.
 - Double-click an SQL statement to insert the statement into **SQL Window**. The statement appears in the SQL window and is selected.

20.3.2.4. Use the SQL template

This topic describes how to use the SQL template in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the top navigation bar, choose **SQL Operations > SQL Window**. A SQL window appears.
The SQL template is displayed in the rightmost part of the SQL window.
3. Double-click an SQL command or drag it into the SQL window. Then you can use or reference the command.
You can directly modify the commands referenced from the template even if you are not familiar with the commands.

20.3.3. Table operations (based on the Table directory tree)

20.3.3.1. Open a table-based SQL window

This topic describes how to open a table-based SQL window in DMS.

Context

A MySQL database is used as an example.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side directory tree, right-click a table and choose **SQL Operation Data** from the shortcut menu to open an SQL window. DMS automatically runs the SQL statement that queries top 50 records of the table.

20.3.3.2. Edit table data

This topic describes how to edit table data.

Context

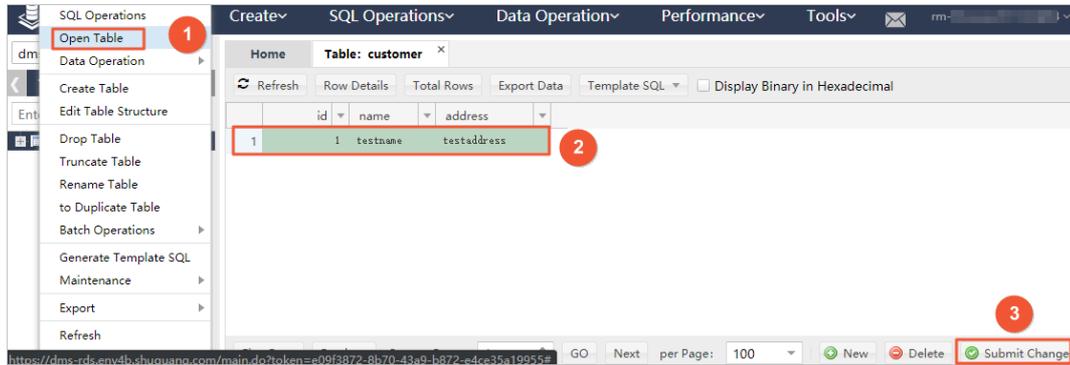
- An ApsaraDB RDS for MySQL database is used in this example.
- This feature applies to tables with a small volume of data. For tables containing a large volume of data, you must find the target data before you can edit the data. It may take time to find the data to be edited.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side navigation pane, right-click a table and select **Open Table**. A tab appears on the right,

displaying the data of the selected table, as shown in **Table tab**.

Table tab



- 1: Right-click a table and select **Open Table**.
- 2: Modify the values of the columns in the table.
- 3: Click **Submit Change** to submit the modified data.

20.4. Database development

20.4.1. Overview

This topic describes how to add, modify, delete, and manage objects such as indexes, foreign keys, and stored procedures.

20.4.2. Table

20.4.2.1. Create a table

This topic describes how to create a table in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. You can create a table in one of the following ways:
 - In the top navigation bar, choose **Create > Table**.
 - In the left-side navigation pane, click **Table**, right-click a table, and then select **Create Table**.
3. **Edit columns.**

On the **New: Table** tab, the **Edit Column** tab appears by default.

You can edit the basic information and extended information about the fields as needed.

You can also click **Columns** to edit the table information.
4. **Click Index** to edit indexes.
 - Click **New** to add an index.
 - Click **Remove** to delete an index.
 - Click the index row to modify index information.
5. **Click Foreign Key** to go to the **Edit Foreign Key** tab.
 - Click **New** to add a foreign key. The new key is editable.
 - Click **Remove** to delete a foreign key.
 - To edit a foreign key, enter the key name, column name, and information about the referenced database, table, and column.

6. Click **Partition** to edit the partition information. The SQL statements of the partition must be configured.
7. Click **Basic** to edit the basic information about the table.
 - You can edit the table name, storage engine, character set, and description.
 - You can click **Advanced** to edit table parameters.
8. Click **Save**. DMS generates the SQL statements used to create a table.

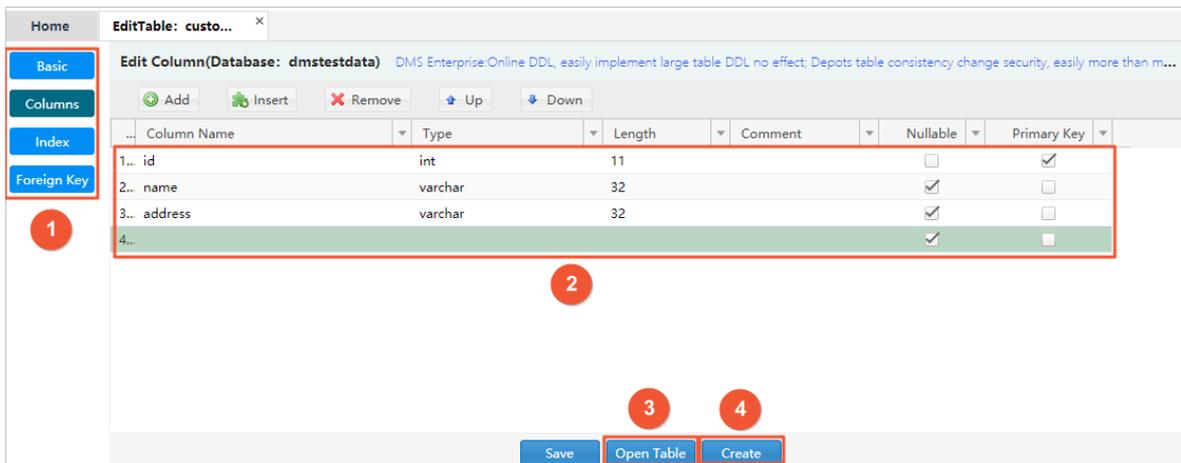
In the Submit Change dialog box, click **OK** after you confirm the SQL statements. DMS then adds the table to your database.

20.4.2.2. Edit a table

This topic describes how to edit a table in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side Table directory tree, right-click a table and choose **Edit Table Structure** from the shortcut menu to edit the table structure.
3. The Edit Table window is similar to the Create Table window. DMS automatically loads the table structure into the window.



- 1: Select a table object type, such as Columns or Index.
 - 2: Click a specific operation on the table object, which is similar to the Create and Edit operations on tables.
 - 3: Click **Open Table** to view and modify table data.
 - 4: Click **Create** to view the statements used to create a table.
4. Click **Save**. DMS displays the SQL statements used to modify the table structure.
Click **OK** after you confirm the SQL statements. DMS then saves the modified table structure to your database.

20.4.2.3. Delete a table

This topic describes how to delete a table in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side Table directory tree, right-click the table you want to delete and choose **Delete Table** from the shortcut menu.

Warning Deleting tables is a high-risk operation. Therefore, exercise caution when deleting tables.

3. Click Yes to delete the table.

20.4.2.4. Duplicate a table

This topic describes how to duplicate a table in DMS.

Procedure

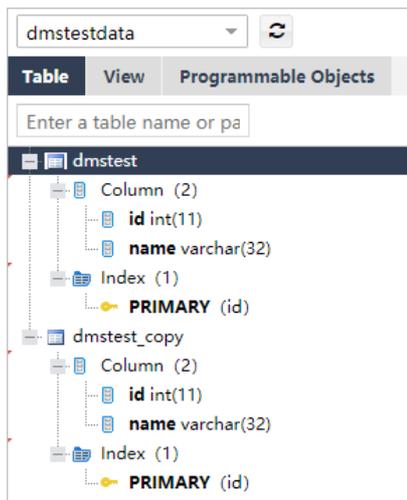
1. Log on to an RDS instance through DMS.
2. In the left-side navigation pane, click Table, right-click the table that you want to duplicate, and then choose To Duplicate Table from the shortcut menu.

The dialog box to enter the name of the destination table appears.

3. Enter the name of the destination table and then click OK. DMS duplicates the selected table.
4. The schema of the created destination table is the same as that of the selected source table.

A new table is created, as shown in Schema of the destination table.

Schema of the destination table



20.4.2.5. Generate SQL statement templates

This topic describes how to generate SQL templates in DMS.

Procedure

1. Log on to an RDS instance through DMS.
2. In the left-side Table directory tree, right-click the table you want to copy and choose Create SQL Template from the shortcut menu.
3. DMS generates SQL INSERT, UPDATE, SELECT and CREATE TABLE statement templates as a reference when you perform SQL operations.

20.4.2.6. Query table information

This topic describes how to query table information in DMS.

Procedure

1. Log on to an RDS instance through DMS.

2. In the left-side Table directory tree, right-click the table you want to query and choose **Object Info** from the shortcut menu.
3. DMS obtains information about the table object. Click the **Basic Info** tab to view basic information of the table.
4. Click the **Create Statement** tab to view the table creation statements.

20.4.2.7. Clear data

This topic describes how to clear table data in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the left-side Table directory tree, right-click the table that you want to clear data from and choose **Clear Table** from the shortcut menu.

 **Notice** Clearing table data is a high-risk operation and may affect your data usage. DMS prompts you to confirm whether to clear table data.

3. Click **Yes** if you want to clear table data. DMS then clears data of the selected table.
4. Open the table to check whether its data is cleared.

20.4.2.8. Perform operations on multiple tables

This topic describes how to perform operations on multiple tables in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. Delete multiple tables at a time.
 - i. In the left-side navigation pane, click **Table**, right-click a table name, and then choose **Batch Operations > Drop Batch Tables**.

The dialog box for you to select multiple tables to delete appears.
 - ii. Select the tables to be deleted.
 - iii. Click **OK**. DMS prompts you to confirm whether you want to perform the batch operation.
 - iv. Click **Drop**.

DMS deletes the selected tables at a time.
3. Perform other operations on multiple tables. You can clear data, delete or maintain tables, and modify table name prefixes in batches.
 - i. In the left-side navigation pane, click **Table**, right-click a table name, and then choose **Batch Operations > More**.
 - ii. Select multiple tables and then perform a batch operation. For example, click **Truncate Data**.
 - iii. In the **Information** dialog box that appears, click **Yes**.

DMS performs the batch operation.

20.4.2.9. Maintain a table

This topic describes how to maintain and optimize a table in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the left-side Table directory tree, right-click the table you want to maintain and choose **Maintain Table**

> **Optimize Table.**

3. Click **Yes**.

Click **Yes** if you want to optimize the table. Then DMS starts optimization.

Optimization allows you to reuse the table space in the database and organize file fragments.

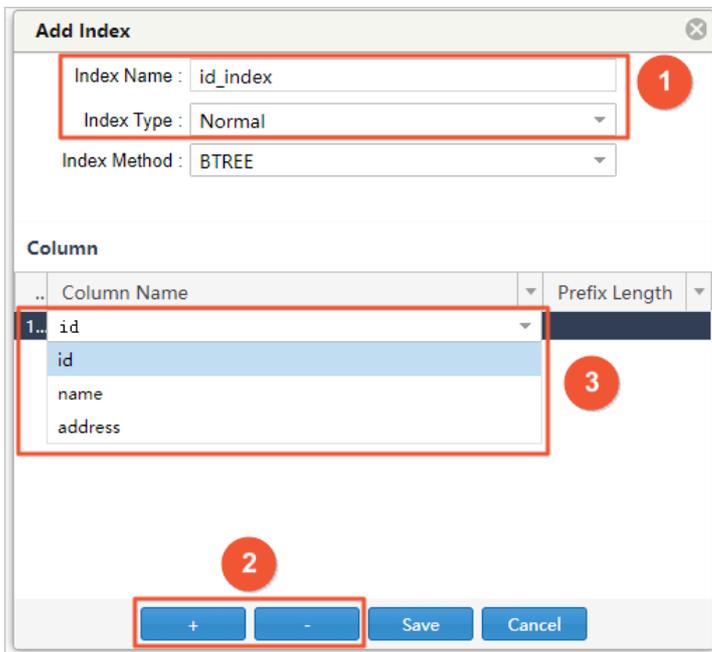
 **Note** You can check, restore, and analyze tables in a way similar to optimizing tables.

20.4.3. Manage indexes

This topic describes how to add, modify, or delete indexes in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side **Table** directory tree, expand the table you want to modify and choose **Index > Add Index**. The **Add Index** page appears.
3. Set index parameters.



- 1: Enter an index name and select an index type.
 - 2: Click **+** or **-** to add or delete a field to or from the index.
 - 3: Edit the fields of the index. You can enter or select values from the drop-down list. You can set a prefix length for a variable-length field (such as varchar) to save space occupied by the index.
4. Click **Save**.
DMS generates SQL statements used to add the index. Confirm the change.
 5. Click **Run**.
 6. After the index is added, check the indexes of the table to verify that the new index takes effect.
You can modify or delete the new index as needed.
 - In the left-side **Table** directory tree, right-click an index and choose **Modify Index** from the shortcut menu. The **Modify Index** window appears.
The method of modifying an index is similar to that of adding one, except that the SQL statements delete the old index before adding a new one.

- In the left-side Table directory tree, right-click an index and choose **Delete Index** from the shortcut menu. The **Delete Index** window appears. Click OK to delete the index.

20.4.4. Manage foreign keys

This topic describes how to add foreign keys in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the left-side Table directory tree, right-click the table to be modified and choose **Edit Table Structure** from the shortcut menu.
3. On the **Edit Table** page that appears, click the **Foreign Key** tab to edit foreign keys.
4. Enter the foreign key information, and set the fields of foreign keys and referenced tables.
5. Click **Save**.

20.4.5. Create a partition

This topic describes how to create partitions in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the left-side navigation page, click **Table**, right-click a table name, and then choose **Create Table** from the shortcut menu.
3. Enter the basic table information and set the table fields and partition information.
4. Click **Save** to save the created table schema.
A dialog box appears to prompt you to confirm the SQL statements used to create the table.
5. Click **OK**. DMS creates the partitioned table based on the configured partition fields and partitioning logic.
6. After you execute the SQL statements, check whether the partitioned table has been created.

20.4.6. Create a stored procedure

This topic describes how to create and manage stored procedures in DMS.

Context

A MySQL database is used as an example.

Stored procedures, functions, triggers, and events are considered programmable objects in DMS.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. Click the left-side **Programmable Object** directory tree, and choose **Stored Procedure > Create (Stored Procedure)**. The **Create Stored Procedure** tab is displayed.
3. Enter a name and a description for the stored procedure.
4. Click **OK**.
5. DMS provides a template for creating stored procedures. You only need to edit the stored procedure part.
6. Click **Save** to save the stored procedure to the database.
If a syntax error is found, DMS returns the cause of the error.
7. Click **Run** to run the stored procedure.
DMS displays a page for you to set the input parameters for the stored procedure.

Set the input parameters. In this example, set cnt to 80 to search for records that meet the Value=80 condition.

8. Click **Execute** to execute the stored procedure.

DMS displays output parameters or intermediate result set of the stored procedure, if any.

- The **Message** tab displays messages about the execution, such as output variables and intermediate result sets.
- The **Intermediate Result Set 1** tab displays the result set generated during the execution. If multiple intermediate result sets are available, DMS will generate multiple tabs, such as Intermediate Result Set 1, Intermediate Result Set 2, and Intermediate Result Set 3.

9. Click the **Intermediate Result Set 1** tab. DMS displays records with the value of 80.

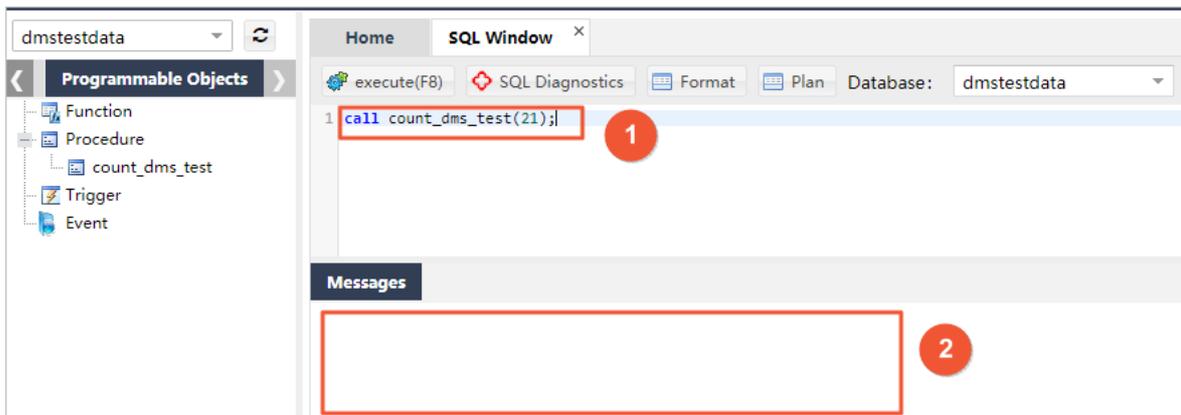
10. You can set the options when creating the stored procedure. Click **Option Settings** to set options for the stored procedure.

11. After a stored procedure is created, it is added to the Programmable Object directory tree.

You can perform other operations related to the stored procedure through the following menu options:

- **Create**
- **Edit**
- **Delete**
- **Execute**

12. You can run the stored procedure in the SQL window.



- 1: Run the call stored_procedure_name command to call a stored procedure.
- 2: The SQL window shows the result set of the stored procedure, if any.

20.4.7. Create a function

This topic describes how to create a function in DMS.

Context

Functions, stored procedures, triggers, and events are considered programmable objects in DMS.

Procedure

1. **Log on to an RDS instance through DMS.**
2. In the left-side **Programmable Object** directory tree, choose **Function > Create (Function)**.

The **Create Function** page is displayed.

3. Set basic information of the new function.
4. Click **OK**.

The **Edit Function** page appears. DMS generates a function creation template.

5. Enter information in the function part.
6. Click **Save**. DMS then checks whether the function is correctly defined. If not, DMS returns an error message.
DMS runs the correct function definition in your database, and returns a message, indicating that the function is saved.
7. Click **Execute** to execute the function.
8. Enter a parameter such as `wednesday` and click **Execute** to execute the function.
9. Click **Option Settings** to set different options for the function.
You can also run the function in the SQL window.

20.4.8. Create a view

This topic describes how to create and manage custom views in DMS.

Procedure

Create a view

1. **Log on to an RDS instance through DMS.**
2. Click the **View** directory tree on the left side to check the views of the current database.
3. Right-click the blank space and choose **New View** from the shortcut menu. The **Create: View** page is displayed.
4. Set basic information of the view.

The following example shows how to filter records in the `dmstest` table whose values are even numbers, and output the `id` and `name` fields.

5. Click **Save Changes**. DMS generates SQL statements used to create the view based on your settings.
6. Click **OK** after you confirm the SQL statements. DMS saves the defined view to your database.
7. The saved view is added to the **View** directory tree on the left side. You can click the view to display its definition.

Check the view

8. Right-click **View** and choose **Check View** from the shortcut menu to query data through the newly created view.
9. You can perform view-related operations in DMS.

The menu options include:

- View Data
- Create View
- Edit View
- Delete View
- Refresh Views

20.4.9. Create a trigger

This topic describes how to create and manage a trigger in DMS.

Context

Triggers, functions, stored procedures, and events are considered programmable objects in DMS.

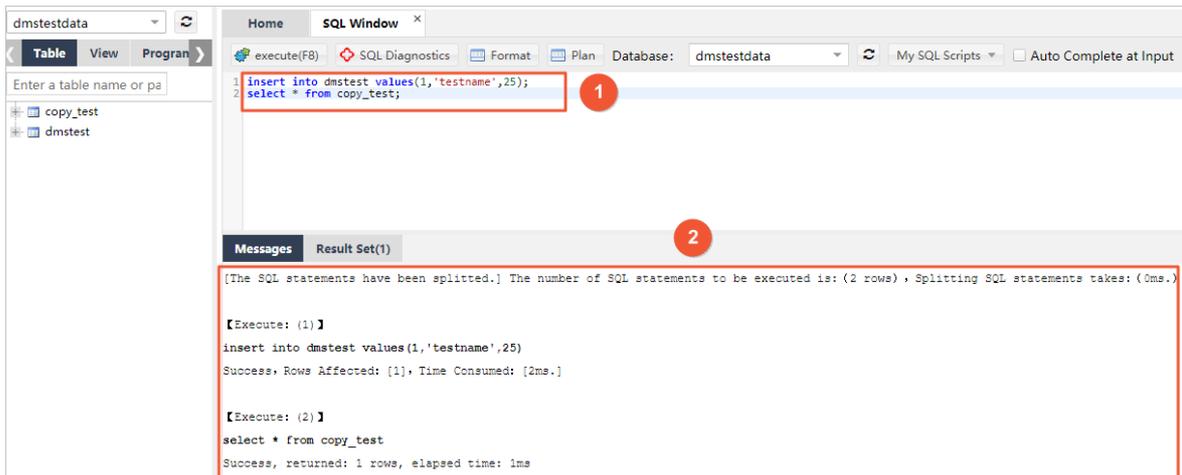
Procedure

1. **Log on to an RDS instance through DMS.**
2. Click the **Programmable Object** directory tree on the left side, and choose **Trigger > Create (Trigger)**.

The Create: Trigger tab is displayed.



- 1: Trigger table.
 - Enter a name for the trigger.
 - Select dmstest from the drop-down list as the trigger table.
 - Select AFTER from the drop-down list as the trigger time.
 - Select INSERT from the drop-down list as the trigger event.
- 2: Trigger settings.
 - Set the operations to be performed when the trigger event occurs.
 - When data is inserted into the dmstest table, the trigger in this example inserts data into the copy_test table and records the insertion time in copy_test.time.
- 3. Click Save after you finish the trigger settings. DMS then generates the SQL statement to be executed by the trigger based on your settings. Confirm the SQL statement.
- 4. Click OK. DMS then saves the trigger to your database. DMS returns a message, indicating that the trigger has been saved. In the left-side navigation pane, choose Programmable Object > Trigger to view the trigger you created.
- 5. You can insert data into the dmstest table to check whether the data is recorded in the copy_test table.



- 1: Insert data into the dmstest table and query the copy_test table for the inserted data.
- 2: The SQL window displays messages about the execution of the SQL statements. The messages indicate that a row is inserted into the dmstest table and that this row is also added to the copy_test table.
- 6. Check the result set in the SQL window to verify whether the insert operation is correctly performed by the trigger.
- 7. In the left-side navigation pane, choose Programmable Object > Trigger to perform trigger-related operations through the following menu options:

- Create (Trigger)
- Edit (Trigger)
- Delete (Trigger)

20.4.10. Create an event

This topic describes how to create and manage events in DMS.

Prerequisites

After you log on to a database, make sure that event support has been enabled for the database.

- Execute the `SELECT @@event_scheduler;` statement to check whether the database supports events. If ON is returned, event support is enabled.
- If OFF is returned, event support is disabled. You need to enable event support by modifying the configuration file or executing the `SET GLOBAL event_scheduler = ON;` statement.

Context

Events, triggers, functions, and stored procedures are considered programmable objects in DMS.

Procedure

1. Log on to an RDS instance through DMS.
2. Click the Programmable Object directory tree on the left side, and choose Event > Create (Event).

The Create Event page is displayed.

1: In the event setup area, set the event name, cycle, start time, end time, status, and comment, and choose whether to enable cyclic execution.

2: In the event execution Statement area, set the operations to be performed when a scheduled event is triggered.

3. Set an event trigger rule and the SQL statements for event execution.
4. Click Save. DMS generates the SQL statements used to create the event.
5. After you confirm that the SQL statements are correct, click OK. DMS then executes the edited event in your database.
 - If the event is created, DMS returns a message, indicating that the event is saved.
 - In the left-side navigation pane, choose Programmable Object > Event to view the event you created.
6. Check whether the event is properly executed in the SQL window.

In this example, the event executes SQL statements to insert a piece of data into the copy_testtable every minute. Check the copy_test table to see whether the data is inserted as programmed.

- 7. You can perform various event-related operations in DMS. The menu options include:
 - Create (Event)
 - Edit (Event)
 - Delete (Event)

20.5. Data processing

20.5.1. Import data

This topic describes how to use DMS to import data.

Context

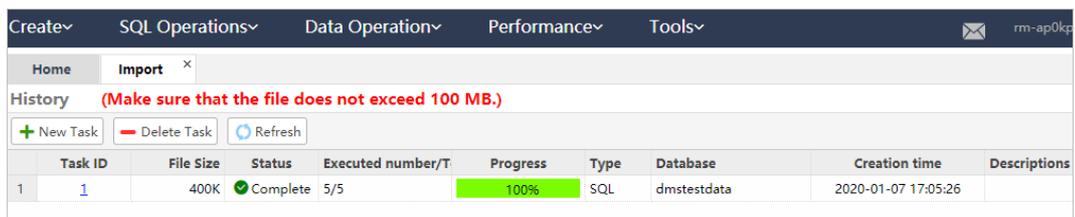
A MySQL database is used in this example.

Procedure

1. Log on to an RDS instance through DMS.
2. In the top navigation bar, choose Data Operation > Import.

The Import tab appears, as shown in [Import tab](#).

Import tab



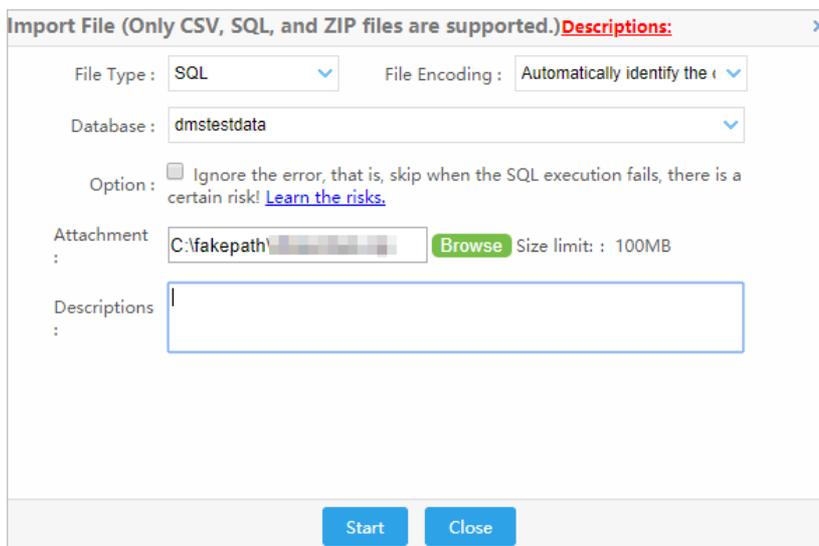
The Import tab contains the import toolbar and import history.

If you have imported data, you can view previous data import tasks in [History](#).

3. Click New Task.

The Import File dialog box appears, as shown in [Import File dialog box](#).

Import File dialog box



- Select the file type of the file to be imported. Currently, only SQL and CSV files are supported.

- If the data file uses a specific character set, you can manually specify the character set. By default, DMS detects the character sets used in a file.
 - If an error occurs while an SQL statement is being executed, DMS will terminate the import task. You can select Ignore Error to proceed. However, this operation may affect subsequent operations.
 - You can enter a brief description of the import task for later review.
4. Click **Start** to start the import task.
- If the imported data has any error, DMS terminates the import process and returns an error message. You can modify the data file to correct the error and import it again.
- If the imported data and SQL statements are correct, DMS displays the import progress, volume of imported data, and time elapsed.
- After the import is complete, you can view the import task in **History**.
- Click a task ID to view the execution details of the task.

20.5.2. Export data

20.5.2.1. Export a database

This topic describes how to use DMS to export a database.

Context

A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **Data Operation > Export**. The **Export** tab appears.
3. On the **Export** tab, choose **New > Export Database**.
4. In the **New** dialog box that appears, select a database, a file type (SQL or CSV), and the content to be exported (schema and data, only data, or only schema). Select tables on the right side, and select additional information in the **Additional Information** section.
5. Click **OK**.
6. In the **Information** dialog box that appears, click **Yes**. DMS then starts executing the export task.

DMS refreshes the export progress every two seconds.

You can close the **Export** tab, and view and download the exported file in the export task list later.

After the export is complete, DMS automatically downloads the exported file to your local computer. You can also click **Download** to download the exported file.

You can view previously submitted export tasks in the export task list. You can click a task ID to view the task details, and can download the corresponding exported file.

20.5.2.2. Export an SQL result set

This topic describes how to use DMS to export an SQL result set.

Context

A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **Data Operation > Export**. The **Export** tab appears.
3. On the **Export** tab, choose **New > Export SQL result sets**.

4. In the **New** dialog box that appears, complete the settings as needed.

Select a file type (CSV or SQL_Insert) and a database, set the maximum number of rows of the result set, and enter the SQL statements.

5. Click **OK**. DMS then executes the task of exporting the SQL result set in the background.

After the export task is complete, DMS automatically downloads the exported files to your local computer. You can also click **Download** to download the exported files.

DMS also summarizes the export results and automatically downloads the files of exported SQL result sets.

You can view the export tasks of SQL result sets that you submitted in the export task list and download SQL result set files.

20.6. Performance

20.6.1. Lock wait

20.6.1.1. View sessions in the lock wait state

When a session of an ApsaraDB RDS for MySQL instance is waiting for a row exclusive lock held by another session, an InnoDB lock wait will occur. This topic describes how to use DMS to view sessions in the lock wait state.

Context

A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **Performance > InnoDB Lock Waits**.
If transactions of the current instance are waiting for locks, the Lock and Lock-Wait icons are displayed.
3. Move the pointer over the Lock or Lock-Wait icon to view the sessions that hold a lock or sessions that are waiting for a lock, and related session IDs.
4. Click  to reload the data.

20.6.1.2. Terminate a session in the lock wait state

This topic describes how to use DMS to terminate a session in the lock wait state.

Context

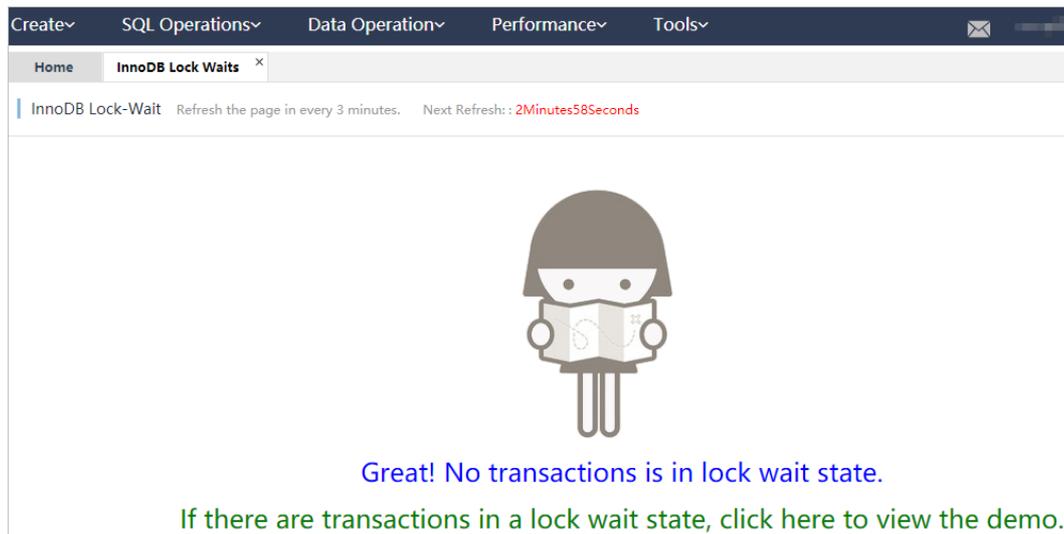
A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **Performance > InnoDB Lock Waits**. If transactions of the current instance are waiting for locks, the Lock and Lock-Wait icons are displayed.
3. Move the pointer over the Lock or Lock-Wait icon to view the sessions that hold a lock or sessions that are waiting for a lock, and related session IDs.
4. Click the Lock or Lock-Wait icon.
The **Delete Session** message appears.
5. Click **Yes** to terminate the current session.

InnoDB Lock Waits tab shows the page when no transactions are waiting for locks.

InnoDB Lock Waits tab



20.7. Extended tools

20.7.1. View statistics on table data volumes

This topic describes how to view statistics on table data volumes in DMS.

Context

A MySQL database is used in this example.

Procedure

1. [Log on to an RDS instance through DMS.](#)
2. In the top navigation bar, choose **Tools > Table Data Amount**. The **Table Data Amount** tab appears, as shown in [Collection of statistics on table data volumes](#).

Collection of statistics on table data volumes

Database	Table Name	Storage	Rows	Row Leng.	Data	Index	All	Creation time	Collation Rules
d	customer	InnoDB	10137	156	1.52MB	0B	1.52MB	2020-01-07 17:05:27	utf8_general_ci
d	order	InnoDB	0	0	16KB	0B	16KB	2020-01-07 16:54:42	utf8_general_ci

3. This tab displays the information about all user tables of the current instance, including the database name, table name, storage engine, number of rows, row size (in bytes), data, index, total table size, creation time, and collation rules.

You can filter statistics on table data volumes based on a range of criteria such as database name, table name, total table size (in MB), number of table rows, global sorting, and storage engine. You can also perform the paging, refresh, and reset operations.

20.7.2. View ER diagrams

This topic describes how to view ER diagrams in DMS.

Context

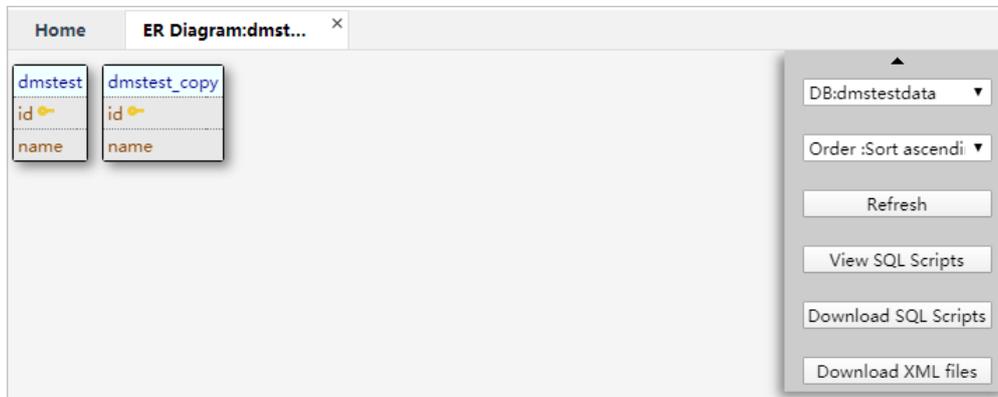
This topic uses a MySQL database as an example.

Procedure

1. Log on to an RDS instance through DMS.
2. In the top navigation bar, choose **Tools** > **ER Diagram**.

The ER Diagram tab appears, as shown in [ER Diagram tab](#).

ER Diagram tab



The ER Diagram tab displays the relationship between tables in the current database and provides methods for representing table names, column names, indexes, and relationships.

3. You can perform the following operations:
 - Select a database from the **DB:<Database name>** drop-down list to switch to the database.
 - Select a value from the **Order:<Sorting option>** drop-down list. Tables can be sorted by table name or column count in ascending or descending order.
 - Click **Refresh** to refresh the current ER Diagram tab.
 - Click **View SQL Scripts** to view the SQL statements used to create tables in the current database.
 - Click **Download SQL Scripts** to download the SQL statements used to create tables in the current database.
 - Click **Download XML files** to download the SQL statements used to create tables in the current database in XML format.
 - Double-click the name of a table column to view the column definition.
 - Double-click a table name to edit the table on a new tab.

21. Server Load Balancer (SLB)

21.1. What is SLB?

This topic provides an overview of Server Load Balancer (SLB). SLB distributes inbound network traffic across multiple Elastic Compute Service (ECS) instances that act as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

Overview

After you add ECS instances that reside in the same region to an SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize these ECS instances into backend servers in a high-performance server pool that ensures high availability. Client requests are distributed to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically removes unhealthy ones from the server pool to eliminate single points of failure (SPOFs). This enhances the resilience of your applications.

Components

SLB consists of three components:

- SLB instances

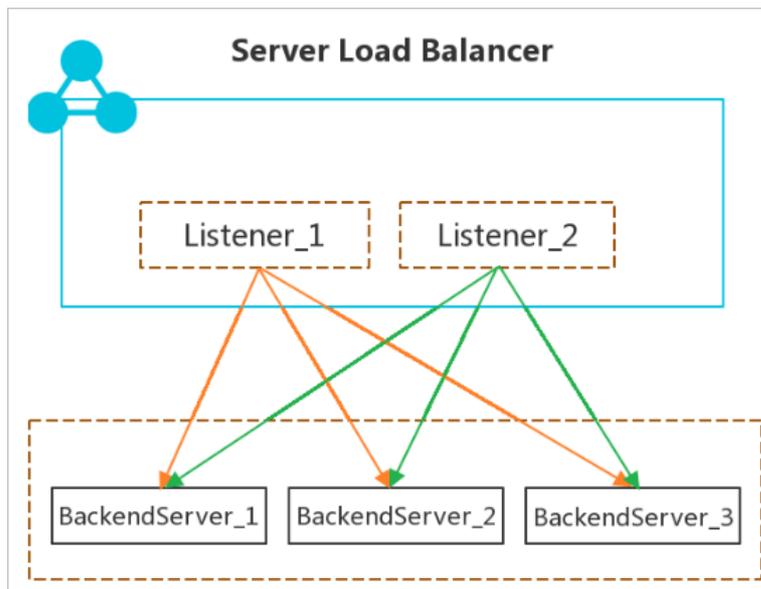
An SLB instance is a key load-balancing component in SLB. It receives traffic and distributes traffic to backend servers. To get started with SLB, you must create an SLB instance and add at least one listener and two ECS instances to the SLB instance.

- Listeners

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

- Backend servers

ECS instances are used as backend servers in SLB to receive and process distributed requests. ECS instances can be added to the default server group of an SLB instance. You can also add multiple ECS servers to VServer groups or primary/secondary server groups after the corresponding groups are created.



Benefits

- High availability

SLB features full redundancy that avoids SPOFs and supports zone-disaster recovery. You can use SLB with Apsara Stack DNS to achieve geo-disaster recovery with an availability of up to 99.95%.

SLB can be scaled based on network traffic to protect your services from outages caused by fluctuating traffic flows.

- **Strong scalability**

You can increase or decrease the number of backend servers to adjust the load balancing capacity for your applications.

- **Low costs**

SLB can save 60% of load balancing costs compared with using traditional hardware solutions.

- **Outstanding security**

You can use SLB with Apsara Stack Security to defend your applications against 5 Gbit/s distributed denial of service (DDoS) attacks.

- **High concurrency**

An SLB cluster supports hundreds of millions of concurrent connections, and a single SLB instance supports tens of millions of concurrent connections.

21.2. Log on to the SLB console

This topic provides an example of how to log on to the Server Load Balancer (SLB) console by using Google Chrome.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > Server Load Balancer**.

21.3. Quick start

21.3.1. Overview

This quick start tutorial describes how to create a public-facing Server Load Balancer (SLB) instance and how to forward requests to two backend servers.

 **Note** Before creating an SLB instance, you must determine the region, type, and billing method of the SLB instance. For more information, see [Before you begin](#).

This tutorial includes the following content:

1. **Create an SLB instance**

Create an SLB instance. An SLB instance is a running entity of the SLB service.

2. **Add listeners and backend servers.**

Configure listening rules and backend servers for the SLB instance.

3. **Release an SLB instance**

If you no longer need the SLB instance, delete it to avoid extra fees.

21.3.2. Before you begin

This article presents the essential considerations for configuring an SLB instance. Before you create an SLB instance, you must determine the types of listeners and the network traffic you want to balance.

Instance region

When you select a region, note the following points:

- To reduce latency and increase the download speed, we recommend that you select a region closest to your end-users.
- SLB offers stable and reliable load balancing services by providing support for primary/secondary failovers in most regions. This implements disaster recovery across different zones within the same region. We recommend that you select a region that supports the primary/secondary SLB deployment.
- SLB instances cannot span across regions. Therefore, you must make sure that the SLB instance and its backend Elastic Compute Service (ECS) instances are located in the same region.

Network traffic

SLB provides load balancing services for both Internet and internal network traffic:

- If you need to use SLB to distribute requests from the Internet, you can create an Internet-facing SLB instance.

An Internet-facing SLB instance comes with a public IP address to receive requests from the Internet.

- If you need to use SLB to distribute requests from the internal network, you can create an internal SLB instance.

Internal SLB instances only have private IP addresses and are only accessible from the internal network and not from the Internet.

Listener protocol

SLB supports Layer-4 load balancing of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic, and Layer-7 load balancing of HTTP and HTTPS traffic.

- A Layer-4 listener directly distributes requests to backend servers without modifying packet headers. After a client request reaches a Layer-4 listener, SLB uses the backend port configured for the listener to establish a TCP connection with an Elastic Compute Service (ECS) instance (backend server).
- A Layer-7 listener is implemented as a reverse proxy. After a client request reaches a Layer-7 listener, SLB establishes a new TCP connection over HTTP to a backend server, instead of directly forwarding the request to the backend server (ECS instance).

Compared with Layer-4 listeners, Layer-7 listeners require an additional step of engine processing. Therefore, Layer-4 listeners provide better performance than Layer-7 listeners. In addition, the performance of Layer-7 listeners can also be affected by factors such as insufficient client ports or excessive backend server connections. Therefore, we recommend that you use Layer-4 listeners for high-performance load-balancing services.

Backend servers

Before you use the SLB service, you must create ECS instances, deploy applications on them, and add the ECS instances to your SLB instance to process client requests.

When you create and configure an ECS instance, note the following points:

- Select a region and zone for the ECS instance

Make sure that the ECS instance resides in the same region and Virtual Private Cloud (VPC) as the SLB instance. We recommend that you deploy ECS instances in different zones to improve availability. For more information about how to create an ECS instance, see [Create an instance by using the provided wizard](#).

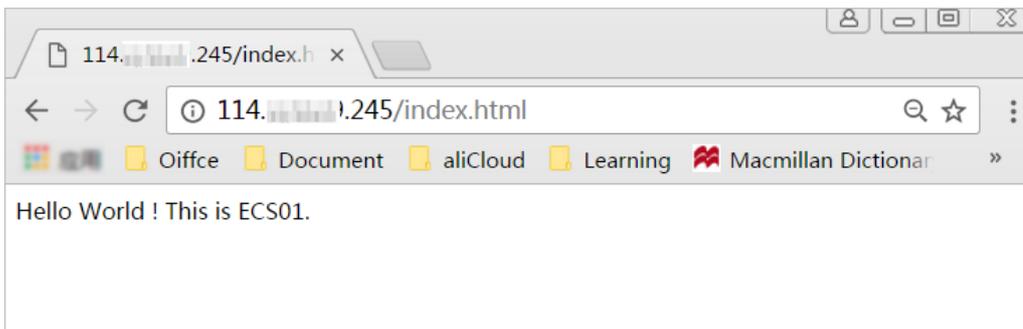
In this example, two ECS instances named ECS01 and ECS02 are created in the China (Hangzhou) region. The following figure shows their basic configurations.

Instance Name/ID	IP Address	Status	Monitoring	Health Check	Port/Health Check/Backend Server	Actions
ECS01	[Public IPv4 Address]	Active			HTTP:80 - VServer Group doctest	Configure Listener Add Backend Server
ECS02	[Public IPv4 Address]	Active			HTTP:80 (R) 443 - VServer Group test1	Configure Listener Add Backend Server

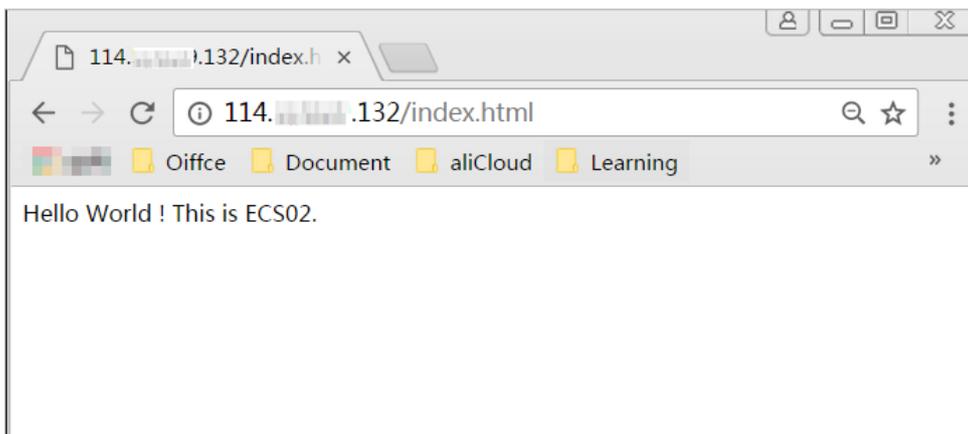
- Configure applications

In this example, two static web pages are built on ECS01 and ECS02 by using Apache.

- Enter the Elastic IP address (EIP) associated with ECS01 in the address box of your browser.



- Enter the EIP associated with ECS02 in the address box.



No additional configuration is required after you deploy applications on the ECS instances. However, if you need to use a Layer-4 (TCP or UDP) listener and the ECS instances run on Linux, make sure that the following parameters in the `net.ipv4.conf` file under `/etc/sysctl.conf` are set to 0:

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.eth0.rp_filter = 0
```

21.3.3. Create an SLB instance

This topic describes how to create an SLB instance. To get started with SLB, you must create an SLB instance. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- You have created ECS instances and deployed applications on them.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules of the ECS instances permit traffic on HTTP port 80 and HTTPS port 443.

Procedure

1. [Log on to the SLB console](#).
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. On the **Server Load Balancers** page, click **Create Instance**.
 - **Organization:** Select the organization to which the SLB instance belongs.

 **Note** Make sure that the organization of the SLB instance is the same as the organization of its backend servers.

- **Resource Set:** Select the resource set to which the SLB instance belongs.
- **Region:** Select a region for the SLB instance.
- **Zone:** Select a zone for the SLB instance from the drop-down list.
- **Name:** Enter a name for the SLB instance.

The name must be 2 to 128 characters in length and can contain letters, digits, full-width characters, underscores (_), hyphens (-), periods (.), and colons (:). Line breaks and spaces are also supported. The name must start with a letter and cannot start with `http://` or `https://`.

- **Type:** Select the instance type of the SLB instance. Possible options are **Shared-Performance** and **Guaranteed-Performance**. Shared-performance instances share resources with each other, which means their performance cannot be guaranteed. The performance of a guaranteed-performance SLB instance varies by type.
- **Network Access:** Select the type of network traffic to balance. Possible options are **Internal Network** and **Public Network**. In this example, select **Internal Network**.
- **Network Type:** Select the network type of the SLB instance. Possible options are **Classic Network** and **VPC**. In this example, select **VPC**.
- **IP Version:** Select an IP version.
- **Service IP:** Enter the service IP address of the SLB instance. Make sure that the service IP address is valid. Otherwise, the SLB instance cannot be created. If you do not specify the service IP address, the system automatically allocates an IP address to the SLB instance.

4. Click **Submit**.

What's next

[Configure an SLB instance](#)

21.3.4. Configure an SLB instance

This topic describes how to configure an SLB instance. After you create an SLB instance, you must add at least one listener and a group of backend servers to this SLB instance so that it can forward traffic. The following example sets up a TCP listener and adds two ECS instances (ECS01 and ECS02) that host static web pages as backend servers to an SLB instance.

Procedure

1. **Log on to the SLB console**
2. On the **Server Load Balancers** page, find the target SLB instance and click **Configure Listener** in the **Actions** column.
3. In the **Protocol and Listener** step, complete the following information and use the default values for other fields to configure the listening rule.
 - **Select Listener Protocol:** Select a listener protocol. In this example, select **TCP**.
 - **Listening Port:** Set a frontend port to receive and forward requests to backend servers.

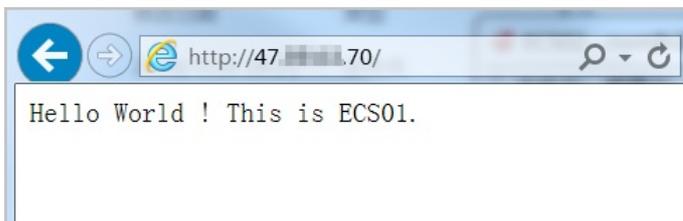
This port is configured for external load balancing. Generally, port 80 is used for HTTP and port 443 is used for HTTPS.

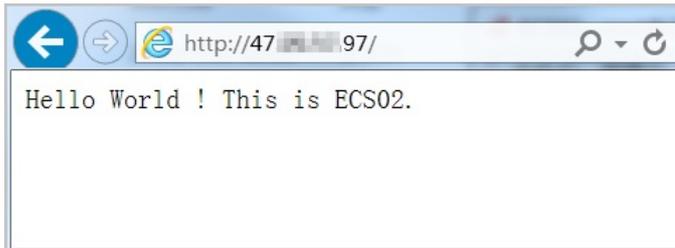
In this example, set the port number to **80**.
 - **Enable Peak Bandwidth Limit:** You can switch on this option and then set a bandwidth limit to control the bandwidth that is used by the applications running on backend servers to provide external services.
 - **Scheduling Algorithm:** Select a scheduling algorithm for distributing requests to backend servers. SLB supports the following scheduling algorithms. In this example, select **Round-Robin (RR)**.
 - **Weighted Round-Robin (WRR):** Requests are distributed proportionally based on the assigned weights of backend servers. Backend servers with higher weights receive more requests.
 - **Weighted Least Connections (WLC):** Requests are distributed based on the combination of the weights and active connections of backend servers. If multiple backend servers have the same weight, requests are routed to the backend server with the lowest number of active connections.
 - **Round-Robin (RR):** Requests are evenly and sequentially distributed to backend servers.
4. Click **Next**. In the **Backend Servers** step, select **Default Server Group** and click **Add More** to add backend servers.
 - i. In the **Select Servers** dialog box, select the previously created ECS01 and ECS02 instances and click **Next**.
 - ii. Configure the weights of the ECS instances. The higher the weight, the more requests a backend server receives. The default weight is 100. We recommend that you use the default value.
 - iii. Click **Add**.
 - iv. In the **Default Server Group** section, specify backend server ports. A backend server uses this port to receive requests. You can specify the same port for multiple backend servers of an SLB instance. In this example, set the port number to 80.
5. Click **Next** to configure the health check. The default health check settings are used in this example.

After you enable the health check feature, when a backend server is detected unhealthy, SLB bypasses requests from this backend server to other healthy backend servers. SLB will only send requests to this backend server when it has been restored and is considered healthy.
6. Click **Next**. In the **Submit** step, check the configuration and click **Submit**.
7. Click **OK** to go back to the **Server Load Balancers** page, and click  to refresh the page.

If the health check state of a backend ECS instance is **Active**, the backend server is working properly and is able to process requests.

8. In the web browser, enter the service IP address of the SLB instance in the address box to test network load balancing.





21.3.5. Release an SLB instance

This topic describes how to release an SLB instance. You can release an SLB instance when you no longer need it to avoid accumulating unnecessary charges. Releasing an SLB instance will not affect the operation of the backend ECS instances.

Procedure

1. [Log on to the SLB console](#)
2. Find and select the target SLB instance and click **Release** at the bottom of the list. You can also choose 
 - > **Release** in the Actions column.
3. In the Release dialog box, select **Release Now**.

 **Note** The system performs release operations at 30-minute and hour marks. However, billing for the SLB instance is stopped at the specified release time.

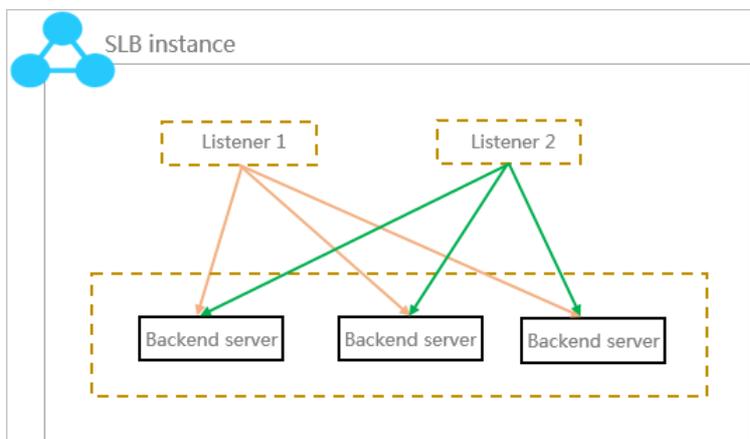
4. Click **Next**.
5. Click **OK**.

 **Note** Pay-as-you-go SLB instances cannot be restored once deleted. We recommend that you exercise caution when you release SLB instances.

21.4. SLB instances

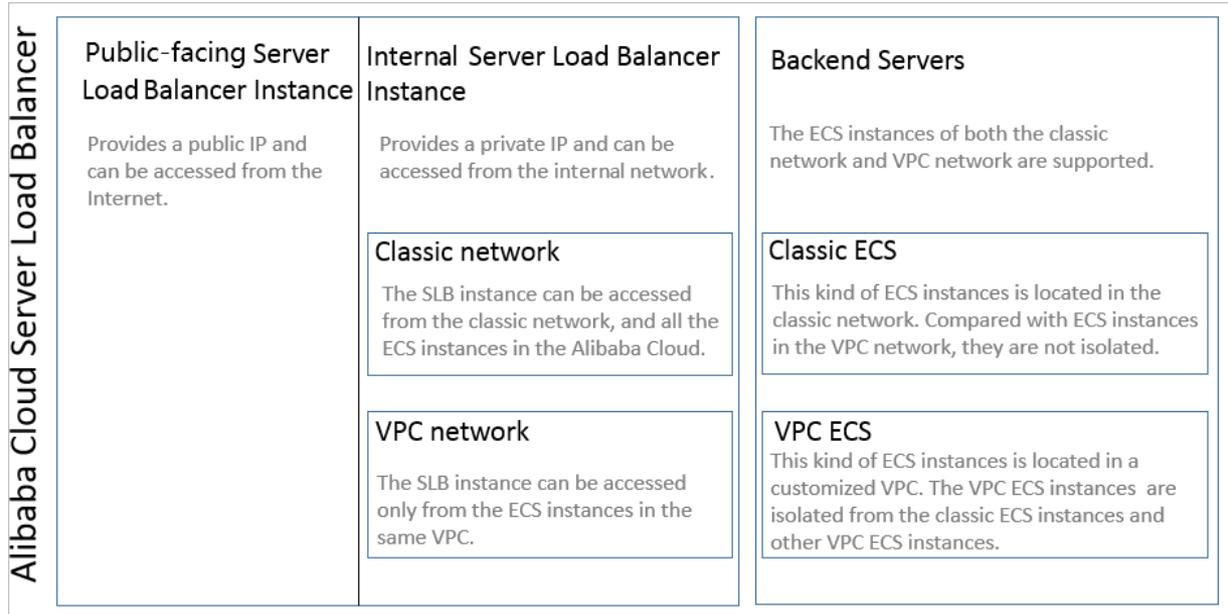
21.4.1. SLB instance overview

A Server Load Balancer (SLB) instance is a virtual machine in which the SLB service runs. To use the SLB service, you must create an SLB instance first, and then add listeners and backend servers to the SLB instance.



Instance network type

Alibaba Cloud provides public-facing and internal load balancing services. If you create a public-facing SLB instance, a public IP address is allocated to it. If you create an internal SLB instance, a private IP address is allocated.



- **Public-facing SLB instances**

A public-facing SLB instance distributes client requests over the Internet to backend servers according to configured forwarding rules.

When you create a public-facing SLB instance, the system allocates a public IP address to the instance. You can resolve a domain name to the public IP address to provide public services.

- **Internal SLB instances**

Internal SLB instances can only be used inside Alibaba Cloud and can only forward requests from clients that can access the internal network of SLB.

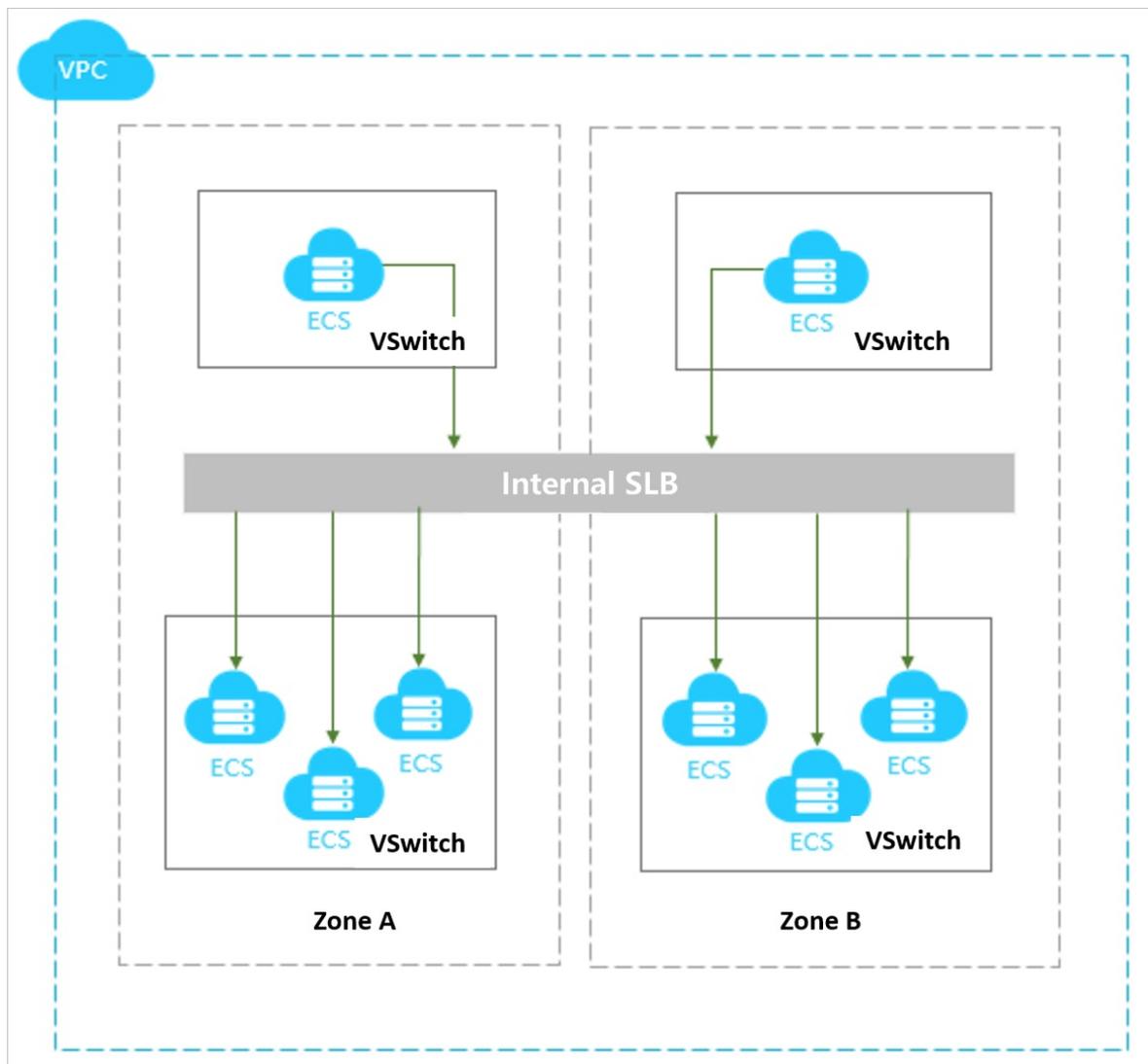
For an internal SLB instance, you can select the network type:

- **Classic network**

If you choose classic network for an internal SLB instance, the IP address of the SLB instance is allocated and maintained by Alibaba Cloud. The instance can only be accessed by classic-network ECS instances.

- VPC network

If you choose VPC network for an internal SLB instance, the IP address of the SLB instance is allocated from the CIDR block of the VSwitch that the instance belongs to. An SLB instance of the VPC-type network can only be accessed by ECS instances in the same VPC.



21.4.2. Create an SLB instance

This topic describes how to create an SLB instance. To get started with SLB, you must create an SLB instance. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- You have created ECS instances and deployed applications on them.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules of the ECS instances permit traffic on HTTP port 80 and HTTPS port 443.

Procedure

1. [Log on to the SLB console](#).
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. On the **Server Load Balancers** page, click **Create Instance**.
 - **Organization:** Select the organization to which the SLB instance belongs.

 **Note** Make sure that the organization of the SLB instance is the same as the organization of its backend servers.

- **Resource Set:** Select the resource set to which the SLB instance belongs.
- **Region:** Select a region for the SLB instance.
- **Zone:** Select a zone for the SLB instance from the drop-down list.
- **Name:** Enter a name for the SLB instance.

The name must be 2 to 128 characters in length and can contain letters, digits, hyphens (-), colons (:), commas (,), periods (.), and underscores (_). It must start with a letter and cannot start with `http://` or `https://`.

- **Type:** Select the instance type of the SLB instance. Possible options are Shared-Performance and Guaranteed-Performance. Shared-performance instances share resources with each other, which means their performance cannot be guaranteed. The performance of a guaranteed-performance SLB instance varies by type.
 - **Maximum concurrent connections:** the maximum number of concurrent connections that an SLB instance can support. Connection requests that exceed this limit will be discarded.
 - **Connections per second (CPS):** the number of new connections per second. Connection requests that exceed this limit will be discarded.
 - **QPS:** the number of HTTP or HTTPS queries or requests that can be processed per second. This metric is specific to Layer-7 listeners. Connection requests that exceed this limit will be discarded.
- **Network Access:** Select the type of network traffic to balance. Possible options are Internal Network and Public Network.

If you select Public Network, you can choose an IP version between IPv6 and IPv4 for the SLB instance and specify a bandwidth for billing.
- **Network Type:** Select the network type of the SLB instance. Possible options are Classic Network and VPC.
- **IP Version:** Select an IP version. Internal SLB instances support only IPv4. Internet-facing SLB instances support IPv6 and IPv4.
- **Service IP:** Enter the service IP address of the SLB instance. Make sure that the service IP address is valid. Otherwise, the SLB instance cannot be created. If you do not specify the service IP address, the system automatically allocates an IP address to the SLB instance.

4. Click **Submit**.

What's next

[Configure an SLB instance](#)

21.4.3. Start and stop an SLB instance

This topic describes how to start and stop an SLB instance. SLB instances can be started or stopped at any time. A stopped SLB instance does not receive or forward client traffic.

Procedure

1. [Log on to the SLB console](#)
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Find the target SLB instance. In the **Actions** column, choose  > **Start** or  > **Stop**.
4. To start or stop multiple instances at a time, select the instances and click **Start** or **Stop** at the bottom of the page.

21.4.4. Tags

21.4.4.1. Tag overview

This topic provides an overview of tags in SLB. SLB provides the tag management feature that allows you to classify SLB instances by using tags.

Each tag consists of a key and a value. Before you use tags, note the following limits:

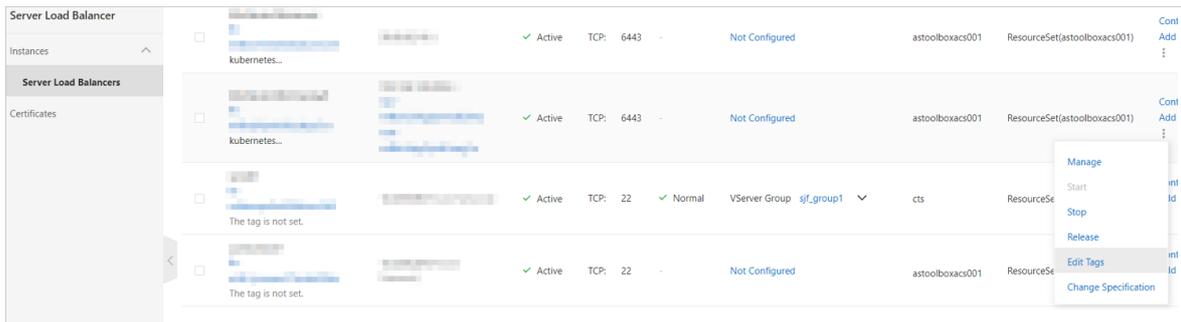
- Tags must be added to SLB instances.
- Each SLB instance can have a maximum of ten tags. You can add or remove a maximum of 5 tags at a time.
- The key of each tag added to an SLB instance must be unique. If a tag with the same key already exists, the tag is overwritten with the new value.

21.4.4.2. Add tags

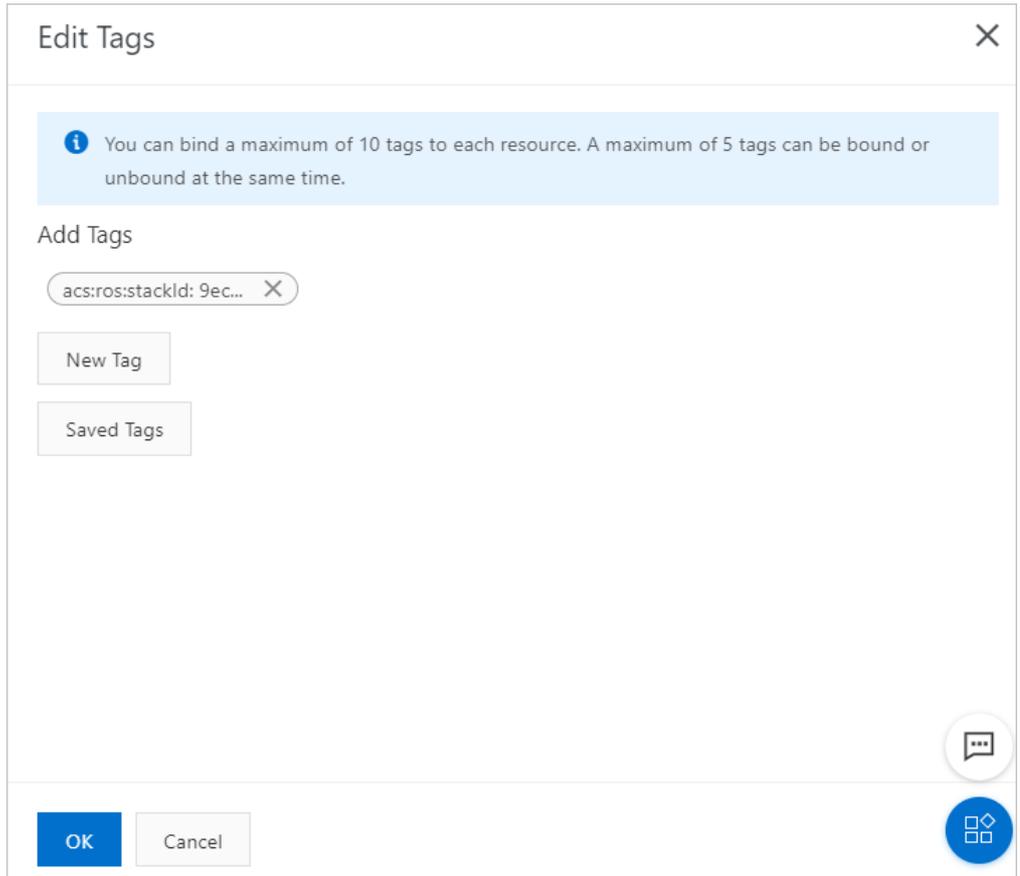
This topic describes how to add tags to an SLB instance.

Procedure

1. **Log on to the SLB console**
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. In the **Actions** column, choose  > **Edit Tags**.



4. Edit tags in the **Edit Tags** dialog box. To add a tag, perform the following operations:
 - To add an existing tag, click **Saved Tags** and then select a tag.
 - To create and add a new tag, click **New Tag** in the **Edit Tags** dialog box, enter the key and value of the new tag, and then click **OK** next to the value.



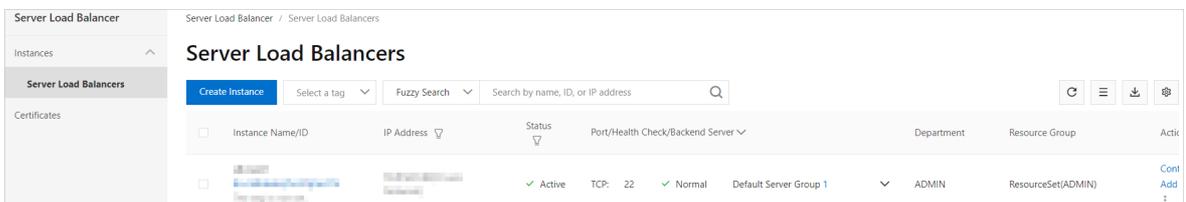
5. Click OK.

21.4.4.3. Query SLB instances by tag

This topic describes how to use tags to query SLB instances.

Procedure

1. [Log on to the SLB console](#).
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Select a data from the **Select a tag** drip-down list to filter instances.



Note To clear the search condition, move the pointer over the selected tag and click the displayed deletion icon next to it.

21.4.4.4. Remove tags

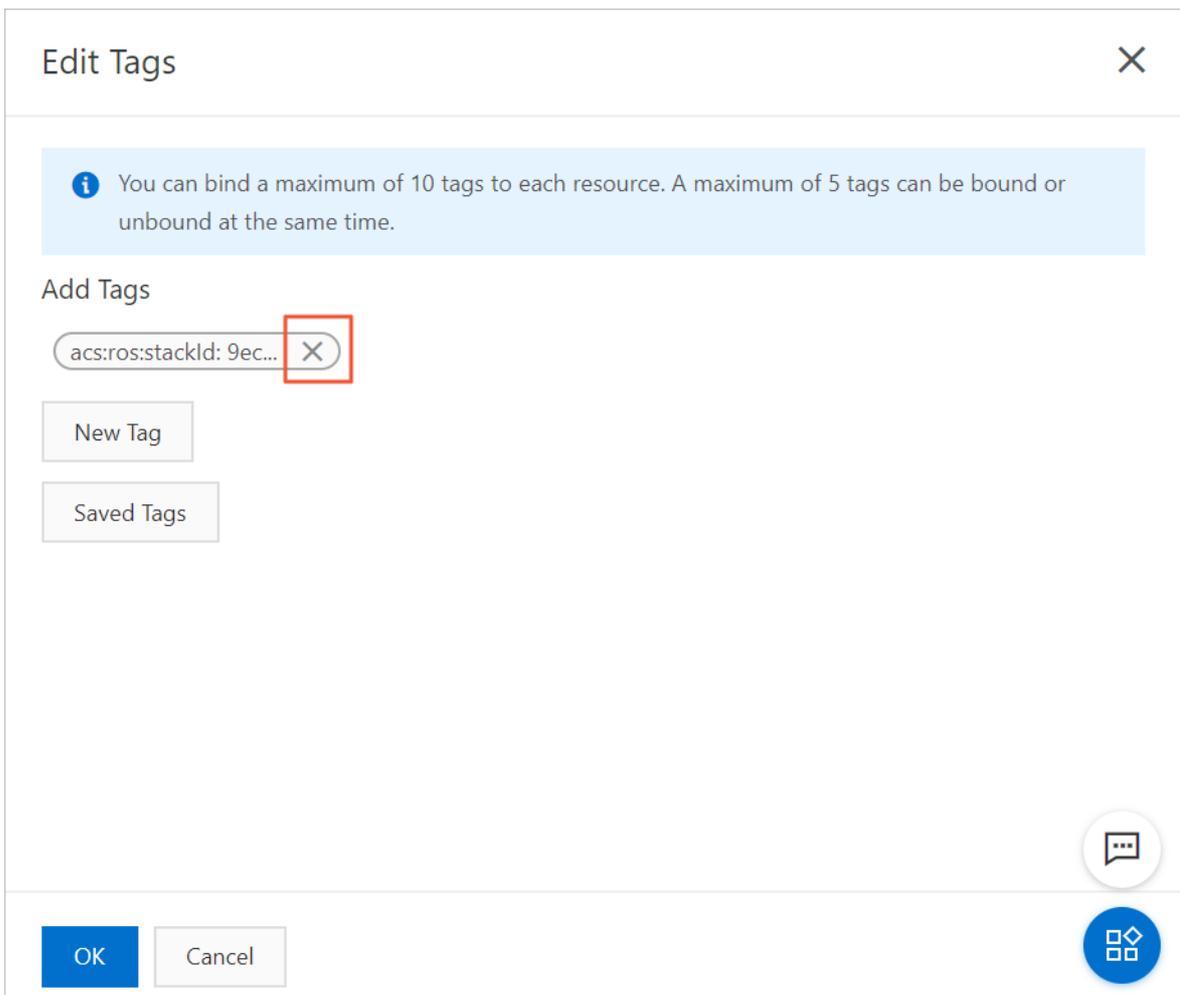
This topic describes how to remove tags from an SLB instance. You can only remove tags for one SLB instance at a time.

Procedure

1. [Log on to the SLB console.](#)
2. In the left-side navigation pane, choose **Instances > Server Load Balancers.**
3. In the **Actions** column, choose  > **Edit Tags.**

4. In the **Edit Tags** dialog box, click the deletion icon next to the tags to be removed, and then click **OK.**

 **Note** If a tag is removed from an SLB instance and is not added to any other instances, the tag is deleted from the system.



21.4.5. Release an SLB instance

This topic describes how to release an SLB instance. You can release SLB instances immediately.

Procedure

1. [Log on to the SLB console.](#)
2. In the left-side navigation pane, choose **Instances > Server Load Balancers.**
3. Find the target SLB instance and choose  > **Release.**

4. In the **Release** dialog box, select **Release Now.**
5. Click **Next.**

6. Confirm the displayed information and click **OK** to release the instance.

21.5. Listeners

21.5.1. Listener overview

This topic provides an overview of listeners. After you create an SLB instance, you must configure one or more listeners for it. A listener checks for connection requests and then distributes the requests to backend servers based on the forwarding rules that are defined by a specified scheduling algorithm.

SLB provides listeners for Layer-4 (TCP and UDP) and Layer-7 (HTTP and HTTPS) load balancing. The following table lists the features and use cases of these listeners.

Protocol	Feature	Use case
TCP	<ul style="list-style-type: none"> • A connection-oriented protocol that requires a logical connection to be established before any data can be transmitted • Source IP address-based session persistence • Source IP addresses readable at the network layer • Fast data transmission 	<ul style="list-style-type: none"> • Services that prioritize high reliability and data accuracy over high-speed data transmission, such as file transmission, email sending or receiving, and remote logon • Basic web applications <p>For more information, see Add a TCP listener.</p>
UDP	<ul style="list-style-type: none"> • A connectionless protocol that transmits data packets without implementing the three-way handshake and does not provide error recovery or data retransmission • Fast data transmission with relatively low reliability 	<p>Services that focus on real-time content delivery other than reliability, such as video chats and real-time quotes</p> <p>For more information, see Add a UDP listener.</p>
HTTP	<ul style="list-style-type: none"> • An application-layer protocol used to package data • Cookie-based session persistence • Support for using the X-Forwarded-For (XFF) header to identify original client IP addresses 	<p>Applications that need to identify data content, such as web applications and small-sized mobile games</p> <p>For more information, see Add an HTTP listener.</p>
HTTPS	<ul style="list-style-type: none"> • Encrypted data transmission that prevents unauthorized access • Centralized certificate management that allows certificate uploading and decryption on SLB 	<p>Applications that require encrypted transmission</p> <p>For more information, see Add an HTTPS listener.</p>

21.5.2. Add a TCP listener

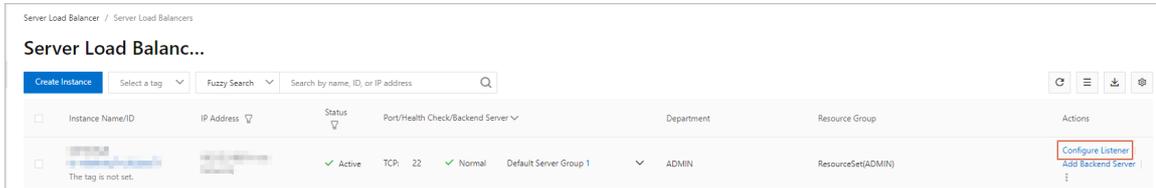
This topic describes how to add a TCP listener to an SLB instance. TCP provides reliable and accurate data delivery at relatively low connection speeds and therefore is applicable to services such as file transmission, email sending or receiving, and remote logon. You can add a TCP listener to forward TCP requests.

Step 1: Start the listener configuration wizard

To start the listener configuration wizard, perform the following operations:

1. [Log on to the SLB console](#).

2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Use one of the following methods to start the listener configuration wizard:
 - On the **Server Load Balancers** page, find the target SLB instance and then click **Configure Listener** in the **Actions** column.



- On the **Server Load Balancers** page, click the ID of the target SLB instance. On the **Listener** tab, click **Add Listener**.

Step 2: Configure the TCP listener

Perform the following operations to configure the TCP listener.

1. Specify the following information:

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, select TCP .
Listening Port	Set the listening port used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced Settings	
Scheduling Algorithm	SLB supports three scheduling algorithms: round robin (RR), weighted round robin (WRR), and weighted least connections (WLC). <ul style="list-style-type: none"> ○ Weighted Round-Robin (WRR): Backend servers with higher weights receive more requests. ○ Round-Robin (RR): Requests are evenly and sequentially distributed to backend servers. ○ Weighted Least Connections (WLC): Requests are distributed based on the combination of the weights and active connections of backend servers. Requests are distributed to the backend server with the least number of active connections. If two backend servers have the same number of active connections, the backend server with a higher weight receives more requests.
Enable Session Persistence	Specify whether to enable session persistence. Unit: seconds. Valid values: 1 to 3600. After session persistence is enabled, the listener forwards all requests from the same client to a specific backend server for the duration of a session. For TCP listeners, session persistence is implemented based on IP addresses. Requests from the same IP address are forwarded to the same backend server.

Parameter	Description
Enable Peak Bandwidth Limit	<p>You can switch on this option and then set a bandwidth limit for the listener.</p> <p>If an SLB instance incurs fees based on the bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic that flows in each listener. The sum of the peak bandwidth values of all listeners added to an SLB instance cannot exceed the bandwidth of this SLB instance.</p> <p>By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.</p>
Idle Timeout	Specify the idle timeout period of TCP connections. Unit: seconds. Valid values: 10 to 900.
Obtain Client Source IP Address	For Layer-4 listeners, backend servers can directly obtain the actual IP addresses of clients.
Automatically Enable Listener After Creation	Specify whether to start the listener after the listener is configured. By default, the listener is started after configuration.

2. Click **Next**.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create VServer groups or primary/secondary server groups and then add servers to them. For more information, see [Backend server overview](#).

This example adds backend servers to the default server group.

1. Select **Default Server Group** and click **Add More**.
2. Select ECS instances (backend servers) that you want to add, and then click **Next**.
3. Configure weights for the added backend servers. A backend server with a higher weight receives more requests.

 **Note** If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the **Default Server Group** tab, configure ports for the backend servers. Set a port for each backend server to receive requests. Valid values: 1 to 65535. You can specify the same port for multiple backend servers of an SLB instance.
5. Click **Next**.

Step 4: Configure the health check

SLB checks the availability of backend servers by performing the health check. The health check feature improves the availability of frontend services by minimizing downtime caused by health issues of backend servers. Click **Modify** to configure advanced health check settings. For more information, see [Health check overview](#).

Click **Next**.

Step 5: Confirm the settings

Complete the following steps to confirm and apply the listener settings:

1. In the **Submit** step, check the configuration. You can click **Modify** to modify configuration settings.
2. Click **Submit**.
3. In the **Configure Successful** dialog box, click **OK**.

You can check the created listener on the Listener tab.

21.5.3. Add a UDP listener

This topic describes how to add a UDP listener to an SLB instance. UDP is applicable to services that prioritize real-time content delivery over reliability, such as video chats and real-time quotes. You can add a UDP listener to forward UDP requests.

Context

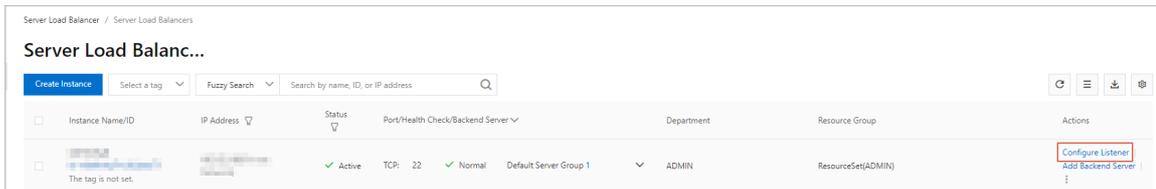
Before you configure a UDP listener, note the following limits:

- Ports 250, 4789, and 4790 of a UDP listener are reserved and therefore are unavailable for your configuration.
- Fragmented packets are not supported.
- The UDP listeners of an SLB instance in a classic network do not support the viewing of source IP addresses.
- The following operations take five minutes to take effect if they are performed for a UDP listener:
 - Remove backend servers
 - Set the weight of a backend server to 0 after it is detected unhealthy

Step 1: Start the listener configuration wizard

To start the listener configuration wizard, perform the following operations:

1. **Log on to the SLB console.**
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Use one of the following methods to start the listener configuration wizard:
 - On the **Server Load Balancers** page, find the target SLB instance and then click **Configure Listener** in the **Actions** column.



- On the **Server Load Balancers** page, click the ID of the target SLB instance. On the **Listener** tab, click **Add Listener**.

Step 2: Configure the UDP listener

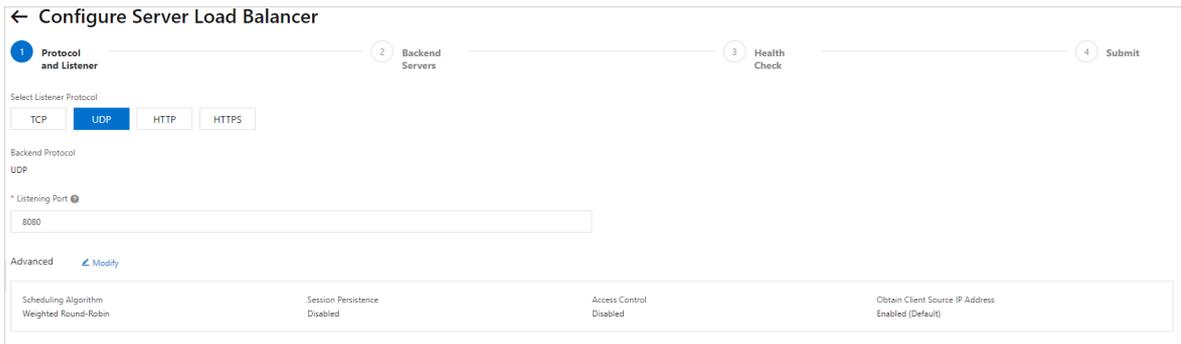
To configure the UDP listener, perform the following operations:

1. In the **Protocol and Listener** step, specify the following information:

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, select UDP .
Listening Port	Set the listening port used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced Settings	

Parameter	Description
Scheduling Algorithm	<p>SLB supports three scheduling algorithms: RR, WRR, and WLC.</p> <ul style="list-style-type: none"> ◦ Weighted Round-Robin (WRR): Backend servers with higher weights receive more requests. ◦ Round-Robin (RR): Requests are evenly and sequentially distributed to backend servers. ◦ Weighted Least Connections (WLC): Requests are distributed based on the combination of the weights and active connections of backend servers. Requests are distributed to the backend server with the least number of active connections. If two backend servers have the same number of active connections, the backend server with a higher weight receives more requests.
Enable Peak Bandwidth Limit	<p>You can switch on this option and then set a bandwidth limit for the listener.</p> <p>If an SLB instance incurs fees based on the bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic that flows in each listener. The sum of the peak bandwidth values of all listeners added to an SLB instance cannot exceed the bandwidth of this SLB instance.</p> <p>By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.</p>
Obtain Client Source IP Address	<p>Backend servers of a UDP listener can directly obtain the actual IP addresses of clients.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p>? Note UDP listeners of an SLB instance in a classic network do not support the viewing of source IP addresses.</p> </div>
Automatically Enable Listener After Creation	<p>Specify whether to start the listener after the listener is configured. By default, the listener is started after configuration.</p>

2. Click Next.



Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create VServer groups or primary/secondary server groups and then add servers to them. For more information, see [Backend server overview](#).

This example adds backend servers to the default server group.

1. Select **Default Server Group** and click **Add More**.
2. Select ECS instances (backend servers) that you want to add, and then click **Next**.

3. Configure weights for the added backend servers. A backend server with a higher weight receives more requests.

Note If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the **Default Server Group** tab, configure ports for the backend servers. Set a port for each backend server to receive requests. Valid values: 1 to 65535. You can specify the same port for multiple backend servers of an SLB instance.
5. Click **Next**.

Step 4: Configure the health check

SLB checks the availability of backend servers by performing the health check. The health check feature improves the availability of frontend services by minimizing downtime caused by health issues of backend servers. Click **Modify** to configure advanced health check settings. For more information, see [Health check overview](#).

Click **Next**.

Step 5: Confirm the settings

Complete the following steps to confirm and apply the listener settings:

1. In the **Submit** step, check the configuration. You can click **Modify** to modify configuration settings.
2. Click **Submit**.
3. In the **Configure Successful** dialog box, click **OK**.

You can check the created listener on the **Listener** tab.

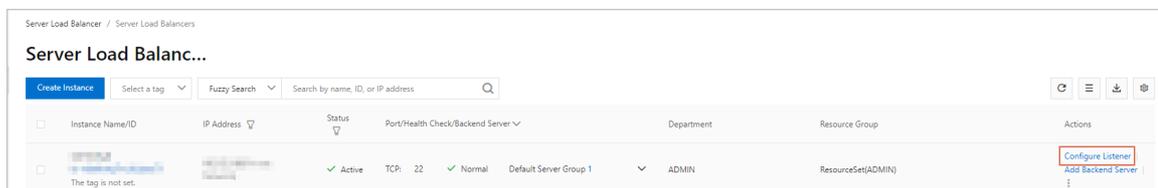
21.5.4. Add an HTTP listener

This topic describes how to add an HTTP listener to an SLB instance. HTTP is applicable to the applications that need to identify data content, such as web applications and small-sized mobile games. You can add an HTTP listener to forward HTTP requests.

Step 1: Start the listener configuration wizard

To start the listener configuration wizard, perform the following operations:

1. [Log on to the SLB console](#).
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Use one of the following methods to start the listener configuration wizard:
 - On the **Server Load Balancers** page, find the target SLB instance and then click **Configure Listener** in the **Actions** column.
- On the **Server Load Balancers** page, click the ID of the target SLB instance. On the **Listener** tab, click **Add Listener**.



Step 2: Configure the HTTP listener

Perform the following operations to configure the HTTP listener.

1. In the **Protocol and Listener** step, specify the following information:

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, select HTTP.
Listening Port	Set the listening port used to receive requests and forward them to backend servers. Valid values: 1 to 65535. Note The HTTP listening ports must be unique in an SLB instance.
Advanced Settings	
Scheduling Algorithm	SLB supports three scheduling algorithms: RR, WRR, and WLC. <ul style="list-style-type: none"> Weighted Round-Robin (WRR): Backend servers with higher weights receive more requests. Round-Robin (RR): Requests are evenly and sequentially distributed to backend servers. Weighted Least Connections (WLC): Requests are distributed based on the combination of the weights and active connections of backend servers. Requests are distributed to the backend server with the least number of active connections. If two backend servers have the same number of active connections, the backend server with a higher weight receives more requests.
Redirection	Specify whether to redirect traffic from the HTTP listener to an HTTPS listener. Note Before you enable redirection, make sure that you have created an HTTPS listener.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, the listener forwards all requests from the same client to a specific backend server for the duration of a session. HTTP implements cookie-based session persistence. You can choose from the following approaches to configure session persistence with cookies: <ul style="list-style-type: none"> Insert cookie: You only need to specify the cookie timeout period. SLB inserts a cookie (SERVERID) to the first HTTP or HTTPS response packet sent to a client. The next request from the client will contain this cookie, and the listener will distribute this request to the recorded backend server. Rewrite cookie: You can specify the cookie to be inserted into the HTTP or HTTPS response. You must specify the timeout period and lifecycle of this cookie on the backend server. SLB rewrites the original cookie when the new cookie is detected. The next request that contains the new cookie will be distributed to the recorded backend server.

Parameter	Description
Enable Peak Bandwidth Limit	<p>You can switch on this option and then set a bandwidth limit for the listener.</p> <p>If an SLB instance incurs fees based on the bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic that flows in each listener. The sum of the peak bandwidth values of all listeners added to an SLB instance cannot exceed the bandwidth of this SLB instance.</p> <p>By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.</p>
Idle Timeout	<p>Specify the idle timeout period of connections. Unit: seconds. Valid values: 1 to 60.</p> <p>If no request is received during the idle timeout period, SLB closes the connection and will restart the connection when the next request is received.</p> <p>This feature is available in all regions.</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Note This feature is unavailable for HTTP/2 requests.</p> </div>
Request Timeout	<p>Specify the request timeout period. Unit: seconds. Valid values: 1 to 180.</p> <p>If no response is received from the backend server during the request timeout period, SLB sends an HTTP 504 error code to the client.</p> <p>This feature is available in all regions.</p>
Enable Gzip Compression	<p>Specify whether to enable compression for a specific file type.</p> <p>Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.</p>
Add HTTP Header Fields	<p>Select one or more custom HTTP header fields that you want to add:</p> <ul style="list-style-type: none"> ◦ Use the <code>X-Forwarded-For</code> header field to retrieve the IP addresses of clients. ◦ Use the <code>X-Forwarded-Proto</code> header field to retrieve the listener protocol used by the SLB instance. ◦ Use the <code>SLB-IP</code> header field to retrieve the public IP address of the SLB instance. ◦ Use the <code>SLB-ID</code> header field to retrieve the ID of the SLB instance.
Obtain Client Source IP Address	<p>HTTP listeners use the X-Forwarded-For header field to identify the originating IP addresses of clients.</p>
Automatically Enable Listener After Creation	<p>Specify whether to start the listener after the listener is configured. By default, the listener is started after configuration.</p>

2. Click **Next**.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create VServer groups or primary/secondary server groups and then add servers to them. For more information, see [Backend server overview](#).

This example adds backend servers to the default server group.

1. Select **Default Server Group** and click **Add More**.
2. Select ECS instances (backend servers) that you want to add, and then click **Next**.
3. Configure weights for the added backend servers. A backend server with a higher weight receives more requests.

Note If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the **Default Server Group** tab, configure ports for the backend servers. Set a port for each backend server to receive requests. Valid values: 1 to 65535. You can specify the same port for multiple backend servers of an SLB instance.
5. Click **Next**.

Step 4: Configure the health check

SLB checks the availability of backend servers by performing the health check. The health check feature improves the availability of frontend services by minimizing downtime caused by health issues of backend servers. Click **Modify** to configure advanced health check settings. For more information, see [Health check overview](#).

Click **Next**.

Step 5: Confirm the settings

Complete the following steps to confirm and apply the listener settings:

1. In the **Submit** step, check the configuration. You can click **Modify** to modify configuration settings.
2. Click **Submit**.
3. In the **Configure Successful** dialog box, click **OK**.

You can check the created listener on the **Listener** tab.

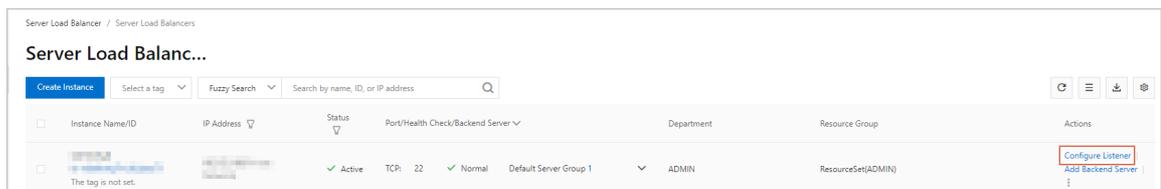
21.5.5. Add an HTTPS listener

HTTPS is applicable to applications that require encrypted data transmission. You can add an HTTPS listener to forward HTTPS requests.

Step 1: Start the listener configuration wizard

To start the listener configuration wizard, perform the following operations:

1. **Log on to the SLB console**.
2. In the left-side navigation pane, choose **Instances > Server Load Balancers**.
3. Use one of the following methods to start the listener configuration wizard:
 - On the **Server Load Balancers** page, find the target SLB instance and then click **Configure Listener** in the **Actions** column.



- On the **Server Load Balancers** page, click the ID of the target SLB instance. On the **Listener** tab, click **Add**

Listener.

Step 2 Configure an HTTPS listener

To configure an HTTPS listener, perform the following operations:

1. In the **Protocol and Listener** step, configure the listener parameters listed in the following table.

Parameter	Description
Select Listener Protocol	Select the protocol type of the listener. In this example, select HTTPS.
Listening Port	The listening port used to receive requests and forward the requests to backend servers. Valid values: 1 to 65535. <div style="background-color: #e0f2f7; padding: 5px; border: 1px solid #ccc;"> <p> Note The listening port numbers must be unique in an SLB instance.</p> </div>
Advanced	
Scheduling Algorithm	SLB supports three scheduling algorithms: round robin, weighted round robin (WRR), and weighted least connections (WLC). <ul style="list-style-type: none"> ◦ Weighted Round-Robin (WRR): A backend server with a higher weight is more likely to be scheduled and receives more requests. ◦ Round-Robin (RR): Requests are evenly and sequentially distributed to backend servers. ◦ Weighted Least Connections (WLC): Requests are distributed to the backend server with the least number of connections. If two backend servers have the same number of connections, the backend server with a higher weight will receive more requests.
Enable Session Persistence	Specifies whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from the same client to the same backend server. HTTP session persistence is implemented through cookies. SLB provides two methods to handle cookies: <ul style="list-style-type: none"> ◦ Insert cookie: You only need to specify the cookie timeout period. SLB inserts a cookie (SERVERID) to the first HTTP or HTTPS response packet sent to a client. The next request from the client will contain the cookie, and the listener will distribute the request to the recorded backend server. ◦ Rewrite cookie: You can specify the cookie to be inserted into the HTTP or HTTPS response to meet your specific needs. You must maintain the timeout period and lifecycle of the cookie on the backend server. SLB will overwrite the original cookie if it discovers that a new cookie has been specified. The next request will contain the new cookie, and the listener will distribute the request to the recorded backend server.

Parameter	Description
Enable Peak Bandwidth Limit	<p>Specifies whether to configure the bandwidth limit for the listener.</p> <p>If an SLB instance is billed based on bandwidth, you can set different peak bandwidth values for different listeners to limit the amount of traffic passing through each listener. The sum of the peak bandwidth values of all listeners for an SLB instance cannot exceed the bandwidth value of that SLB instance.</p> <p>By default, this feature is disabled, and all listeners share the bandwidth of the SLB instance.</p>
Enable Gzip Compression	<p>Specifies whether to enable compression for a specific file type.</p> <p>Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.</p>
Add HTTP Header Fields	<p>Select the custom HTTP header fields that you want to add:</p> <ul style="list-style-type: none"> ◦ Use the X-Forwarded-For header field to retrieve the IP addresses of clients. ◦ Use the X-Forwarded-Proto header field to retrieve the listener protocol used by the SLB instance. ◦ Use the SLB-IP header field to retrieve the public IP address of the SLB instance. ◦ Use the SLB-ID header field to retrieve the ID of the SLB instance.
Obtain Client Source IP Address	<p>HTTP listeners use the X-Forwarded-For header field to obtain the actual IP addresses of clients.</p>
Automatically Enable Listener After Creation	<p>Specifies whether to start the listener after the listener is configured. By default, the listener is started after configuration.</p>

2. Click Next.

Step 3 Configure an SSL certificate

To add an HTTPS listener, you must upload a server certificate or CA certificate.

 Notice SLB supports the following public key algorithms:

- RSA 1024
- RSA 2048
- RSA 4096
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

Certificate	Description	Required for one-way authentication	Required for mutual authentication

Certificate	Description	Required for one-way authentication	Required for mutual authentication
Server certificate	The certificate used to identify the server. The client checks whether the certificate sent by the server is issued by a trusted center.	Yes. You must upload the server certificate to the certificate management system of SLB.	Yes. You must upload the server certificate to the certificate management system of SLB.
Client certificate	The certificate that is used to identify the client. The server identifies the client by checking the certificate sent by the client. You can sign a client certificate with a self-signed CA certificate.	No.	Yes. You must install the client certificate on the client.
CA certificate	The server uses a CA certificate to verify the signature on the client certificate. If the certificate cannot be verified, the connection request is rejected.	No.	Yes. You must upload the CA certificate to the certificate management system of SLB.

Before you upload a certificate, take note of the following items:

- The uploaded certificate must be in the PEM format.
- After you upload the certificate to SLB, SLB can manage the certificate and you do not need to bind the certificate to backend servers.
- It may take some time to activate an HTTPS listener because the certificate must be uploaded, loaded, and verified. It can take up to three minutes for an HTTPS listener to be activated.
- The ECDHE cipher suites used by HTTPS listeners support forward secrecy but do not support the security enhancement parameters required by DHE cipher suites. As a result, strings that contain the `BEGIN DH PARAMETERS` field in a PEM certificate file cannot be uploaded. For more information, see [Certificate requirements](#).
- HTTPS listeners do not support Server Name Indication (SNI). You can use TCP listeners instead, and then configure SNI on backend servers.
- The session ticket timeout period of HTTPS listeners is 300 seconds.
- The actual amount of traffic generated on an HTTPS listener is larger than the billed traffic amount because some traffic is used for handshaking.
- If a large number of new connections are established, a large amount of traffic will be generated.
 1. Select an uploaded server certificate, or click **Create Server Certificate** to upload a server certificate. For more information, see [Certificate overview](#).
 2. To enable HTTPS mutual authentication, click **Modify** next to **Advanced**.
 3. Turn on **Enable Mutual Authentication**, and select an uploaded CA certificate or click **Create CA Certificate** to upload a CA certificate. You can use a self-signed CA certificate. For more information, see [Certificate overview](#).

Step 4 Add backend servers

After configuring the listener, you must add backend servers to process client requests. You can use the default server group configured for the SLB instance, or configure a VServer group or a primary/secondary server group. For more information, see [Backend server overview](#).

This example uses the default server group.

1. Select **Default Server Group** and click **Add More**.
2. Select ECS instances (backend servers) that you want to add, and then click **Next**.

3. Configure weights for the added backend servers. A backend server with a higher weight receives more requests.

Note If the weight of a backend server is set to 0, the backend server does not receive new requests.

4. Click **Add**. On the **Default Server Group** tab, configure ports for the backend servers. Set a port for each backend server to receive requests. Valid values: 1 to 65535. You can specify the same port for multiple backend servers of an SLB instance.
5. Click **Next**.

Step 5 Configure health checks

SLB checks the availability of backend servers by performing the health check. The health check feature improves the availability of frontend services by minimizing downtime caused by health issues of backend servers. Click **Modify** to configure advanced health check settings. For more information, see [Health check overview](#).

Step 6 Submit the configurations

Submit the configurations by performing the following operations:

1. In the **Submit** step, check the configuration. You can click **Modify** to modify configuration settings.
2. Click **Submit**.
3. In the **Configure Successful** dialog box, click **OK**.

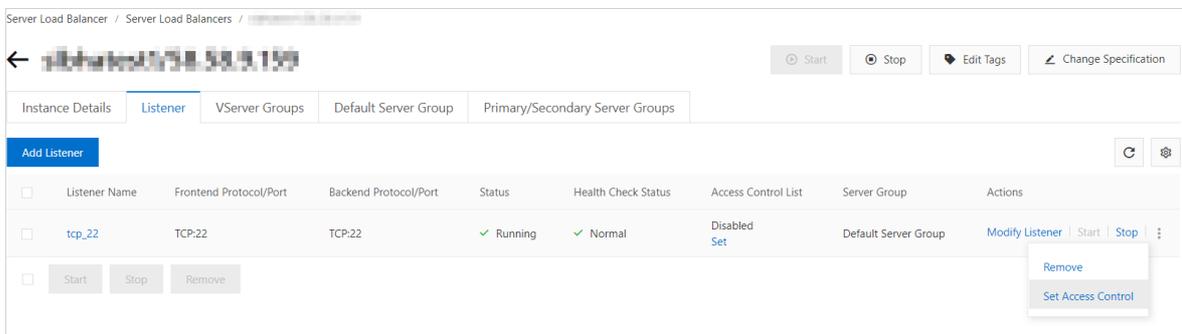
You can check the created listener on the **Listener** tab.

21.5.6. Enable access control

This topic describes how to enable access control for a listener. SLB provides listener-based access control. You can configure different whitelists for different listeners.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. On the page that appears, click the **Listener** tab.
4. Find the target listener and choose  > **Set Access Control** in the **Actions** column.



5. In the **Access Control Settings** dialog box, enable access control and configure the whitelist, and then click **OK**.

Whitelist: Only the requests from the IP addresses or CIDR blocks in the specified ACL are forwarded. You can use the whitelist feature when you want to allow access from specified IP addresses.

Using the whitelist feature may pose risks to your services. The whitelist allows only the traffic from the IP addresses in the specified ACL to access the SLB listener. If the whitelist is used while the corresponding ACL does not contain any IP addresses, the SLB listener forwards all access requests.

Separate multiple IP addresses with commas (,). Each IP address must be unique. You can add a maximum of 300 IP addresses.

- Supported formats: IP addresses (for example, 10.23.12.24) and CIDR blocks (for example, 10.23.12.0/24)
- 0.0.0.0 and x.x.x.x/0 are not supported.

 **Note** The access control feature works only for new connection requests and does not affect existing connections.

21.5.7. Disable access control

This topic describes how to disable access control for a listener.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. On the page that appears, click the **Listener** tab.
4. Find the target listener, and choose  > **Set Access Control** in the Actions column.
5. In the **Access Control Settings** dialog box, disable access control and then click **OK**.

21.6. Backend servers

21.6.1. Backend server overview

Before the Server Load Balancer (SLB) service can be used, you must add ECS instances as backend servers to an SLB instance to process client requests.

You can set virtual IP addresses for SLB instances. This way, the added ECS instances in the same region can be virtualized into an application service pool that provides high performance and availability. You can manage backend servers by using VServer groups. Each listener of an SLB instance can be associated with a specific VServer group so that different listeners can forward requests to their associated backend servers that use different ports.

 **Note** If you associate a VServer group with a listener, the listener will distribute requests to backend servers in the associated VServer group instead of those in the default server group.

You can increase or decrease the number of backend ECS instances at any time and switch ECS instances to receive client requests. However, we recommend that you enable the health check feature and make sure that at least one ECS instance is running properly to maintain service stability.

When you add ECS instances to an SLB instance, take note of the following items:

- You can use different operating systems for the backend ECS instances of an SLB instance. However, the applications deployed in the ECS instances must be the same and have consistent data. We recommend that you use the same operating system to facilitate management and maintenance.
- Up to 50 listeners can be added to a single SLB instance. Each listener corresponds to an application deployed on backend ECS instances. Listening ports of an SLB instance correspond to application service ports opened on backend ECS instances.
- You can specify a weight for each ECS instance in the application service pool. An ECS instance with a higher weight receives more requests.
- If session persistence is enabled, requests may not be evenly distributed to backend ECS instances. To solve

this problem, we recommend that you disable session persistence and check whether the problem persists.

If requests are not distributed evenly, troubleshoot as follows:

- i. Collect statistics on the access logs of the web service on backend ECS instances for a period of time.
 - ii. Check whether the numbers of access logs of backend ECS instances match SLB configurations. For example, if session persistence is enabled, you must differentiate the access logs for the same IP address. If different weights are configured for backend ECS instances, you must check whether the percentage of access logs is normal based on the percentage of weights.
- When an ECS instance is undergoing hot migration, persistent connections to SLB may be interrupted. You can solve this problem by reestablishing the connections.

Default server groups

A default server group contains ECS instances that are used to receive client requests. If a listener is not associated with a VServer group or a primary/secondary server group, the listener will forward requests to ECS instances in the default server group.

For more information about how to create a default server group, see [Add ECS instances to the default server group](#).

Primary/secondary server groups

A primary/secondary server group only contains two ECS instances. One ECS instance acts as the primary server and the other acts as the secondary server. Health checks are not performed on the secondary server. If the primary server is declared unhealthy, traffic is redirected to the secondary server. If the primary server is declared healthy, services will be restored and traffic will be forwarded to the primary server again.

For more information about how to create a primary/secondary server group, see [Add ECS instances to a primary/secondary server group](#).

 **Note** Only TCP and UDP listeners support configuring primary/secondary server groups.

VServer groups

If you want to distribute different requests to different backend servers or configure domain name- or URL-based forwarding rules, you can use VServer groups.

For more information about how to create a VServer group, see [Add ECS instances to a VServer group](#).

21.6.2. Default server groups

21.6.2.1. Add ECS instances to the default server group

This topic describes how to add ECS instances as default backend servers to the default server group of an SLB instance. Before you use the SLB service, you must add at least one default backend server to receive client requests forwarded by SLB.

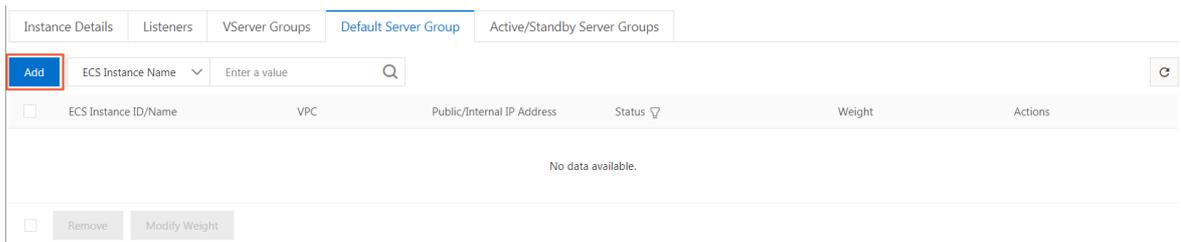
Prerequisites

Before you add ECS instances to the default server group, make sure that the following conditions are met:

- An SLB instance is created. For more information, see [Create an SLB instance](#).
- You have created ECS instances and deployed applications on these ECS instances to process requests.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add**.



5. In the **My Servers** dialog box, select ECS instances that you want to add.
6. Click **Next**.
7. In the **Configure Ports and Weights** dialog box, specify the weight of each added ECS instance. An ECS instance with a higher weight receives more requests.

You can change the weight of a server and then move the pointer over  to change the weights of multiple servers:

- **Click Replicate to Below:** The weights of all servers below the current server are set to the weight of the current server.
- **Click Replicate to Above:** The weights of all servers above the current server are set to the weight of the current server.
- **Click Replicate to All:** The weights of all servers in the default server group are set to the weight of the current server.
- **Click Reset:** The weight fields of all servers in the default server group are cleared.

 **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

8. Click **Add**.
9. Click **OK**.

21.6.2.2. Add IDC servers to the default server group

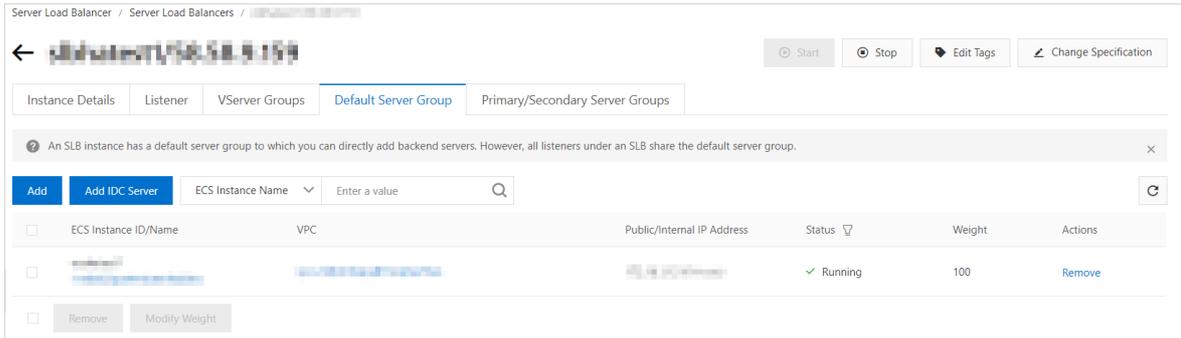
This topic describes how to add servers in on-premises Internet Data Centers (IDCs) as default backend servers to the default server group of an SLB instance. Before you use the SLB service, you must add at least one default backend server to receive client requests forwarded by SLB.

Prerequisites

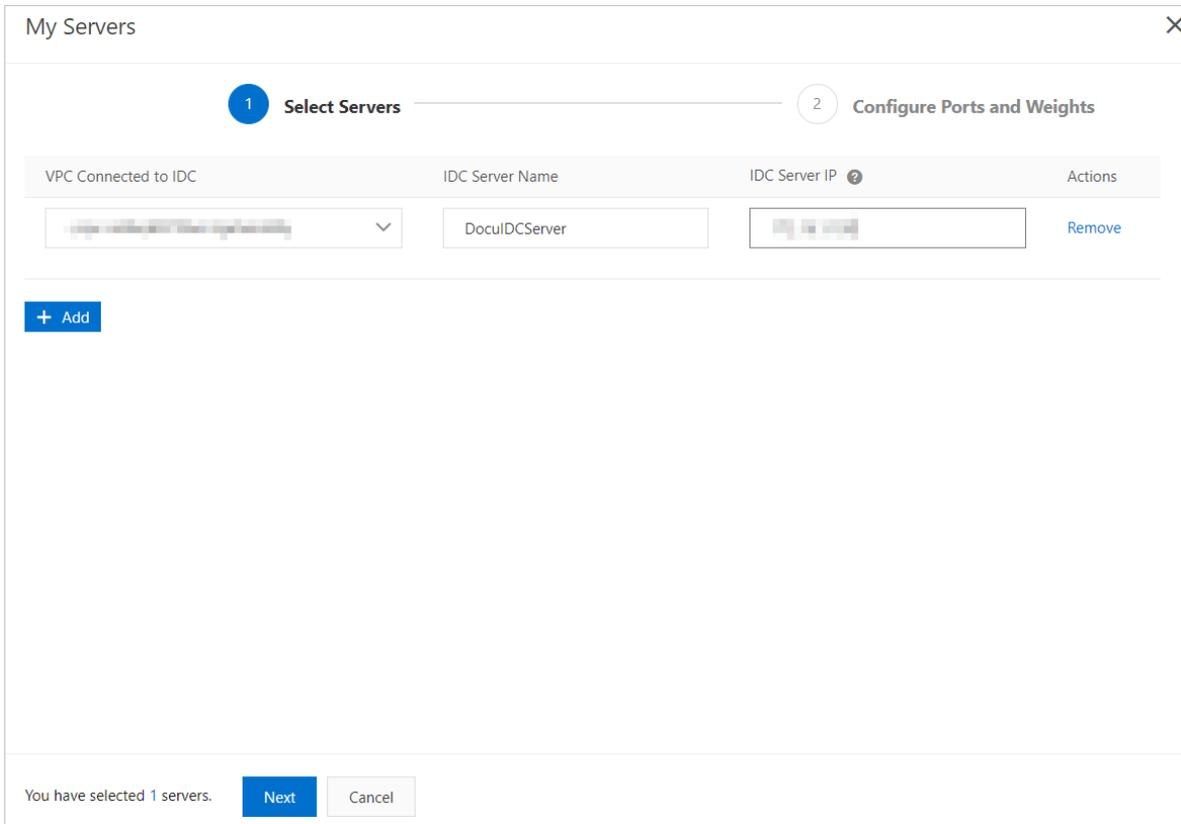
Applications are deployed on the IDC servers, and the IDC servers are ready to receive distributed requests.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add IDC Server**.



- In the **My Servers** dialog box, click **Add**.
- Select a VPC from the **VPC Connected to IDC** drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server.



- Click **Next**.
- In the **Configure Ports and Weights** step, specify the weight of each added IDC server. An IDC server with a higher weight receives more requests.

You can change the weight of a server and then move the pointer over  to change the weights of multiple servers:

- Click **Replicate to Below**: The weights of all servers below the current server are set to the weight of the current server.
- Click **Replicate to Above**: The weights of all servers above the current server are set to the weight of the current server.
- Click **Replicate to All**: The weights of all servers in the default server group are set to the weight of the current server.
- Click **Reset**: The weight fields of all servers in the default server group are cleared.

 **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

9. Click **Add**.
10. Click **OK**.

21.6.2.3. Change the weight of a backend server

This topic describes how to change the weight of a backend server to adjust the proportion of requests sent to the backend server.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Move the pointer over the weight value of the target backend server and click the  icon.
5. Change the weight and then click **OK**. A backend server (ECS instance or IDC server) with a higher weight receives more requests.

 **Notice** The weight value ranges from 0 to 100. If the weight of a backend server is set to 0, no requests are sent to the backend server.

21.6.2.4. Remove a backend server

This topic describes how to remove a backend server that is no longer needed.

Procedure

1. [Log on to the SLB console](#).
2. Find the target SLB instance and click its instance ID.
3. Click the **Default Server Group** tab.
4. Find the target backend server and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

21.6.3. VServer groups

21.6.3.1. Add ECS instances to a VServer group

This topic describes how to create a VServer group and then add ECS instances as backend servers to the VServer group. If you associate a VServer group with a listener, the listener distributes requests only to the backend servers in the VServer group instead of other backend servers.

Prerequisites

Before you create a VServer group, make sure that the following conditions are met:

- An SLB instance is created. For more information, see [Create an SLB instance](#).
- You have created ECS instances and deployed applications on these ECS instances to process requests.

Context

Note the following items before you create a VServer group:

- An ECS instance can be added to multiple VServer groups.
- A VServer group can be associated with multiple listeners of an SLB instance.
- The settings of the VServer group include the settings of ECS instances and application ports.

Procedure

1. **Log on to the SLB console.**
2. Find the target SLB instance and click its instance ID.
3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, configure the VServer group.
 - i. In the **VServer Group Name** field, enter a name for the VServer group.
 - ii. Click **Add**. In the **My Servers** dialog box, select ECS instances that you want to add.
 - iii. Click **Next**.
 - iv. Specify a port and a weight for each ECS instance and then click **Add**. Set the ports and weights based on the following information:
 - **Port:** The backend port opened on an ECS instance to receive requests.
You can set the same port number for multiple backend servers of the same SLB instance. In addition, you can click **Add Port** to add multiple ports for a backend server.
 - **Weight:** An ECS instance with a higher weight receives more requests.

 **Notice** If the weight of an ECS instance is set to 0, the ECS instance no longer receives new requests.

You can change the weight of a server and then move the pointer over  to change the weights of multiple servers:

- **Click Replicate to Below:** The weights of all servers below the current server are set to the weight of the current server.
- **Click Replicate to Above:** The weights of all servers above the current server are set to the weight of the current server.
- **Click Replicate to All:** The weights of all servers in the VServer group are set to the weight of the current server.
- **Click Reset:** The weight fields of all servers in the VServer group are cleared.

 **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

- v. Click **Add**.
6. Click **Create**.

21.6.3.2. Add IDC servers to a VServer group

This topic describes how to create a VServer group and then add IDC servers to the VServer group. You can add ECS instances and IDC servers as backend servers to a VServer group. If you associate a VServer group with a listener, the listener distributes requests only to the backend servers in the VServer group instead of other backend servers.

Prerequisites

Before you create a VServer group, make sure that applications are deployed on the IDC servers and the IDC servers are ready to receive distributed requests.

Context

Note the following items before you create a VServer group:

- An IDC server can be added to multiple VServer groups.
- A VServer group can be associated with multiple listeners of an SLB instance.
- The settings of the VServer group include the settings of IDC servers and application ports.

Procedure

1. **Log on to the SLB console.**
2. Find the target SLB instance and click its instance ID.
3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, configure the VServer group.
 - i. In the **VServer Group Name** field, enter a name for the VServer group.
 - ii. Click **Add IDC Server**.
 - iii. In the **My Servers** dialog box, click **Add**.
 - iv. Select a VPC from the **VPC Connected to IDC** drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server. The IP address of the IDC server must be accessible to the VPC.
 - v. Click **Next**.
 - vi. Specify a port and weight for each IDC server, and then click **Add**. Set the ports and weights based on the following information:
 - **Port:** The backend port opened on an IDC server to receive requests. Multiple ports can be added to an IDC server.
You can set the same port number for multiple backend servers of the same SLB instance.
 - **Weight:** An IDC server with a higher weight receives more requests.

 **Notice** If the weight of an IDC server is set to 0, the IDC server no longer receives new requests.

You can change the weight of a server and then move the pointer over  to change the weights of multiple servers:

- **Click Replicate to Below:** The weights of all servers below the current server are set to the weight of the current server.
- **Click Replicate to Above:** The weights of all servers above the current server are set to the weight of the current server.
- **Click Replicate to All:** The weights of all servers in the VServer group are set to the weight of the current server.
- **Click Reset:** The weight fields of all servers in the VServer group are cleared.

 **Notice** If the weight of a backend server is set to 0, this backend server no longer receives new requests.

vii. Click **Add**.

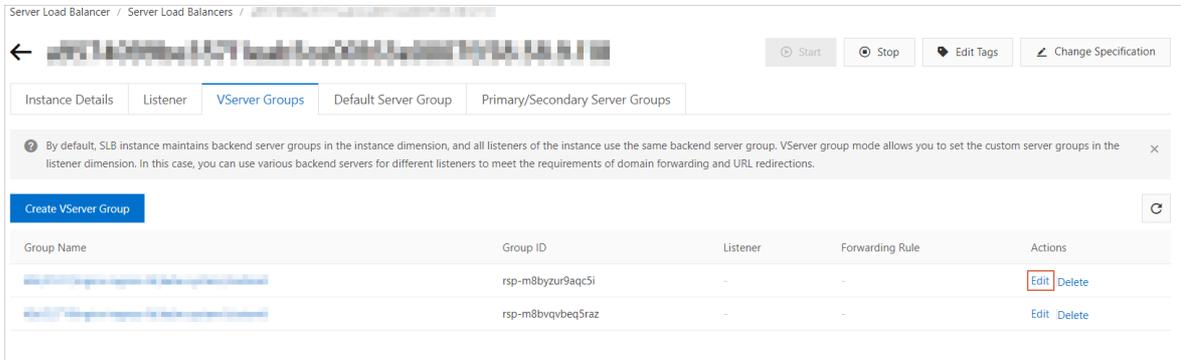
6. Click **Create**.

21.6.3.3. Modify a VServer group

This topic describes how to modify the settings of ECS instances or IDC servers in a VServer group.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the VServer Groups tab.
4. Find the target VServer group and then click Edit in the Actions column.



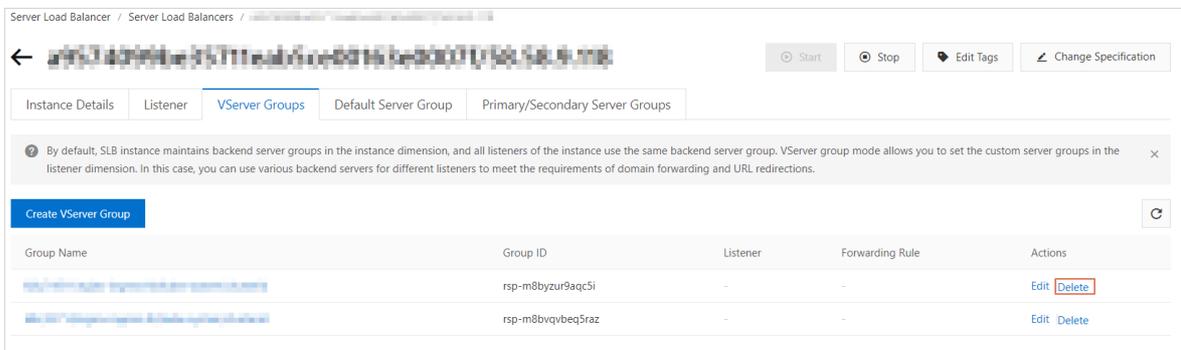
5. Modify the ports and weights of ECS instances or IDC servers, and then click Save.

21.6.3.4. Delete a VServer group

This topic describes how to delete a VServer group that is no longer needed for traffic distribution.

Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the VServer Groups tab.
4. Find the target VServer group, and then click Delete in the Actions column.



5. In the dialog box that appears, click OK.

21.6.4. Active/standby server groups

21.6.4.1. Add ECS instances to a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add ECS instances to the primary/secondary server group. You can use a primary/secondary server group to implement failover between a primary server and a secondary server. By default, the primary server handles all distributed requests. When the primary server fails, traffic is redirected to the secondary server.

Prerequisites

Before you create a primary/secondary server group, make sure that the following conditions are met:

- An SLB instance is created. For more information, see [Create an SLB instance](#).
- You have created ECS instances and deployed applications on these ECS instances to process requests.

Procedure

1. **Log on to the SLB console.**
2. Find the target SLB instance and click its instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, click **Create Primary/Secondary Server Group**.
5. On the **Create Primary/Secondary Server Group** page, configure the primary/secondary server group.
 - i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group.

← Create Primary/Secondary Server Group

Note: The network type of the SLB instance is Classic Network, and the instance type is Private Network. You can add ECS instances in a classic or VPC network to the primary/secondary server group.

* Primary/Secondary Server Group Name

Enter a server group name

Added Servers

Add Add IDC Server Search by server name, ID, or IP

ECS Instance ID/Name	Region	VPC	Public/Private IP	Status	Port	Type	Actions
No data available.							

Create Cancel

- ii. Click Add. In the My Servers dialog box, select ECS instances that you want to add in the Select Servers step.

My Servers

1 Select Servers ————— 2 Configure Ports and Weights

ECS Instance Name Search by name, ID, or IP VPC Select

Show Available Instances Only Advanced Mode

<input type="checkbox"/>	ECS Instance ID/Name	Private IP	Public IP/VPC Property	Status	Number of SLB Associations
<input type="checkbox"/>	ecshatest1	192.168.1.1	192.168.1.1	Running	1
<input checked="" type="checkbox"/>	ecshatest2	192.168.1.2	192.168.1.2	Running	1
<input checked="" type="checkbox"/>	ecshatest3	192.168.1.3	192.168.1.3	Running	1

You have selected 2 servers.

You can add up to two ECS instances to a primary/secondary server group.

- iii. Click Next.

- iv. Configure the backend ports opened on ECS instances to receive requests, and then click Add.

My Servers

1 Select Servers ————— 2 Configure Ports and Weights

ECS Instance ID/Name	Private IP	Port	Reset	Actions
ecshatest3	[Redacted]	<input type="text"/>	Reset	Add Port Remove
ecshatest2	[Redacted]	<input type="text"/>	Reset	Add Port Remove

Previous Add Cancel

You can set multiple ports for an ECS instance.

- v. Set an ECS instance as the primary server.
vi. Click Create.

21.6.4.2. Add IDC servers to a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add IDC servers to the primary/secondary server group. You can use a primary/secondary server group to implement failover between a primary server and a secondary server. By default, the primary server handles all distributed requests. When the primary server fails, traffic is redirected to the secondary server.

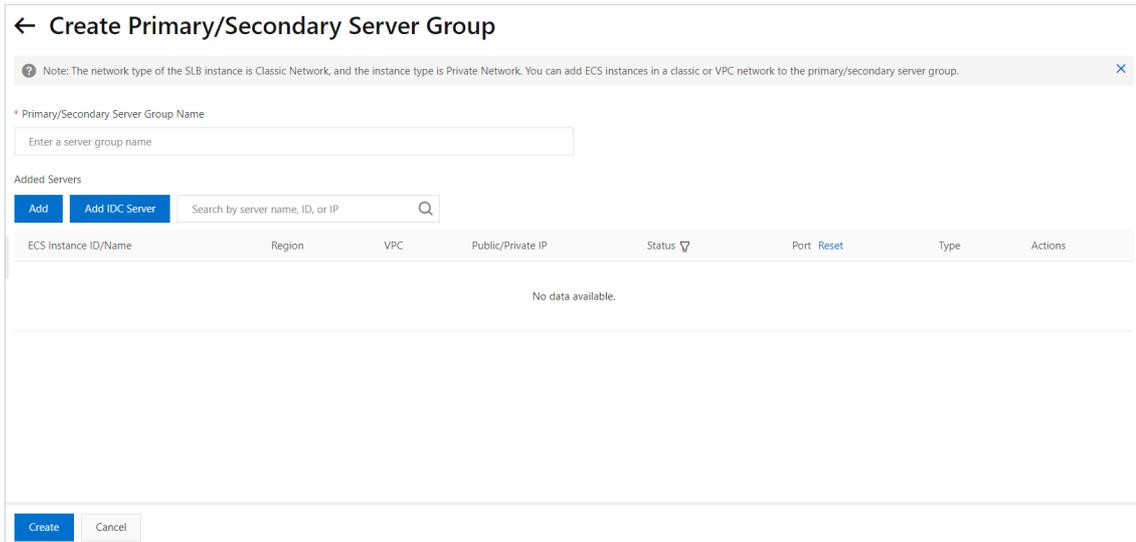
Prerequisites

The IDC servers are created, configured to deploy applications, and ready to receive distributed requests.

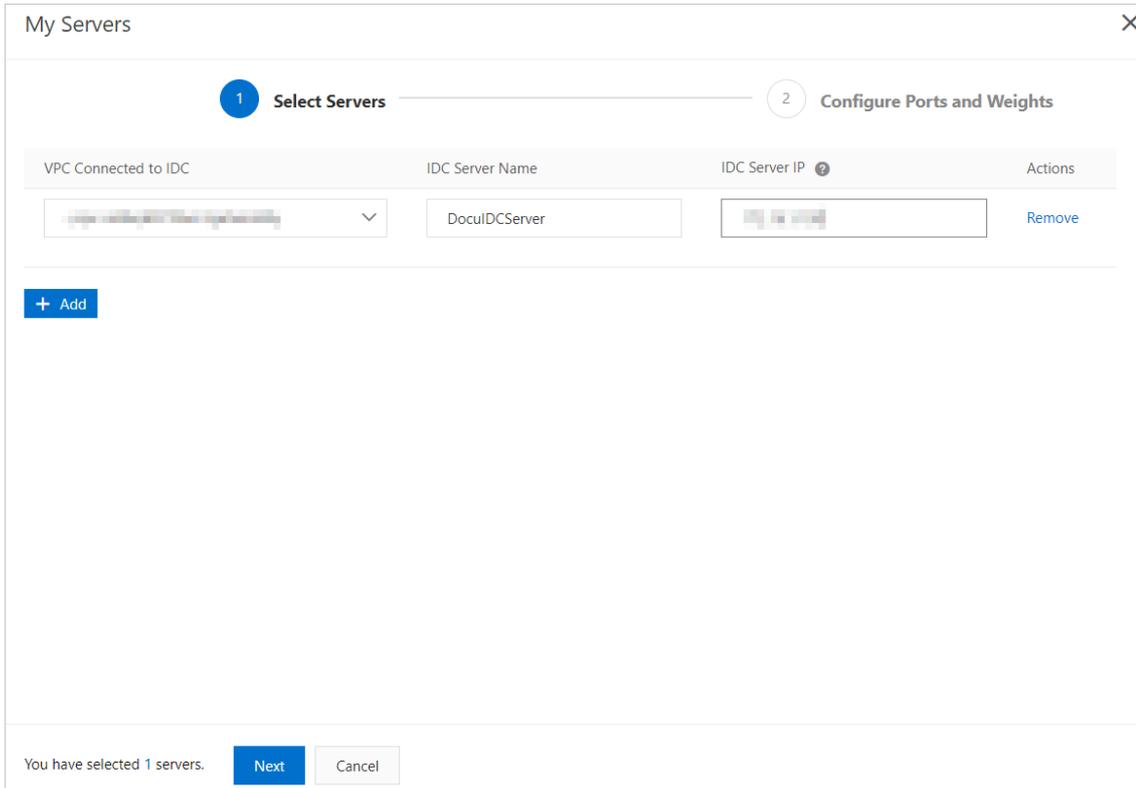
Procedure

1. Log on to the SLB console.
2. Find the target SLB instance and click its instance ID.
3. Click the Primary/Secondary Server Groups tab.
4. On the Primary/Secondary Server Groups tab, click Create Primary/Secondary Server Group.
5. On the Create Primary/Secondary Server Group page, configure the primary/secondary server group.

- i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group, and then click **Add IDC Server**.

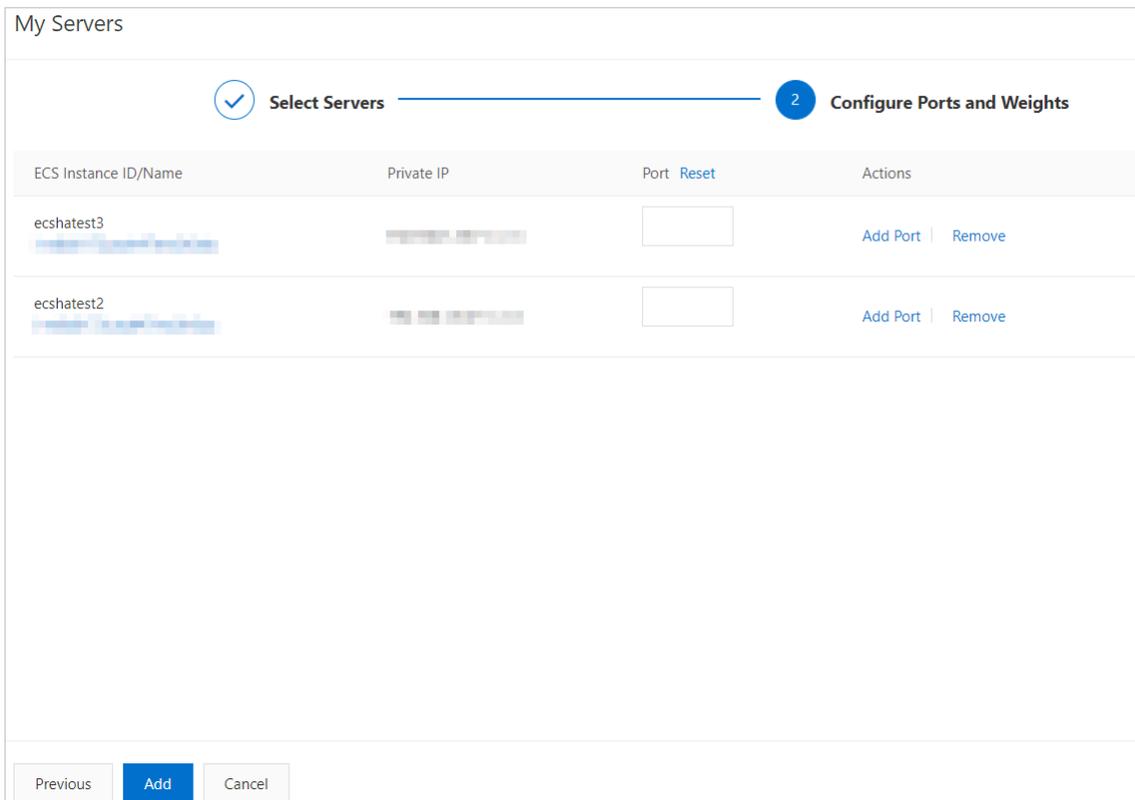


- ii. In the **My Servers** dialog box, click **Add**.



- iii. Select a VPC from the **VPC Connected to IDC** drop-down list, enter a name for the IDC server, and specify the IP address of the IDC server. The IP address of the IDC server must be accessible to the VPC.
- iv. Click **Next**.

v. Configure the backend ports opened on ECS instances to receive requests, and then click **Add**.



You can set multiple ports for an IDC server.

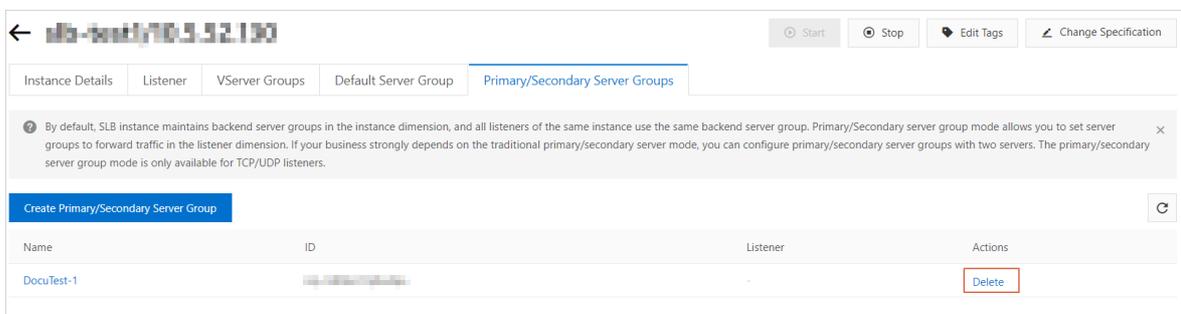
- vi. Set a backend server as the primary server.
- vii. Click **Create**.

21.6.4.3. Delete a primary/secondary server group

If a primary/secondary server group is no longer needed to forward traffic, you can delete the primary/secondary server group.

Procedure

1. **Log on to the SLB console.**
2. Find the target SLB instance and click its instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. Find the target primary/secondary server group and click **Delete** in the Actions column.



5. In the dialog box that appears, click **OK**.

21.7. Health check

21.7.1. Health check overview

Server Load Balancer (SLB) checks the availability of backend servers (ECS instances) by performing health checks. The health check feature improves the overall availability of your frontend business and mitigates the impact of exceptions occurring on backend ECS instances.

After the health check feature has been enabled, SLB stops distributing requests to ECS instances that are declared unhealthy and distributes new requests to healthy ECS instances. When the unhealthy ECS instances have recovered, SLB begins forwarding requests to the ECS instances again.

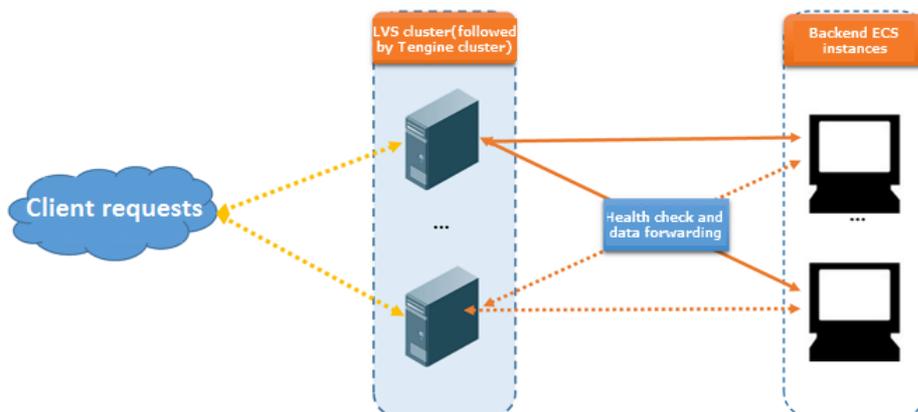
If your business is highly sensitive to traffic loads, frequent health checks may impact the availability of normal business. To reduce the impacts of health checks on your business, you can reduce the health check frequency, increase the health check interval, or change Layer 7 health checks to Layer 4 health checks to match business conditions. We recommend that you do not disable the health check feature to ensure business continuity.

Health check process

SLB is deployed in clusters. Node servers in the LVS or Tengine cluster forward data and perform health checks.

The node servers in the LVS cluster forward data and perform health checks independently and in parallel based on configured load balancing policies. If an LVS node server detects that a backend ECS instance is unhealthy, the node server no longer sends new client requests to the ECS instance.

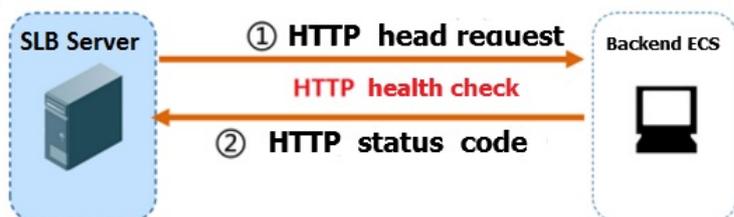
SLB uses the CIDR block of 100.64.0.0/10 for health checks. Make sure that backend ECS instances do not block this CIDR block. You do not need to configure a security group rule to allow access from this CIDR block. However, if you have configured security rules such as iptables, you must allow access from this CIDR block. 100.64.0.0/10 is reserved by Alibaba Cloud. Other users cannot use IP addresses within this CIDR block, and therefore there are no security risks associated with it.



Health checks of HTTP or HTTPS listeners

For Layer 7 (HTTP or HTTPS) listeners, SLB checks the status of backend ECS instances by sending HTTP HEAD requests.

For HTTPS listeners, certificates are managed in SLB. To improve system performance, HTTPS is not used for data exchange (including health check data and business interaction data) between SLB and backend ECS instances.



The health check process of a Layer 7 listener is as follows:

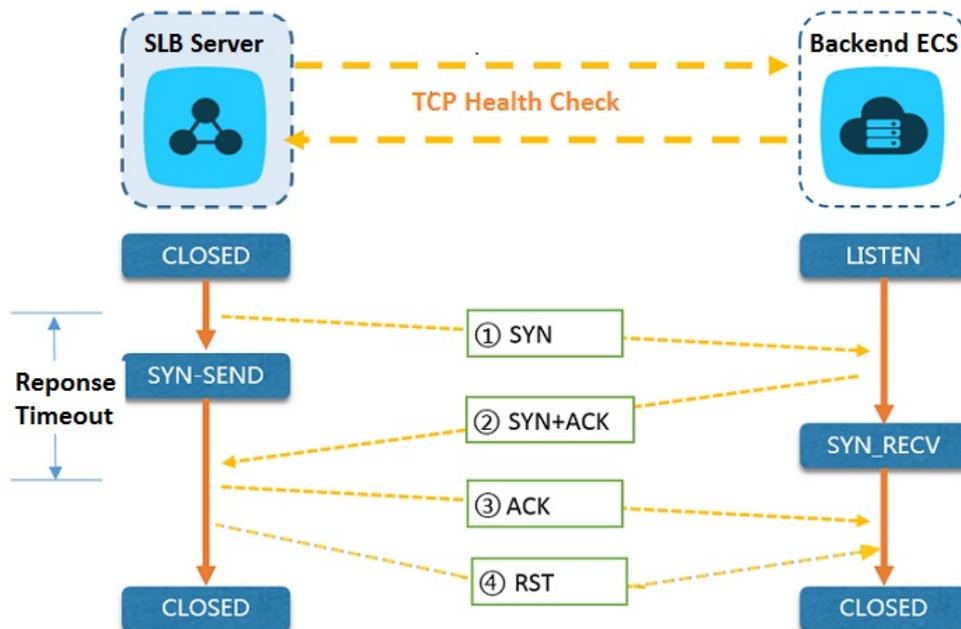
1. A Tengine node server sends an HTTP HEAD request containing the configured domain name to the internal

IP address, health check port, and health check path of a backend ECS instance based on health check settings.

2. After receiving the request, the backend ECS instance returns an HTTP status code based on the running status.
3. If the Engine node server does not receive a response from the backend ECS instance within the specified response timeout period, the node server determines that the service is unresponsive and the health check fails.
4. If the Engine node server receives a response from the backend ECS instance within the specified response timeout period, the node server compares the response information with the configured status code. If the status code in the response is the same as the configured one, the backend ECS instance is declared healthy. Otherwise, the backend ECS instance is declared unhealthy.

Health checks of TCP listeners

For TCP listeners, SLB checks the status of backend ECS instances by establishing TCP connections to improve health check efficiency.



The health check process of a TCP listener is as follows:

1. An LVS node server sends a TCP SYN packet to the internal IP address and health check port of a backend ECS instance.
2. After receiving the request, the backend ECS instance returns an SYN+ACK packet if listening on the corresponding port is normal.
3. If the LVS node server does not receive a packet from the backend ECS instance within the specified response timeout period, the node server determines that the service is unresponsive and the health check fails. Then, the node server sends an RST packet to the backend ECS instance to terminate the TCP connection.
4. If the LVS node server receives a packet from the backend ECS instance within the specified response timeout period, the node server determines that the service runs properly and the health check succeeds. Then, the node server sends an RST packet to the backend ECS instance to terminate the TCP connection.

Note Typically, a TCP three-way handshake is conducted to establish a TCP connection. After the LVS node server receives the SYN+ACK packet from the backend ECS instance, the node server sends an ACK packet, and then immediately sends an RST packet to terminate the TCP connection.

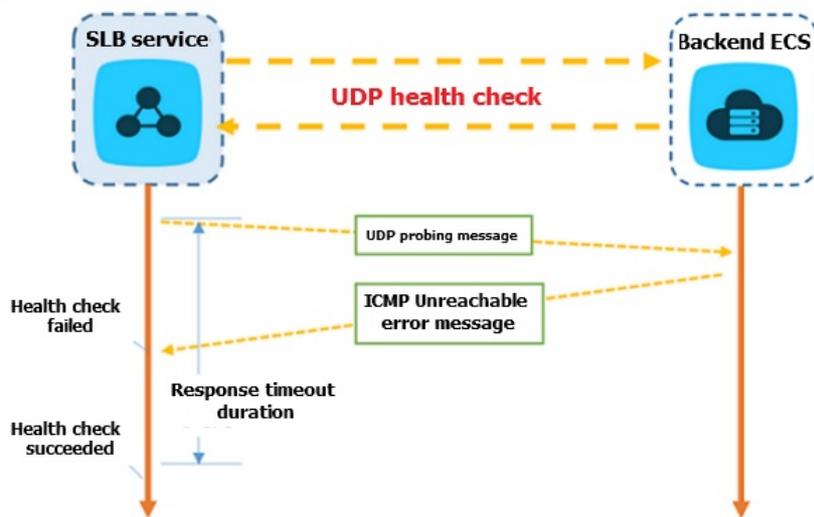
This process may cause the ECS instance to regard that an error such as an abnormal exit occurred in the TCP connection, and then report a corresponding error such as `Connection reset by peer`, for example, in Java connection pool logs.

Solution:

- You can implement HTTP health checks.
- If you have enabled the feature of obtaining actual client IP addresses on backend ECS instances, you can ignore connection errors caused by the access of the SLB CIDR block.

Health checks of UDP listeners

For UDP listeners, SLB checks the status of backend ECS instances by sending UDP packets.



The health check process of a UDP listener is as follows:

1. An LVS node server sends a UDP packet to the internal IP address and health check port of an ECS instance based on health check configurations.
2. If the listening on the corresponding port of the ECS instance is abnormal, the ECS instance returns an ICMP error message such as `port XX unreachable`. Otherwise, no message is returned.
3. If the LVS node server receives the ICMP error message within the response timeout period, the node server determines that the service is abnormal and the health check fails.
4. If the LVS node server does not receive any message within the response timeout period, the node server determines that the service is normal and the health check succeeds.

Note For UDP health checks, the health check result may not reflect the real status of a backend ECS instance in the following situation:

If the backend ECS instance uses a Linux operating system, the speed at which ICMP messages in high concurrency scenarios are sent is limited due to the ICMP attack prevention feature of Linux. In this case, even if a service exception occurs, SLB may declare the backend ECS instance healthy because the error message `port XX unreachable` is not returned. Consequently, the health check result deviates from the actual service status.

Solution:

You can specify a request and a response for UDP health checks. The ECS instance is considered healthy only when the specified response is returned. However, the client must be configured accordingly for returning responses.

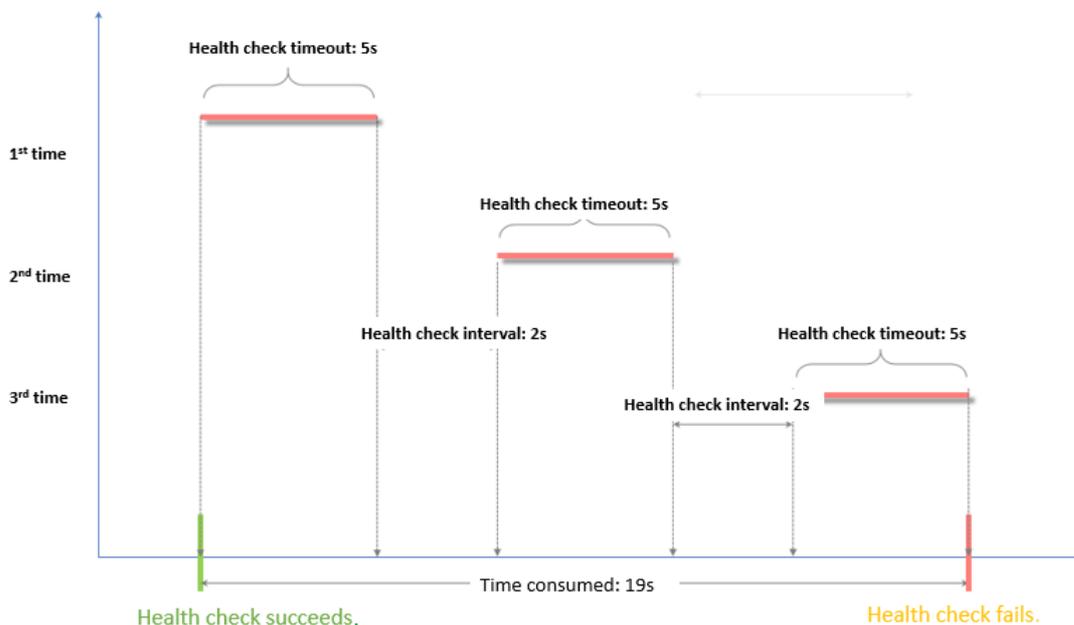
Health check time window

The health check feature effectively improves the availability of your services. However, to avoid impacts on system availability caused by frequent switching after failed health checks, the health check status only switches when health checks successively succeed or fail for a specified number of times within a certain time window. The health check time window is determined by the following three factors:

- Health check interval: how often health checks are performed
- Response timeout: the time to wait for a response
- Health check threshold: the number of times health checks succeed or fail successively

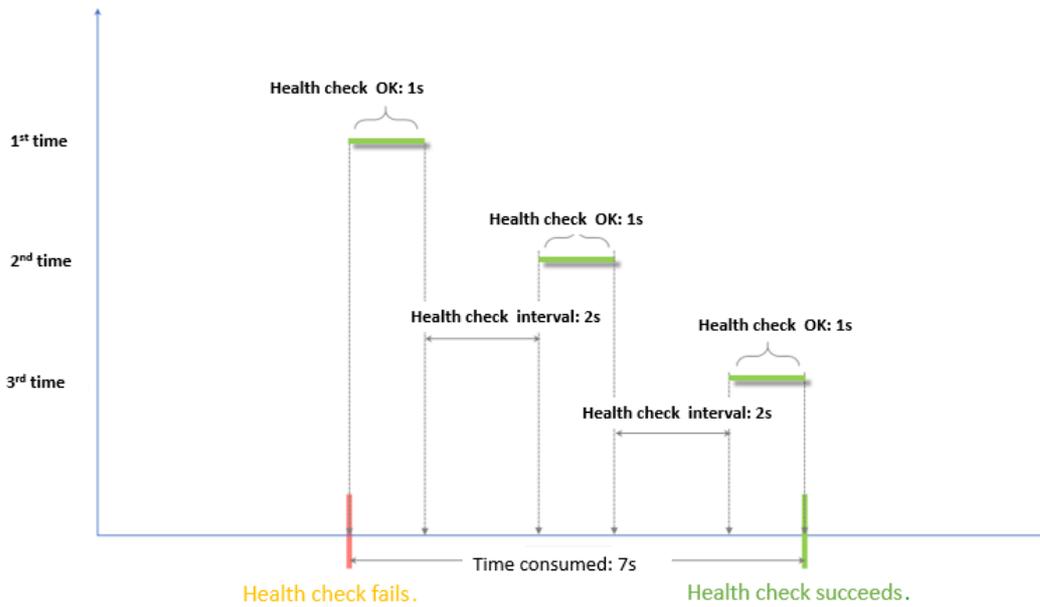
The health check time window is calculated as follows:

- Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1)



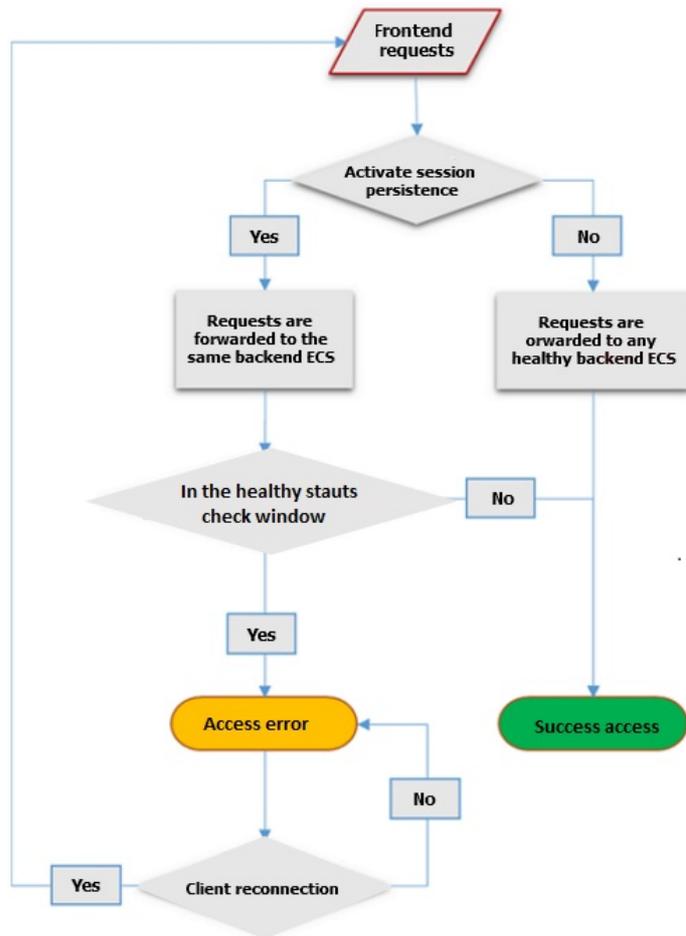
- Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1)

Note The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are used, the response time is almost negligible because only whether the specific port is alive is checked. For HTTP health checks, the response time depends on the performance and load of the application server and is typically within a few seconds.



The health check result has the following impact on request forwarding:

- If the health check of the target ECS instance fails, new requests are distributed to other ECS instances. This does not affect client access.
- If the health check of the target ECS instance succeeds, new requests are distributed to the instance and client access is normal.
- If an exception occurs on the target ECS instance and a request arrives during a time window for health check failures, the request is still sent to the ECS instance because the number of failed health checks has not reached the unhealthy threshold (3 by default). In the case, client access fails.



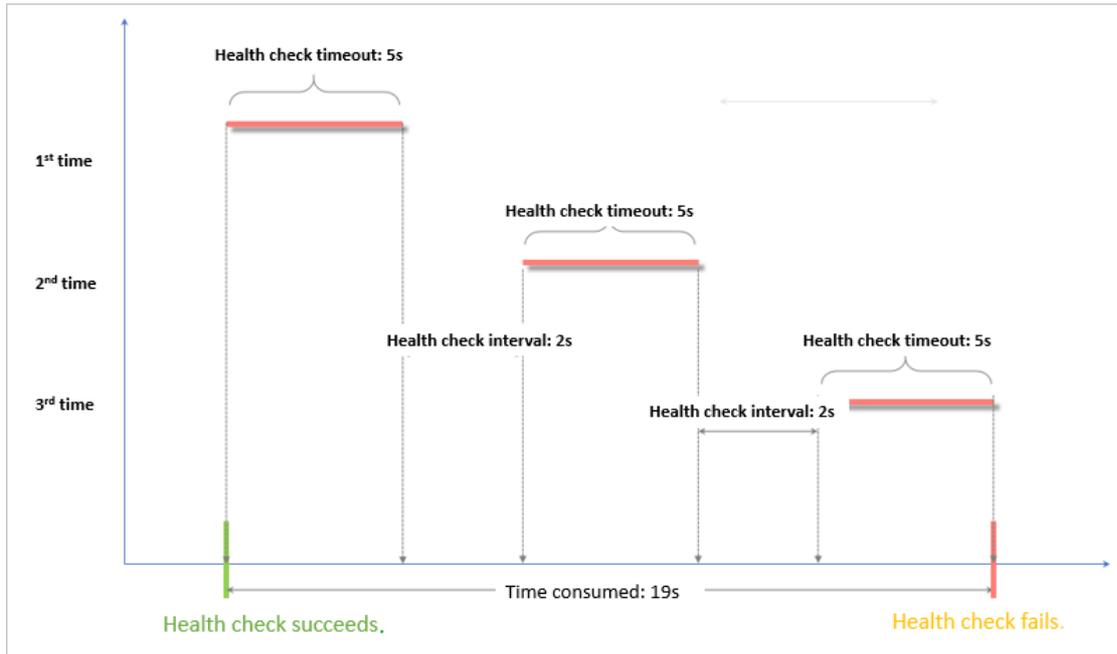
Examples of health check response timeout and health check interval

The examples use the following health check settings:

- Response timeout: 5s
- Health check interval: 2s
- Healthy threshold: 3
- Unhealthy threshold: 3

Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1). That is, $5 \times 3 + 2 \times (3-1) = 19s$.

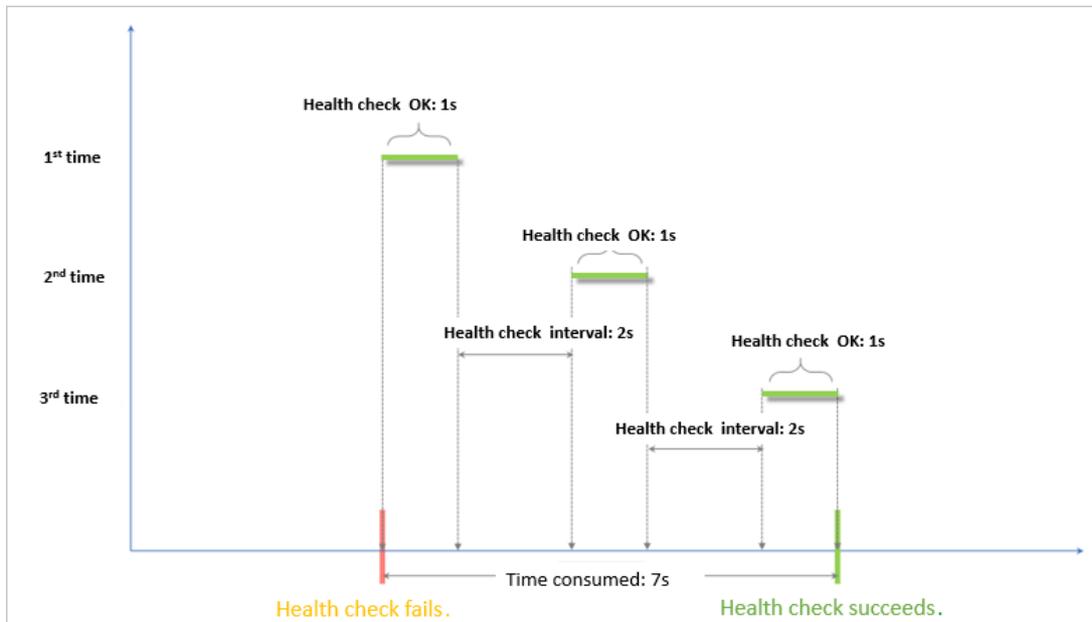
The following figure shows the process of a status change from healthy to unhealthy.



Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1). That is, $(1 \times 3) + 2 \times (3-1) = 7s$ (assuming that the response time of a successful health check is 1s).

Note The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are used, the response time is almost negligible because only whether the specific port is alive is checked. For HTTP health checks, the response time depends on the performance and load of the application server and is typically within a few seconds.

The following figure shows the process of a status change from unhealthy to healthy.



Domain name setting in HTTP health checks

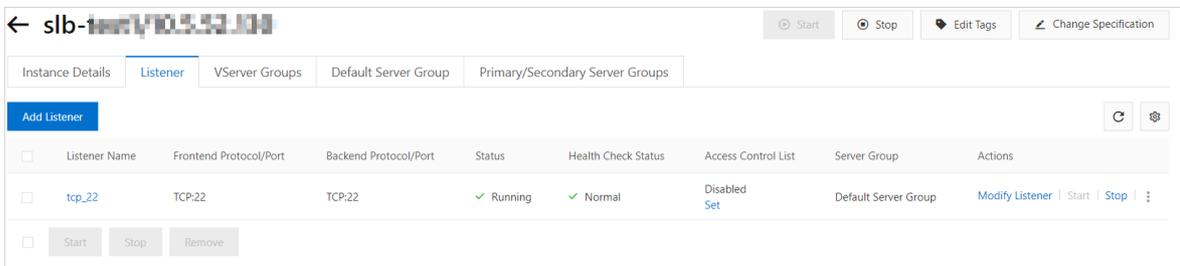
When HTTP health checks are used, you can set a domain name for health checks. The setting is optional. Some application servers verify the host field in requests. In this case, the request header must contain the host field. If a domain name is configured in the health check feature, SLB adds the domain name to the host field when forwarding a request to an application server. If no domain name is configured, the health check request will be denied by the application server because it does not contain a host field and the health check may fail. If your application server verifies the host field in requests, you must configure a domain name to make sure that the health check feature works.

21.7.2. Configure health check

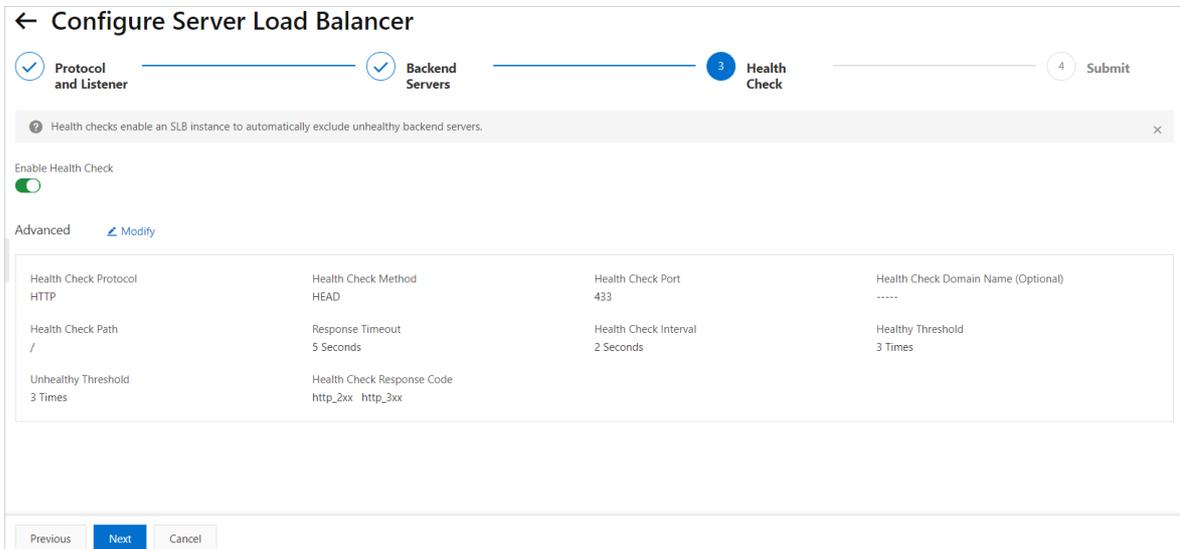
This topic describes how to configure health check. You can configure health check when you create a listener or for an existing listener. The default health check settings can meet your requirements in most cases.

Procedure

1. **Log on to the SLB console.**
2. Find the target SLB instance and click its instance ID.
3. Click the **Listener** tab.
4. Click **Add Listener**, or find the target listener and click **Modify Listener** in the **Actions** column.



5. Click **Next** to go to the **Health Check** step. Click **Modify** next to **Advanced**.



We recommend that you use the default settings when you configure the health check.

Health check configuration

Parameter	Description
-----------	-------------

Parameter	Description
Health Check Protocol	<p>Select the protocol for health check. For TCP listeners, both the TCP health check and HTTP health check are supported.</p> <ul style="list-style-type: none"> The TCP health check implements detection at the network layer by sending SYN packets to check if a port is open. The HTTP health check verifies the health of a backend server by sending HEAD/GET requests to simulate browser access.
Health Check Method (for the HTTP and HTTPS health checks only)	<p>The health check of Layer-7 (HTTP or HTTPS) listeners supports both the HEAD and GET methods. The HEAD method is used by default.</p> <p>If your backend applications do not support the HEAD method or if the HEAD method is disabled, the health check may fail. To resolve this issue, you can use the GET method instead.</p> <p>If the GET method is used and the response size exceeds 8 KB, the response is truncated. However, the health check result is not affected.</p>
Health Check Path and Health Check Domain Name (Optional) (for the HTTP health check only)	<p>By default, SLB performs the health check by sending HTTP HEAD requests to the default homepage of the application deployed on the ECS instance with the internal IP address of the ECS instance.</p> <p>If you need to use another web page other than the default homepage for the health check, you must specify a request path.</p> <p>Some application servers require the request header to contain the host field for verification. If a domain name is configured in health check settings, SLB adds this domain name to the host field when forwarding a health check request to one of the preceding application servers. If no domain name is configured, SLB does not include the host field in the request, which means that the request will be rejected by the application server and the health check may fail. Therefore, if your application server verifies the host field in requests, you must configure a domain name in health check settings to make sure that the health check can run properly.</p> <p>The URL path for the health check must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), underscores (_), forward slashes (/), periods (.), percent signs (%), question marks (?), number signs (#), ampersands (&), and equals signs (=).</p> <p>The health check domain name can contain letters, digits, periods (.), and hyphens (-). The internal IP addresses of backend servers are used as domain names for the health check by default.</p>
Normal Status Code (for the HTTP health check only)	<p>Select the HTTP status code that indicates a successful health check.</p> <p>Default values: http_2xx and http_3xx.</p>
Health Check Port	<p>Set the detection port used by health check to access backend servers.</p> <p>By default, the backend port configured for the listener is used.</p> <p>Valid values: 1 to 65535.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note If a VServer group or an active/standby server group is configured for the listener, and the ECS instances in the group use different ports, leave this parameter empty. SLB uses the backend ports of the ECS instances to perform the health check.</p> </div>

Parameter	Description
Response Timeout	Specify the amount of time to wait for a health check response. If the backend ECS instance does not send an expected response within the specified response timeout period, the health check fails. Valid values: 1 to 300. Unit: seconds. Default value for UDP listeners: 10. Default value for HTTP, HTTPS, and TCP listeners: 5.
Health Check Interval	Specify the interval between health checks. All nodes in the LVS cluster perform health checks independently and in parallel on backend ECS instances at the specified interval. The health check statistics of a single ECS instance cannot reflect the health check interval because the nodes perform health checks at different times. Valid values: 1 to 50. Unit: seconds. Default value for UDP listeners: 5. Default value for HTTP, HTTPS, and TCP listeners: 2.
Unhealthy Threshold	Specify the number of consecutive failed health checks that must occur before an ECS instance is declared unhealthy. Valid values: 2 to 10. Default value: 3.
Healthy Threshold	Specify the number of consecutive successful health checks that must occur before an ECS instance is declared healthy. Valid values: 2 to 10. Default value: 3.

6. Click **Next**.

21.7.3. Disable the health check feature

This topic describes how to disable the health check feature. If you disable the health check feature, requests may be distributed to unhealthy ECS instances and cause impacts on your business. We recommend that you enable the health check feature.

Context

 **Note** You can only disable the health check feature for HTTP and HTTPS listeners. The health check feature for UDP and TCP listeners cannot be disabled.

Procedure

1. [Log on to the SLB console](#).
2. On the **Server Load Balancer** page, find the target SLB instance and click its instance ID.
3. On the **Listeners** tab, find the target listener and click **Configure** in the **Actions** column.
4. On the **Configure Listener** page, click **Next** until the **Health Check** step appears.
5. Turn off **Enable Health Check**.
6. Click **Next**.
7. Click **Submit**, and then click **OK**.

21.8. Certificate management

21.8.1. Certificate overview

information, see [RFC 1421](#).

- The certificate must conform to the corresponding format requirements. Generally, the intermediate CA provides instructions about the certificate format when issuing the certificate. The certificate must conform to the format requirements.

A sample certificate chain is shown as follows.

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

Public keys of certificates

SLB supports the following public key algorithms:

- RSA 1024
- RSA 2048
- RSA 4096
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

RSA private keys

When you upload a server certificate, you must upload the private key of the certificate.

An RSA private key must meet the following format requirements:

- The private key must start with `-----BEGIN RSA PRIVATE KEY-----` and end with `-----END RSA PRIVATE KEY-----`, and these parts must also be uploaded.
- Blank lines are not allowed in the certificate content. Each line except the last line must contain 64 characters. The last line can contain 64 or fewer characters. For more information, see [RFC 1421](#).

You may use an encrypted private key. For example, the private key starts with `-----BEGIN PRIVATE KEY-----` and ends with `-----END PRIVATE KEY-----` or starts with `-----BEGIN ENCRYPTED PRIVATE KEY-----` and ends with `-----END ENCRYPTED PRIVATE KEY-----`. The private key may also contain `Proc-Type: 4,ENCRYPTED`. In this case, you must first run the following command to convert the private key:

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

The following figure shows a sample RSA private key.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAzVziSSSChH67bmT8mFykAxQ1tKCYukwBiWZwk0StFEbTWHy8K
tTHSFD1u9TL6qycrHEG7cjYD4DK+kVIHU/Of/pUWj9LnrE3W34DaVzQdKA00I3A
Xw9SgrqJMJcLva2khNKA1+tNPSCPJoo9DDrP7wx7c0x7LbMb0dfZ8858KI0LuzJ
/fD0XxyuWoqaTePZtk9Qnjn957ZEPHjtUpVZuhs3409DDM/tJ3T18aaNYWHRPBc0
jNcz0Z6XQGf1rZG/Ve520GY6rb5dUYpdCFxNSNM6xYg8a1L7UHDHHP14AYsatdG
z5TMPmE f8yZPUYudTLxgMVAovJr09Dq+5Dm3QIDAQABAoIBAG168Z/nnFyRHRFi
LaF6+Wen8ZvNqkm0hAMQwIjH1Vp1fL74//8Qyea/EvUtuJHy86T/2PZQoNVhxe35
cgQ93T424WGPcWUshSfxewfbAYGF3ur8W0xq0uU078AxaKHncmNG7dGyoLUowRu
S+YxLrnpVzh1YkuH8TT53udd6TeTWi77r8dkGi9KSAZ0pRa19B7t+CHKIzm6ybs/2
06W/zH24YAwwkTYLKGHjoiEys111ah1AJvICVgTc3+LzG2pIpM7I+K0nHC5eswM
t5x9h/OT/ujZsyX9P0PaAyE2bay0t080tGexd076Ssv0KVhFvWjLUnhF6WcqFCD
xqhhxkEcGyEA+PftNb6eyXl+/Y/U8NM2fg3+r5Cms0j9Bg+9+Yz5FghqHuOedU
ZXIHRJ9u6B1XE1arpijVs/WHmFhYSTm6DbdD7S1tLy0BY4cPTRhziFTKt8AkDXMK
605u0UjWsq0Z8hn1X141ox2cW9ZQa/Hc9udeyQotP4NsMJWgp8V7tC0CgYEAAwNf
0F+/jujt0HoyxCh45IAqk4U0o4+hBCQbWcXv5qCz4mRyTawzFEG8/AR3md2rhMzi
GnJ5fdfe7uY+JsQFX2Q5JjwTadLBW4led05a/uKRa04UzVgnYp2aJKxtuWffvVbU
+kf728ZJRA6azSLvGm48hu/GL6bgfU3FkSkw03ECgYBpYK7TT7JvvnAErMtJf2yZ
ICRkQaB3gPSe/LCgzy1nhtaF0UbnXGeuwlAZR0wrz7X3TZqHEDcYoJ7mK346of
QhGLTtyoehkYkAUtq038Y04EKH6S/TzMzB0frXiPKg9s8UKQzkU+GSE7ootli+a
R8Xzu335EwxI6BwNN1abpQKbgQC8TialClq1FteXQyGcNdcReLMncUHKIKcP/+xn
R3kVl06MZCFAdqirAjiQWapkh9Bxbp2eHCrB81MFAWLRQ5Lok79b/jVmTZMC3upd
EJ/iSWjZKPhw7hCFaerPhxyNTJ5idEiu9U8E0id8111giPgn0p3sE0HPI89qZX
aaIMEQK8gQDK2bsnZE9y0ZWhGTeu94vziKmfRskJMGH8pLaTiliwiRhRYWJysZ9
80IDxnrmiPa9bCtEpK80zq28dq7axpCs9CavQRcv08h5Hx0yy23m9hFRzFdeQ7z
NTKh193HHF1joNM81LHFyGRFEWrrroWSgFBudR6USRnR/6iQ11xZXw==
-----END RSA PRIVATE KEY-----
```

21.8.3. Upload a certificate

This topic describes how to create and upload a certificate. Before you create an HTTPS listener, you must upload the required server certificate and CA certificate to SLB. You do not need to configure certificates on backend servers after uploading the certificates to SLB.

Prerequisites

- A server certificate is purchased.
- A CA certificate and a client certificate are generated.

Context

Note that you can create up to 100 certificates per account.

Procedure

1. In the left-side navigation pane, click **Certificates**.
2. Click **Create Certificate**.
3. In the **Create Certificate** dialog box, set certificate parameters and then click **Create**.

Parameter	Description
Certificate Name	Enter a name for the certificate. The name must be 1 to 80 characters in length and can contain letters, digits, hyphens (-), forward slashes (/), periods (.), underscores (_), and asterisks (*).
Organization	Select the organization to which the certificate belongs.
Resource Group	Select the resource group to which the certificate belongs.
Certificate Type	Select Server Certificate . For HTTPS one-way authentication, only the server certificate and the private key are required.
Public Key Certificate	Copy and paste the contents of the server certificate into the field. Click Example to view the valid certificate format. For more information, see Certificate requirements .

Parameter	Description
Private Key	<p>Copy and paste the private key of the server certificate into the field.</p> <p>Click Example to view the valid certificate format. For more information, see Certificate requirements.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Notice A private key is required only when you upload a server certificate.</p> </div>
Region	Select one or more regions where the certificate will be deployed.

4. Click Create.

21.8.4. Generate a CA certificate

When you configure an HTTPS listener, you can use a self-signed CA certificate. This topic describes how to generate a CA certificate and use the CA certificate to sign a client certificate.

Generate a CA certificate by using Open SSL

1. Run the following commands to create a `ca` folder in the `/root` directory and then create four subfolders under the `ca` folder.

```
$ sudo mkdir ca
$ cd ca
$ sudo mkdir newcerts private conf server
```

- `newcerts` is used to store the digital certificate signed by the CA certificate.
 - `private` is used to store the private key of the CA certificate.
 - `conf` is used to store the configuration files used for simplifying parameters.
 - `server` is used to store the server certificate.
2. Create an `openssl.conf` file that contains the following information in the `conf` directory.

```
[ ca ]
default_ca = foo

[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_cr_l_days= 30
default_md = md5
unique_subject = no
policy = policy_any

[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName = optional
commonName = supplied
emailAddress = optional
```

3. Run the following command to generate a private key.

```
$ cd /root/ca
$ sudo openssl genrsa -out private/ca.key
```

The following figure is an example of the key generation.

```
root@iZbp1hfvivcqx1jwbp31iZ:~/ca/conf# cd /root/ca
root@iZbp1hfvivcqx1jwbp31iZ:~/ca# sudo openssl genrsa -out private/ca.key
Generating RSA private key, 2048 bit long modulus
.....++++
....+++
..+++
e is 65537 (0x10001)
```

4. Run the following command and input the required information according to the prompts. Press Enter to generate a *csr* file.

```
$ sudo openssl req -new -key private/ca.key -out private/ca.csr
```

 **Note** Common Name is the domain name of the SLB instance.

```

root@iZbp1hfvivcqx1jwap3liZ:~/ca# sudo openssl req -new -key private/ca.key -out private/ca.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:ZheJiang
Locality Name (eg, city) []:HangZhou
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Alibaba
Organizational Unit Name (eg, section) []:Test
Common Name (e.g. server FQDN or YOUR name) []:mydomain
Email Address []:a@alibaba.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@iZbp1hfvivcqx1jwap3liZ:~/ca#

```

5. Run the following command to generate a *crt* file:

```
$ sudo openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out private/ca.crt
```

6. Run the following command to set the start sequence number for the private key, which can be any four characters.

```
$ sudo echo FACE > serial
```

7. Run the following command to create a CA key library:

```
$ sudo touch index.txt
```

8. Run the following command to create a certificate revocation list for removing the client certificate:

```
$ sudo openssl ca -gencrl -out /root/ca/private/ca.crl -crldays 7 -config "/root/ca/conf/openssl.conf"
```

The output is:

```
Using configuration from /root/ca/conf/openssl.conf
```

Sign the client certificate

1. Run the following command to generate a *users* folder under the *ca* directory to store the client key.

```
$ sudo mkdir users
```

2. Run the following command to create a key for the client certificate:

```
$ sudo openssl genrsa -des3 -out /root/ca/users/client.key 1024
```

 **Note** Enter a pass phrase when creating the key. It is the password to protect the private key from unauthorized access. Enter the same password twice.

3. Run the following command to create a *csr* file for the client key.

```
$ sudo openssl req -new -key /root/ca/users/client.key -out /root/ca/users/client.csr
```

Enter the pass phrase set in the previous step and other required information when prompted.

Note A challenge password is the password of the client certificate. Note that it is not the password of the client key.

- Run the following command to sign the client key.

```
$ sudo openssl ca -in /root/ca/users/client.csr -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/users/client.crt -config "/root/ca/conf/openssl.conf"
```

Enter *y* twice when prompted to confirm the operation.

```
root@izbplhfivvcqxljwv31iZ:~/ca# sudo openssl ca -in /root/ca/users/client.csr
-cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/us
ers/client.crt -config "/root/ca/conf/openssl.conf"
Using configuration from /root/ca/conf/openssl.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :ASN.1 12:'ZheJiang'
localityName          :ASN.1 12:'HangZhou'
organizationName      :ASN.1 12:'Alibaba'
organizationalUnitName:ASN.1 12:'Test'
commonName            :ASN.1 12:'mydomain'
emailAddress          :IA5STRING:'a@alibaba.com'
Certificate is to be certified until Jun  4 15:28:55 2018 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
root@izbplhfivvcqxljwv31iZ:~/ca#
```

- Run the following command to convert the certificate to a *PKCS12* file.

```
$ sudo openssl pkcs12 -export -clcerts -in /root/ca/users/client.crt -inkey /root/ca/users/client.key -out /root/ca/users/client.p12
```

Follow the prompts to enter the pass phrase of client key. Then enter the password used for exporting the client certificate. This password is used to protect the client certificate, which is required when you install the client certificate.

- Run the following commands to view the generated client certificate:

```
cd users
ls
```

21.8.5. Convert the certificate format

Server Load Balancer (SLB) supports PEM certificates only. Certificates in other formats must be converted to the PEM format before they can be uploaded to SLB. We recommend that you use Open SSL for conversion.

Convert DER to PEM

DER: This format is usually used on a Java platform. The certificate file suffix is generally *.der*, *.cer*, or *.crt*.

- Run the following command to convert the certificate format:

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

- Run the following command to convert the private key:

```
openssl rsa -inform DER -outform PEM -in privatekey.der -out privatekey.pem
```

Convert P7B to PEM

P7B: This format is usually used in a Windows server and Tomcat.

Run the following command to convert the certificate format:

```
openssl pkcs7 -print_certs -in incertificate.p7b -out outcertificate.cer
```

Convert PFX to PEM

PFX: This format is usually used in a Windows server.

- Run the following command to extract the certificate:

```
openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
```

- Run the following command to extract the private key:

```
openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
```

21.8.6. Replace a certificate

This topic describes how to replace a certificate with a new certificate. We recommend that you replace certificates before they expire to avoid impacts on your service.

Procedure

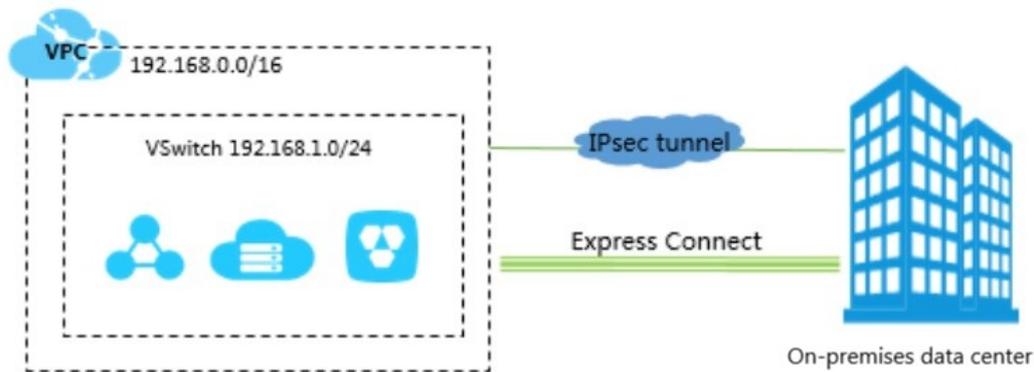
1. Create and upload a new certificate. For more information, see [Certificate overview](#).
2. Configure the certificate for the target HTTPS listener. For more information, see [Add an HTTPS listener](#).
3. On the **Certificates** page, find the certificate to be replaced and click **Delete** in the Actions column.
4. In the dialog box that appears, click **OK**.

22.Virtual Private Cloud (VPC)

22.1. What is a VPC?

A virtual private cloud (VPC) is a private network dedicated for your use. You have full control over your VPC, which you can define and customize by specifying the Classless Inter-domain Routing (CIDR) block, configuring route tables, and creating gateways. You can launch Apsara Stack resources such as Elastic Compute Service (ECS) instances, ApsaraDB for RDS (RDS) instances, and Server Load Balancer (SLB) instances in your VPC.

Furthermore, you can connect your VPC to other VPCs or on-premises networks to create a custom network environment. In this way, you can smoothly migrate applications and extend on-premises data centers to the cloud.



Components

Each VPC consists of one VRouter, at least one private CIDR block, and one or more VSwitches.

- Private CIDR block

When you create a VPC or a VSwitch, you must specify its private IP address range in the form of a CIDR block.

You can use the standard private CIDR blocks listed in the following table and their subsets as CIDR blocks for your VPCs. For more information, see the Plan and design a VPC section in this *User Guide*.

CIDR block	Number of available private IP addresses (excluding those reserved by the system)
192.168.0.0/16	65,532
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- VRouter

A VRouter is a hub that connects all VSwitches in a VPC and serves as a gateway between the VPC and other networks. After a VPC is created, a VRouter is automatically created for the VPC. Each VRouter is associated with a route table.

For more information, see the Route table overview section in this *User Guide*.

- VSwitch

A VSwitch is a basic network component that connects different cloud resources in a VPC. After you create a VPC, you can create VSwitches to partition your VPC into multiple subnets. VSwitches within a VPC can communicate with each other over the private network. You can deploy your applications in VSwitches that belong to different zones to improve service availability.

For more information, see the Create a VSwitch section in this *User Guide*.

22.2. Log on to the VPC console

This topic provides an example of how to log on to the Virtual Private Cloud console by using Google Chrome.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `https://[IP address or domain name of the ASCM console]`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password for logging on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username as prompted. Due to security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two of the following character types: uppercase letters, lowercase letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.

22.3. Quick start

22.3.1. Plan and design a VPC

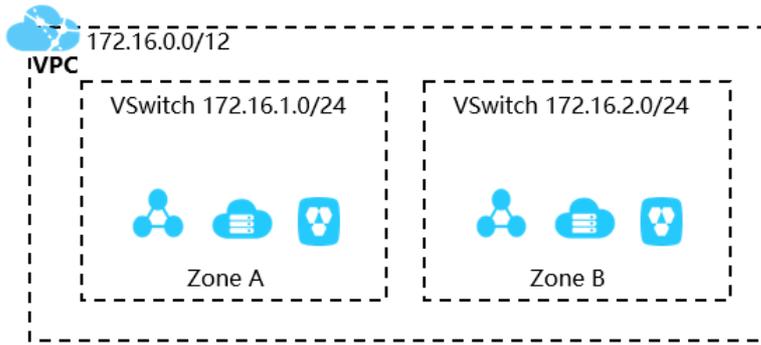
Before you create virtual private clouds (VPCs) and VSwitches, you need to plan the quantity and Classless Inter-domain Routing (CIDR) blocks of VPCs and VSwitches.

- [How many VPCs are required?](#)
- [How many VSwitches are required?](#)
- [How do I specify CIDR blocks?](#)
- [How do I specify CIDR blocks if I want to connect a VPC to other VPCs or on-premises data centers?](#)

How many VPCs are required?

- One VPC

We recommend that you create one VPC if you do not need to deploy systems in multiple regions or separate VPCs.

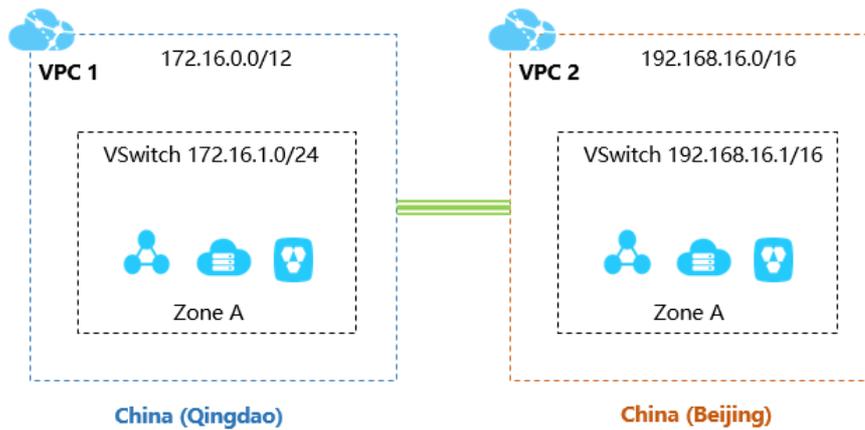


• Multiple VPCs

We recommend that you create multiple VPCs if you need to:

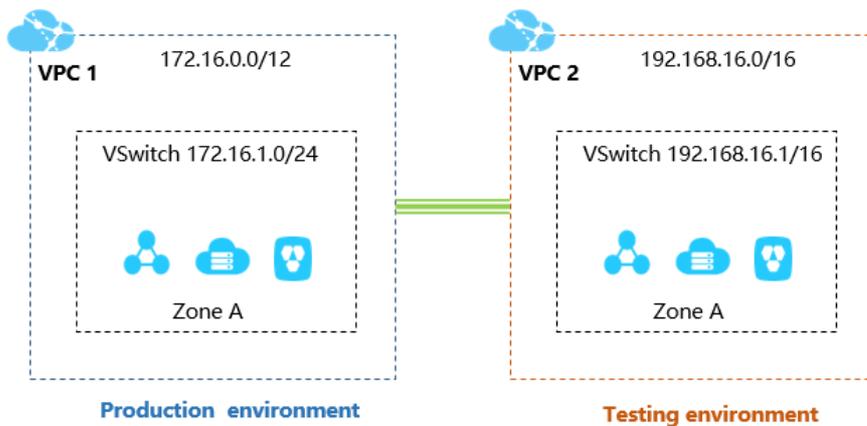
- Deploy application systems across regions.

A VPC cannot be deployed across regions. If you want to deploy your application systems in different regions, you must create multiple VPCs. You can use Express Connect and VPN Gateway to connect VPCs.



- Separate IT systems

To separate IT systems, you must create multiple VPCs. The following figure shows an example of isolating a production environment from a test environment by deploying them in separate VPCs.



How many VSwitches are required?

We recommend that you create at least two VSwitches for each VPC and deploy these VSwitches in different zones to achieve zone-disaster recovery.

After you deploy your applications in different zones within a region, you must measure the network latency between these applications. This is because the cross-zone network latency may be higher than expected due to complex data processing or cross-zone calls. An ideal approach is to optimize and adjust your systems to strike a balance between availability and latency.

In addition, the sizes and designs of your IT systems must also be taken into consideration when you create VSwitches. If you allow traffic from the Internet to be routed to and from the frontend systems, you can deploy the front-end systems in different VSwitches and the backend systems in other VSwitches to create a robust disaster recovery strategy.

How do I specify CIDR blocks?

When you create VPCs and VSwitches, you must specify their private IP address ranges in the form of CIDR blocks.

- VPC CIDR blocks

You can use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or their subsets as the CIDR blocks of your VPCs. To specify CIDR blocks for VPCs, follow these rules:

- If you have only one VPC and this VPC does not need to communicate with any on-premises data center, you can use one of the preceding CIDR blocks or one of their subsets as the CIDR block of the VPC.
- If you have multiple VPCs, or you need to build a hybrid cloud to integrate VPCs and on-premises data centers, we recommend that you use the subsets of the preceding CIDR blocks for your VPCs. In this case, the mask cannot be longer than 16 bits.

- VSwitch CIDR blocks

The CIDR block of a VSwitch must be a subset of the CIDR block of the VPC this VSwitch resides in. For example, if the CIDR block of a VPC is 192.168.0.0/16, the CIDR block of a VSwitch in the VPC must be a segment from 192.168.0.0/17 to 192.168.0.0/29.

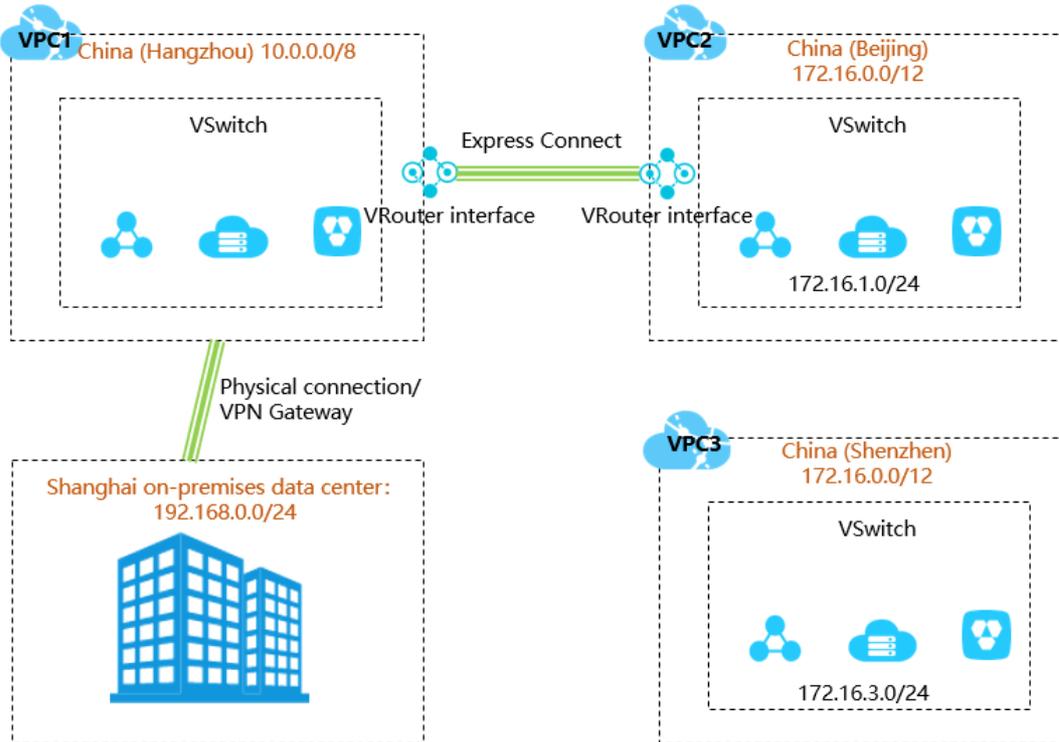
To specify CIDR blocks for VSwitches, follow these rules:

- The CIDR block size for a VSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. This range is set because a 16-bit host address space provides addressing for 65,534 ECS instances, which can meet your needs in most cases, while a mask smaller than 29 bits can only allow very few usable host addresses.
- The first and the last three IP addresses in each VSwitch CIDR block are reserved by the system. For example, if the CIDR block of a VSwitch is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved.
- You must check the number of ECS instances in the VSwitch before you specify the CIDR block of a VSwitch.

How do I specify CIDR blocks if I want to connect a VPC to other VPCs or on-premises data centers?

Before you connect your VPC to another VPC or an on-premises data center, you must make sure that the CIDR block of your VPC does not conflict with that of the peer network.

For example, assume you have three VPCs: VPC1 in China (Hangzhou), VPC2 in China (Beijing), and VPC3 in China (Shenzhen), as shown in the following figure. Express Connect circuit is used for VPC1 and VPC2 to communicate with each other. VPC3 does not communicate with other VPCs, but may need to communicate with VPC2 in the future. Additionally, you have an on-premises data center in Shanghai, and you need to connect it to VPC1 by using an Express Connect circuit.



In this example, the CIDR block of VPC2 is different from the CIDR block of VPC1, but is the same with the CIDR block of VPC3. However, considering that VPC2 and VPC3 may need to communicate with each other later in the private network, the VSwitches in these VPCs are assigned with different CIDR blocks. This example demonstrates that VPCs communicating with each other can have identical CIDR blocks, but their VSwitches must have different CIDR blocks.

When you specify CIDR blocks for multiple VPCs that need to communicate with each other, follow these rules:

- The preferred practice is to specify different CIDR blocks for different VPCs. You can use the subsets of the standard CIDR blocks to increase the number of available CIDR blocks.
- If you cannot assign different CIDR blocks for VPCs, try to specify different CIDR blocks for the VSwitches in these VPCs.
- If you cannot assign different CIDR blocks for all VSwitches in these VPCs, make sure that different CIDR blocks are configured for the VSwitches communicating with each other.

22.3.2. Create an IPv4 VPC

This topic describes how to build a virtual private cloud (VPC) with an IPv4 Classless Inter-domain Routing (CIDR) block and create an Elastic Compute Service (ECS) instance in the VPC.

Prerequisites

Before you create a VPC, you must first plan and design the VPC. For more information, see [Set up network connections](#).

Step 1: Create a VPC

Perform the following steps to create a VPC:

1. Log on to the VPC console.
2. On the VPCs page, click **Create VPC**.
3. On the **Create VPC** page, configure the VPC and click **Submit**. The following table describes the parameters for creating a VPC.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, the administrators of sub-organizations can create resources in the VPC. In this tutorial, select No.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> . In this tutorial, enter <code>VPCtest</code> .
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting methods are supported: <ul style="list-style-type: none"> ◦ Recommended CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8. ◦ Custom CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/16. In this tutorial, select Recommended CIDR Block and then select 192.168.0.0/16. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note After you create a VPC, you cannot change its IPv4 CIDR block. </div>
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block to the VPC. <ul style="list-style-type: none"> ◦ Do Not Assign: No IPv6 CIDR block will be assigned to the VPC. ◦ Assign: An IPv6 CIDR block will be automatically assigned to the VPC. In this tutorial, select Do Not Assign .
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .

Step 2: Create a VSwitch

Perform the following steps to create a VSwitch in a VPC:

1. In the left-side navigation pane, click **VSwitches**.
2. On the **VSwitches** page, click **Create VSwitch**.
3. On the **VSwitch** page, configure the VSwitch and click **Submit**. The following table describes the parameters for creating a VSwitch.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.
Zone	<p>Select a zone to deploy the VSwitch.</p> <p>Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches that reside in different zones to achieve zone-disaster recovery.</p> <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note Each cloud resource can be deployed in only one VSwitch.</p> </div>
VPC	<p>Select the VPC for which you want to create the VSwitch.</p> <p>In this tutorial, select VPCtest.</p>
Dedicated for Off-Cloud Servers	<p>Specify whether the VSwitch is dedicated for off-cloud servers.</p> <p>For more information, see the Features of off-cloud servers for VPC section in the <i>ECS Bare Metal Instance User Guide</i>.</p> <p>In this tutorial, select No.</p>
VSwitch Name	<p>Enter a name for the VSwitch.</p> <p>The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>
IPv4 CIDR Block	<p>Enter an IPv4 CIDR block for the VSwitch.</p> <p>This tutorial uses the default IPv4 CIDR block.</p>
IPv6 CIDR Block	<p>Specify whether to assign an IPv6 CIDR block to the VSwitch.</p> <p>In this tutorial, select Do Not Assign.</p>
Description	<p>Enter a description for the VSwitch.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>

Step 3: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, configure the security group and click **Submit**. The following table describes the parameters for creating a security group.

Parameter	Description
Organization	Select the organization to which the security group belongs.
Resource Set	Select the resource set to which the security group belongs.
Region	Select the region that will use the security group. The security group and the VPC must belong to the same region.
Zone	Select the zone that will use the security group.
VPC	Select the VPC to which the security group belongs.
Security Group Name	Enter a name for the security group. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
Description	Enter a description for the security group. The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .

Step 4: Create an ECS instance

Perform the following steps to create an ECS instance in the specified VPC:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **VSwitches**.
3. In the top navigation bar, select the region of the target VSwitch.
4. On the **VSwitches** page, find the target VSwitch and choose **Purchase > ECS Instance** in the **Actions** column.
5. On the **Create ECS Instance** page, configure the ECS instance and click **Submit**. For more information about how to configure an ECS instance, see **Create an instance** under **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.

22.3.3. Create an IPv6 VPC

This topic describes how to build a virtual private cloud (VPC) with an IPv6 Classless Inter-domain Routing (CIDR) block and create an Elastic Compute Service (ECS) instance assigned with an IPv6 address in the VPC.

Step 1: Create a VPC and a VSwitch

Before you deploy cloud resources in a VPC, you must create a VPC and a VSwitch.

Perform the following steps to create a VPC and a VSwitch:

1. Log on to the VPC console.
2. On the **VPCs** page, click **Create VPC**.
3. On the **Create VPC** page, configure the VPC and click **Submit**. The following table describes the parameters for creating a VPC.

Parameter	Description
Organization	Select the organization to which the VPC belongs.

Parameter	Description
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, the administrators of sub-organizations can create resources in the VPC. In this tutorial, select No.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> . In this tutorial, enter <code>VPCtest</code> .
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting methods are supported: <ul style="list-style-type: none"> ◦ Recommended CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8. ◦ Custom CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/16. In this tutorial, select Recommended CIDR Block and then select 192.168.0.0/16. <div style="background-color: #e0f2f7; padding: 5px; margin-top: 10px;"> ? Note After you create a VPC, you cannot change its IPv4 CIDR block. </div>
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block to the VPC. <ul style="list-style-type: none"> ◦ Do Not Assign: No IPv6 CIDR block will be assigned to the VPC. ◦ Assign: An IPv6 CIDR block will be automatically assigned to the VPC. In this tutorial, select Assign.
Description	Enter a description for the VPC. The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .

4. Click **Back to Console**. In the left navigation pane, click **VSwitches**.
5. On the **VSwitches** page, click **Create VSwitch**.
6. On the **VSwitch** page, configure the VSwitch and click **Submit**. The following table describes the parameters for creating a VSwitch.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.

Parameter	Description
Zone	<p>Select a zone to deploy the VSwitch.</p> <p>Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches that reside in different zones to achieve zone-disaster recovery.</p> <p> Note Each cloud resource can be deployed in only one VSwitch.</p>
VPC	<p>Select the VPC for which you want to create the VSwitch.</p> <p>In this tutorial, select VPCtest.</p>
Dedicated for Off-Cloud Servers	<p>Specify whether the VSwitch is dedicated for off-cloud servers.</p> <p>For more information, see the Features of off-cloud servers for VPC section in the <i>ECS Bare Metal Instance User Guide</i>.</p> <p>In this tutorial, select No.</p>
VSwitch Name	<p>Enter a name for the VSwitch.</p> <p>The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>
IPv4 CIDR Block	<p>Enter an IPv4 CIDR block for the VSwitch.</p> <p>This tutorial uses the default IPv4 CIDR block.</p>
IPv6 CIDR Block	<p>Enter an IPv6 CIDR block for the VSwitch.</p> <p>This tutorial uses the default IPv6 CIDR block.</p>
Description	<p>Enter a description for the VSwitch.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>

Step 2: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, configure the security group and click **Submit**. The following table describes the parameters for creating a security group.

Parameter	Description
Organization	Select the organization to which the security group belongs.
Resource Set	Select the resource set to which the security group belongs.

Parameter	Description
Region	Select the region that will use the security group. The security group and the VPC must belong to the same region.
Zone	Select the zone that will use the security group.
VPC	Select the VPC to which the security group belongs.
Security Group Name	Enter a name for the security group. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
Description	Enter a description for the security group. The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .

Step 3: Create and configure an ECS instance

After you create a VPC and a VSwitch, you must create an ECS instance and assign an IPv6 address to the ECS instance, and then associate this IPv6 address with the network interface controller (NIC) of the ECS instance.

Perform the following steps to create and configure an ECS instance:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where your VSwitch resides.
4. On the **VSwitches** page, find the target VSwitch and choose **Purchase > ECS Instance** in the **Actions** column.
5. On the **Create ECS Instance** page, configure the ECS instance and click **Submit**. In this tutorial, select **Assign** to assign an IPv6 address to the ECS instance. For more information about other parameters, see **Create an instance** under **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.
6. Return to the **Instances** page, and click the instance ID to view the assigned IPv6 address.
7. Configure a static IPv6 address.
 - If the image of your ECS instance supports DHCPv6, you do not need to manually configure the static IPv6 address. DHCPv6 enables automatic configuration of IPv6 addresses. Therefore, if your ECS instance image supports DHCPv6, the ECS instance can use the assigned IPv6 address to communicate in the private network directly after being created.

The following images support DHCPv6:

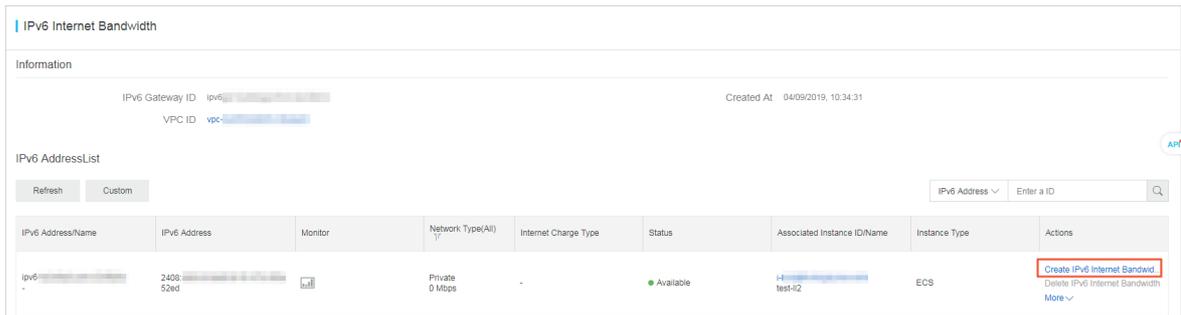
- Linux images:
 - CentOS 7.6 IPV6 64Bit
 - CentOS 6.10 64Bit
 - SUSE Linux Enterprise Server 12 SP4 64Bit
- Windows Server images
- If the image of your ECS instance does not support DHCPv6, you must manually configure an IPv6 address for the ECS instance. We recommend that you refer to the related documentation for each image for configuration guidance.

Step 4: Purchase an IPv6 Internet bandwidth plan

By default, IPv6 addresses are only used for communication inside private networks. If you want to allow an instance assigned with an IPv6 address to access the Internet or be accessed by IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address.

Perform the following steps to purchase an Internet bandwidth plan for the IPv6 address:

1. In the top navigation bar, choose **Products > Networking > IPv6 Gateway**.
2. Select the region where the IPv6 Gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 Gateway and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the **IPv6 Internet Bandwidth** page, find the target IPv6 address and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Select a bandwidth plan and then click **Submit**. The maximum IPv6 Internet bandwidth for an IPv6 Gateway of the Free, Enterprise, or Enhanced Edition is 2 Gbit/s.

Step 5: Configure the security group rules

IPv4 and IPv6 addresses are independent of each other. If the current security group rules cannot serve your IPv6 services, you must configure security group rules for IPv6 traffic.

For more information, see *Add security group rules* under *Security groups* in the *Apsara Stack Elastic Compute Service User Guide*.

Step 6: Test the network connectivity

Log on to the ECS instance and run the ping command to test the network connectivity.

```
[root@iZhp3aehva ~]# ping6 ipv6.baidu.com
PING ipv6.baidu.com(2400:da00:2::29 (2400:da00:2::29)) 56 data bytes
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=1 ttl=45 time=77.1 ms
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=2 ttl=45 time=77.1 ms
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=3 ttl=45 time=77.0 ms
^C
--- ipv6.baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 77.070/77.101/77.127/0.227 ms
[root@iZhp3ae ~]#
```

22.4. VPCs and VSwitches

22.4.1. Overview

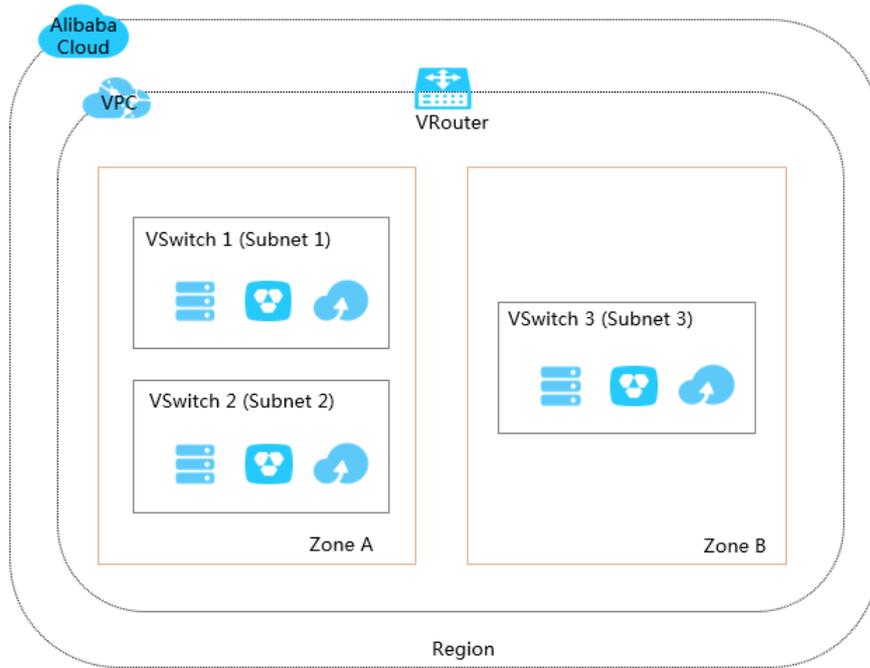
To get started with Virtual Private Cloud, you need to create at least one virtual private cloud (VPC) and one VSwitch. You can create VSwitches in a VPC to partition a VPC into multiple subnets. By default, subnets (VSwitches) within a VPC can communicate with each other over the private network.

VPCs and VSwitches

A VPC is a virtual network dedicated for your use. You can deploy cloud resources in VPCs that you define.

Note A cloud resource cannot be directly deployed in a VPC, but can be deployed in a VSwitch of the VPC.

A VSwitch is a basic network component in a VPC and is used to connect cloud resources. Each VPC must reside entirely within one region and cannot span multiple regions. However, a VPC spans all of the zones in a region, which means you can create one or more VSwitches in each zone to partition a VPC into subnets.



CIDR blocks and IP addresses

VPCs support both IPv4 and IPv6 addressing protocols. By default, VPCs use the IPv4 addressing protocol. You can enable the IPv6 addressing protocol as needed.

VPCs can operate in a dual-stack mode, which allows your resources to communicate over IPv4, or IPv6, or both. IPv4 and IPv6 addresses are independent of each other. Therefore, you must configure routing and security groups in your VPC separately for IPv4 and IPv6.

The following table summarizes the differences between IPv4 and IPv6 in Apsara Stack VPC.

IPv4 VPC	IPv6 VPC
The format is 32-bit, 4 groups of up to 3 decimal digits.	The format is 128-bit, 8 groups of 4 hexadecimal digits.
The IPv4 addressing protocol is enabled for all VPCs by default.	The IPv6 addressing protocol is optional for a VPC.
The VPC Classless Inter-domain Routing (CIDR) block size can be from /8 to /24.	The VPC CIDR block size is fixed at /61.
The VSwitch CIDR block size can be from /16 to /29.	The VSwitch CIDR block size is fixed at /64.
You can choose the private IPv4 CIDR block for your VPC.	Apsara Stack automatically assigns an IPv6 CIDR block for your VPC from its IPv6 address pool. You cannot select your own range.

IPv4 VPC	IPv6 VPC
Supported on all instance types.	Not supported on certain instance types. For more information, see <i>Instance types</i> under <i>What is ECS</i> in the <i>Apsara Stack Elastic Compute Service User Guide</i> .
Elastic IPv4 addresses are supported.	Elastic IPv6 addresses are not supported.
VPN gateways and NAT gateways are supported.	VPN gateways and NAT gateways are supported.

By default, the IPv4 and IPv6 addresses provided for VPCs can only be used for communication within the private network. Resources in different VSwitches within a VPC communicate with one another over private network connections. To connect a VPC to another VPC or an on-premises data center, you need to configure Express Connect or VPN Gateway. .

To enable cloud resources in a VPC to access the Internet, set the following configurations:

- IPv4 communication

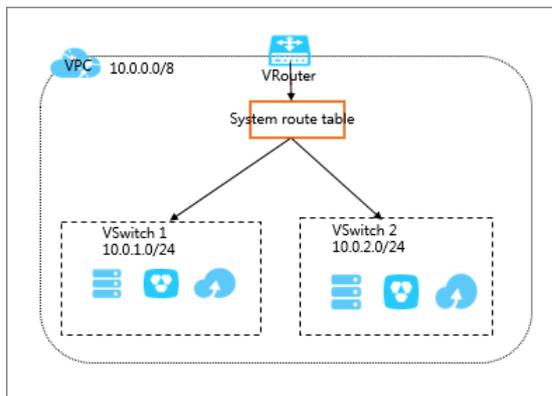
You can configure a NAT gateway or associate elastic IP addresses (EIPs) to the Elastic Compute Service (ECS) instances in a VPC to allow these ECS instances to access the Internet by using IPv4 addresses.

- IPv6 communication

To enable cloud resources in a VPC to access the Internet by using an IPv6 address, you must purchase public bandwidth for the IPv6 address. You can also configure an egress-only rule for an IPv6 address to allow outbound communication over IPv6 from instances in your VPC to the Internet, and prevent clients on the Internet from initiating IPv6 connections with your instances.

Routing

After a VPC is created, the system automatically creates a system route table and adds system routes to the route table for traffic management. Each VPC has only one system route table, which is generated automatically upon the creation of the VPC. You cannot create or delete system route tables.



If one destination address matches more than one route entry in a route table, the system selects an entry by implementing the longest prefix match algorithm, whereby the most specific of the matching entries, the one with the longest subnet mask, is used to route traffic. You can add a custom route entry to route traffic destined for a specific destination. For more information, see [Add a custom route entry](#).

22.4.2. VPC management

22.4.2.1. Create a VPC

A virtual private cloud (VPC) is a private network dedicated for your use. You have full control over your VPC. For example, you can specify Classless Inter-domain Routing (CIDR) blocks, configure route tables, and set network gateways for your VPC. You can deploy Apsara Stack resources in your VPC, such as Elastic Compute Service (ECS) instances, ApsaraDB for RDS (RDS) instances, and Server Load Balancer (SLB) instances.

Procedure

1. **Log on to the VPC console.**
2. In the top navigation bar, select a region to deploy your VPC.

 **Note** The VPC must be in the same region as the cloud resources that you want to deploy in this VPC.

3. On the VPC page, click **Create VPC**.
4. On the **Create VPC** page, configure the VPC and click **Submit**. The following table describes the parameters for creating a VPC.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, the administrators of sub-organizations can create resources in the VPC.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting methods are supported: <ul style="list-style-type: none"> ◦ Recommended CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8. ◦ Custom CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/16. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2; margin-top: 10px;"> <p> Note After you create a VPC, you cannot change its IPv4 CIDR block.</p> </div>
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block to the VPC. <ul style="list-style-type: none"> ◦ Do Not Assign: No IPv6 CIDR block will be assigned to the VPC. ◦ Assign: An IPv6 CIDR block will be automatically assigned to the VPC. <p>If you set this parameter to Assign, the system automatically creates a free IPv6 gateway for this VPC, and assigns an IPv6 CIDR block with the subnet mask /56, such as 2xx1: db8::/56. By default, IPv6 addresses are only used for communication inside private networks. If you want to allow an instance assigned with an IPv6 address to access the Internet or be accessed by IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address. For more information, see <i>Enable IPv6 Internet bandwidth</i> under <i>Manage IPv6 Internet bandwidth</i> in the <i>Apsara Stack Elastic Compute Service User Guide</i>.</p>

Parameter	Description
Description	<p>Enter a description for the VPC.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>

22.4.2.2. Modify the name and description of a VPC

This topic describes how to modify the name and description of a virtual private cloud (VPC).

Procedure

1. [Log on to the VPC console.](#)
2. In the top navigation bar, select the region where your VPC is deployed.
3. On the VPCs page, find the target VPC network and click **Manage** in the **Actions** column.
4. In the **VPC Details** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the VPC and click **OK**. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
5. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description, and click **OK**. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

22.4.2.3. Delete a VPC

This topic describes how to delete a virtual private cloud (VPC). After you delete a VPC, the VRouter and route tables associated with this VPC are also deleted.

Prerequisites

Before you delete a VPC, make sure that the following requirements are met:

- No VSwitch exists in the VPC. If the VPC has one or more VSwitches, we recommend that you delete the VSwitches first before deleting the VPC. For more information, see [Delete a VSwitch](#).
- No IPv6 gateway is associated with the VPC. If the VPC is associated with an IPv6 gateway, we recommend that you delete the IPv6 gateway first before deleting the VPC.

Procedure

1. [Log on to the VPC console.](#)
2. In the top navigation bar, select the region where your VPC is deployed.
3. On the VPCs page, find the target VPC and click **Delete** in the **Actions** column.
4. In the **Delete VPC** dialog box, click **OK**.

22.4.3. VSwitch management

22.4.3.1. Create a VSwitch

A VSwitch is a basic network component in a virtual private cloud (VPC) and is used to connect cloud resources.

Context

After you create a VPC, you can create VSwitches to partition your VPC into multiple subnets. VSwitches within a VPC can communicate with each other over the private network. Cloud resources are deployed in VSwitches. You can deploy your applications in VSwitches that belong to different zones to improve service availability.

 **Note** VSwitches do not support multicasting or broadcasting.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click VSwitches.
3. Select the region of the VPC in which you want to create a VSwitch.
4. On the VSwitches page, click Create VSwitch.
5. On the VSwitch page, configure the VSwitch and click Submit. The following table describes the parameters for creating a VSwitch.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.
Zone	<p>Select a zone to deploy the VSwitch.</p> <p>Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches that reside in different zones to achieve zone-disaster recovery.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Each cloud resource can be deployed in only one VSwitch.</p> </div>
VPC	Select the VPC for which you want to create the VSwitch.
Dedicated for Off-Cloud Servers	<p>Specify whether the VSwitch is dedicated for off-cloud servers.</p> <p>For more information, see the Features of off-cloud servers for VPC section in the <i>ECS Bare Metal Instance User Guide</i>.</p>
VSwitch Name	<p>Enter a name for the VSwitch.</p> <p>The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (<code>_</code>), hyphens (<code>-</code>), periods (<code>.</code>), colons (<code>:</code>), and commas (<code>,</code>). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>
IPv4 CIDR Block	<p>Enter an IPv4 CIDR block for the VSwitch.</p> <ul style="list-style-type: none"> ○ You must specify the IP address range for the VSwitch in the form of a CIDR block. The CIDR block size for a VSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. ○ The CIDR block of a VSwitch must be a subset of the CIDR block of the VPC this VSwitch resides in. ○ The first and the last three IP addresses in each VSwitch CIDR block are reserved by the system. For example, if the CIDR block of a VSwitch is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved. ○ The CIDR block of a VSwitch must be more specific than the CIDR range of a route in any of the VPC route tables. ○ After you create a VSwitch, you cannot modify its CIDR block.

Parameter	Description
IPv6 CIDR Block	<p>Enter an IPv6 CIDR block for the VSwitch.</p> <ul style="list-style-type: none"> You must check whether IPv6 is enabled for the specified VPC. If not, you cannot assign an IPv6 CIDR block to the VSwitch. If yes, you can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block of the VSwitch. <p>For example, if the IPv6 CIDR block of the VPC that contains the VSwitch is 2xx1:db8::/64, you can enter 255 (FF in the hexadecimal system) in this field to define the IPv6 CIDR block of the VSwitch as 2xx1:db8:ff::/64.</p>
Description	<p>Enter a description for the VSwitch.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>

22.4.3.2. Create cloud resources in a VSwitch

This topic describes how to create cloud resources in a VSwitch. Note that cloud resources are not directly deployed in a virtual private cloud (VPC), but are provisioned into VSwitches (subnets) of a VPC. You can create cloud resources in a VSwitch.

Procedure

1. [Log on to the VPC console.](#)
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region of the VPC to which the VSwitch belongs.
4. On the **VSwitches** page, find the target VSwitch, click **Purchase** in the **Actions** column, and select the cloud resource you want to create. You can create Elastic Compute Service (ECS) instances, ApsaraDB for RDS (RDS) instances, and Server Load Balancer (SLB) instances in a VSwitch.
5. On the page that appears, set the parameters to create a cloud resource.

22.4.3.3. Modify the name and description of a VSwitch

This topic describes how to modify the name and description of a VSwitch.

Procedure

1. [Log on to the VPC console.](#)
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region of the VPC to which the VSwitch belongs.
4. On the **VSwitches** page, find the target VSwitch and click **Manage** in the **Actions** column.
5. In the **VSwitch Basic Information** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the VSwitch and click **OK**. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.
6. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description for the VSwitch and click **OK**. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

22.4.3.4. Delete a VSwitch

This topic describes how to delete a VSwitch that you no longer need. Cloud resources cannot be deployed in deleted VSwitches.

Prerequisites

Before you delete a VSwitch, make sure that the following conditions are met:

- You have deleted all of your cloud resources created under the VSwitch, such as Elastic Compute Service (ECS) instances, Server Load Balancer (SLB) instances, and ApsaraDB for RDS (RDS) instances.
- You have deleted or disabled all resources and actions associated with this VSwitch, such as high-availability virtual IP addresses (HAVIPs) and source network address translation (SNAT) entries.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region of the VPC to which the VSwitch belongs.
4. On the **VSwitches** page, find the target VSwitch and click **Delete** in the **Actions** column.
5. In the **Delete VSwitch** dialog box that appears, click **OK**.

22.5. Route tables

22.5.1. Overview

Routes and route tables

Each virtual private cloud (VPC) comes with a default route table pre-configured with system route entries that direct traffic flowing in and out of the VPC. You cannot create or delete system route entries. However, you can create custom route entries to route traffic destined for specific Classless Inter-domain Routing (CIDR) blocks to the destinations that you define.

Route tables

When you create a VPC, it automatically has a system route table that controls the routing for all VSwitches (subnets) within the VPC by default. You cannot create or delete the system route table of a VPC.

Each *route entry* in the route table defines a route that is used to direct traffic, and consists of multiple fields, including the destination CIDR block, the next hop for the traffic, and the type of the next hop. Route entries are classified into system route entries and custom route entries.

System routes

After you create a VPC, the system automatically adds the following system routes to the route table:

- A route entry with a destination CIDR block of 100.64.0.0/10. This route is used for communication among cloud resources within the VPC.
- Route entries with destination CIDR blocks same as the CIDR blocks of the VSwitches in this VPC. Such routes are used for communication among cloud resources within VSwitches.

For example, if you create a VPC and specify 192.168.0.0/16 as its CIDR block, and then create two VSwitches whose CIDR blocks are 192.168.1.0/24 and 192.168.0.0/24, three system routes listed in the following table are automatically added to the route table of the VPC.

Destination CIDR block	Next hop	Route entry type
100.64.0.0/10	-	System route
192.168.1.0/24	-	System route
192.168.0.0/24	-	System route

Custom routes

You can add custom routes to replace system routes or route traffic to specified destinations. You can specify the following next hop types when you create a custom route:

- **ECS instance:** Traffic destined for the destination CIDR block is forwarded to a specified Elastic Compute Service (ECS) instance in the VPC.
You can select this type if you want to access the Internet or other applications through the applications deployed on an ECS instance.
- **VPN gateway:** Traffic destined for the destination CIDR block is forwarded to a specified VPN gateway.
You can select this type if you want to connect a VPC to another VPC or a local network through a VPN connection.
- **NAT gateway:** Traffic destined for the destination CIDR block is forwarded to a specified NAT gateway.
You can select this type if you want to connect a VPC to the Internet by using a NAT gateway.
- **Router interface (to VPC):** Traffic destined for the destination CIDR block is forwarded to a specified VPC.
You can select this type if you want to connect two VPCs by using Express Connect.
- **Router interface (to VBR):** Traffic destined for the destination CIDR block is forwarded to a specified Virtual Border Router (VBR).
You can select this type if you want to connect a VPC to an on-premises network by using Express Connect.
- **Secondary ENI:** Traffic destined for the destination CIDR block is forwarded to a specified secondary Elastic Network Interface (ENI).

IPv6 routes

If IPv6 is enabled for your VPC, the following route entries are automatically added to the system route table of the VPC:

- A custom route entry with a destination CIDR block of `::/0` and whose next hop is the IPv6 gateway. This route is used to direct traffic between the cloud resources deployed in the VPC and the Internet by using IPv6 addresses.
- System route entries whose destination CIDR blocks are the IPv6 CIDR blocks of the VSwitches in the VPC. Such routes are used for communication within the VSwitches.

Routing rules

If one destination address matches more than one route entry in a route table, the system selects an entry by implementing the longest prefix match algorithm, whereby the most specific of the matching entries, the one with the longest subnet mask, is used to route traffic.

The following table describes a route table of a VPC.

Destination CIDR block	Next hop type	Next hop	Route entry type
100.64.0.0/10	-	-	System route
192.168.0.0/24	-	-	System route
0.0.0.0/0	Instance	i-12345678	Custom route
10.0.0.0/24	Instance	i-87654321	Custom route

As shown in the preceding table, the route entries destined for `100.64.0.0/10` and `192.168.0.0/24` are system route entries, and the two destined for `0.0.0.0/0` and `10.0.0.0/24` are custom route entries. Traffic destined for `0.0.0.0/0` is forwarded to the ECS instance `i-12345678`, and traffic destined for `10.0.0.0/24` is forwarded to the ECS instance `i-87654321`. According to the longest prefix match algorithm, traffic destined for `10.0.0.1` is forwarded to the ECS instance `i-87654321`, and traffic destined for `10.0.1.1` is forwarded to the ECS instance `i-12345678`.

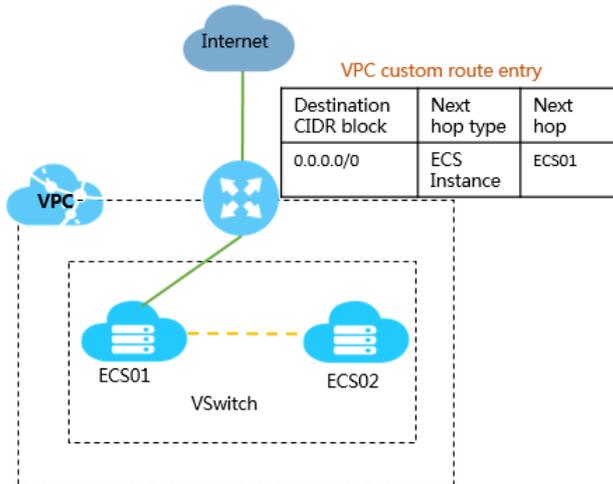
Routing examples

You can add custom route entries to a route table to control inbound and outbound traffic for a VPC.

- Routing within a VPC

As shown in the following figure, a NAT gateway is deployed on an ECS instance (ECS01) in a VPC. To enable the cloud resources in this VPC to access the Internet by this ECS instance, you can add the following route entry to the route table.

Destination CIDR block	Next hop type	Next hop
0.0.0.0/0	ECS instance	ECS01



- Connect two VPCs by using Express Connect

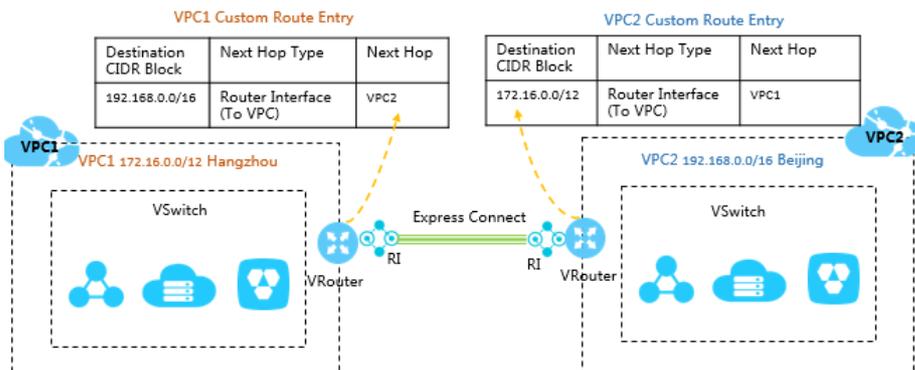
As shown in the following figure, VPC1 (172.16.0.0/12) needs to be connected to VPC2 (192.168.0.0/16) by using Express Connect. After you create router interfaces for interconnection, you must add the following route entries in the route table of each VPC respectively.

- VPC1

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	Router interface (to VPC)	VPC2

- VPC2

Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	Router interface (to VPC)	VPC1



- Connect two VPCs by using VPN gateways

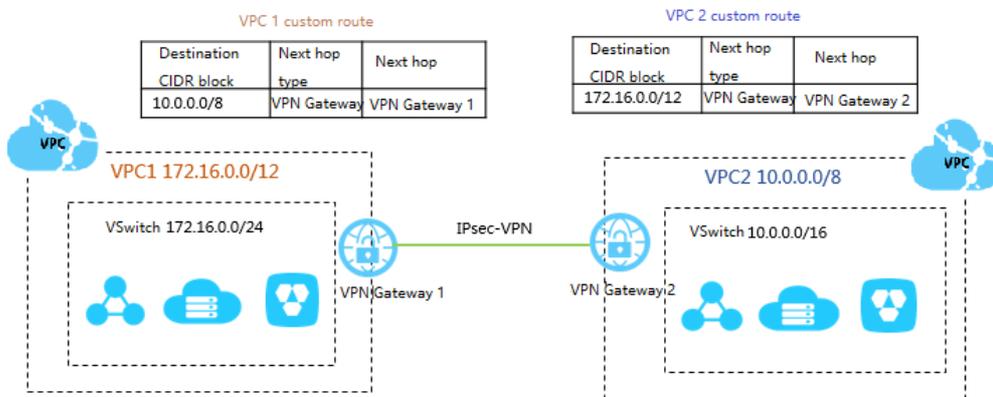
As shown in the following figure, VPC1 (172.16.0.0/12) needs to be connected to VPC2 (10.0.0.0/8) with VPN gateways. After you configure the VPN gateways, you must add the following route entries in the route table of each VPC respectively.

- VPC1

Destination CIDR block	Next hop type	Next hop
10.0.0.0/8	VPN gateway	VPN gateway 1

- VPC2

Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	VPN gateway	VPN gateway 2



- Connect a VPC to an on-premises data center by using Express Connect

As shown in the following figure, a VPC needs to be connected to an on-premises data center by using Express Connect. After you configure an Express Connect circuit and a VBR, you must add the following route entries for related networks and devices:

- VPC

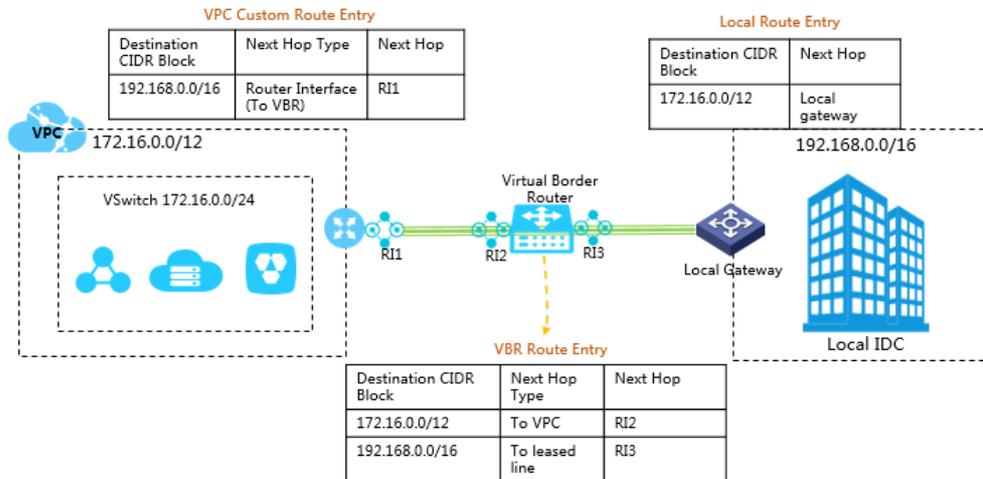
Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	Router interface (general routing)	Router interface RI1

- VBR

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	To the Express Connect circuit	Router interface RI3
172.16.0.0/12	To VPC	Router interface RI2

○ On-premises network

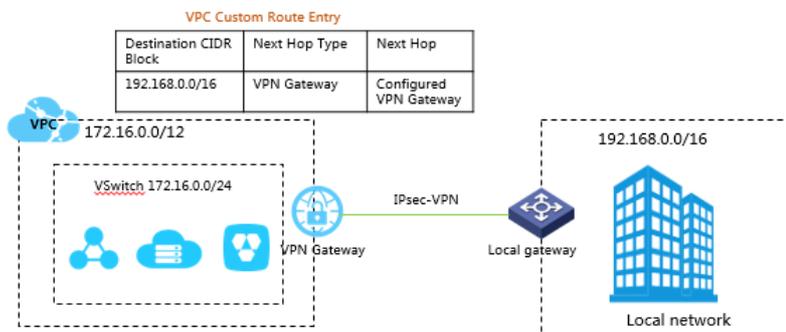
Destination CIDR block	Next hop type	Next hop
172.16.0.0/12	-	A specified local gateway device



● Connect a VPC to an on-premises data center by using VPN gateways

As shown in the following figure, a VPC (172.16.0.0/12) needs to be connected to an on-premises data center (192.168.0.0/16). After you configure the VPN gateway, you must add the following route entry to the route table of the VPC.

Destination CIDR block	Next hop type	Next hop
192.168.0.0/16	VPN gateway	A specified VPN gateway



22.5.2. Add a custom route entry

This topic describes how to add a custom route entry. After you create a Virtual Private Cloud (VPC) network, the system creates a default route table and adds system route entries to the route table for traffic management. You cannot create or delete system route entries. However, you can create custom route entries to route traffic from source CIDR blocks to specific destinations.

Context

Each entry in the route table is a route entry. A route entry, which specifies the destination for network traffic, consists of the destination CIDR block, next hop type, and next hop. Route entries are classified into system route entries and custom route entries.

Procedure

1. [Log on to the VPC console.](#)
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route tables belong.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.
5. In the **Route Table Details** section, click the **Route Entry List** tab, and then click **Add Route Entry**.
6. On the **Add Route Entry** page, set the following parameters and click **OK**.

Parameter	Description
Name	The name of the route entry. The name must be 2 to 128 characters and can contain digits, underscores (_), and hyphens (-). The name must start with a letter or a Chinese character.
Destination CIDR Block	The destination CIDR block to which you want to route traffic.
Next Hop Type	<p>The type of the next hop. Valid values:</p> <ul style="list-style-type: none"> ◦ ECS Instance: Traffic destined for the specified CIDR block is routed to the Elastic Compute Service (ECS) instance you select. Select this type if you want to route traffic to an ECS instance for centralized traffic forwarding and management. For example, when an ECS instance is configured as the Internet-facing gateway to manage the traffic from other ECS instances to the Internet. ◦ VPN Gateway: Traffic destined for the specified CIDR block is routed to the VPN gateway you select. ◦ NAT Gateway: Traffic destined for the specified CIDR block is routed to the NAT gateway you select. ◦ Router Interface (To VPC): Traffic destined for the specified CIDR block is routed to the VPC network you select. Select this type if you want to connect VPC networks through Express Connect. ◦ Router Interface (To VBR): Traffic destined for the specified CIDR block is routed to the router interface that is associated with a Virtual Border Router (VBR). Select this type if you want to connect the VPC network to an on-premises data center through Express Connect. <p>If you select Router Interface (To VBR), you must also select a routing mode:</p> <ul style="list-style-type: none"> ▪ General Routing: Select an associated router interface. ▪ Active/Standby Routing: Select two instances as the next hop. The active route has a weight of 100 and the standby route has a weight of 0. The standby route takes over when the active route fails the health check. ▪ Load Balancing: Select two to four router interfaces as the next hop. The peer router of each router interface must be a VBR. Valid values of the instance weight: 1 to 255. The value must be an integer and the default value is 100. The weights of the selected instances must be the same. This way, traffic can be evenly distributed to the next-hop instances.
ECS Instance/VPN Gateway/NAT Gateway	Select the next-hop instance.

Related information

- [CreateRouteEntry](#)

22.5.3. Export route entries

This topic describes how to export route entries from a route table for backup.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route tables belong.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.
5. In the **Route Table Details** section, click the **Route Entry List** tab, and then click **Export**. The route entries are exported to a `.csv` file in your local computer.

22.5.4. Modify a route table

This topic describes how to modify the name and description of a route table.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route tables belong.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.
5. In the **Route Table Details** section, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the route table and click **OK**. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (`_`), and hyphens (`-`). It must start with a letter or a Chinese character.
6. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description of the route table, and click **OK**. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

22.5.5. Delete a custom route entry

This topic describes how to delete a custom route entry. A route table consists of one or more route entries that determine which way to forward traffic. Note that system route entries cannot be deleted.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **Route Tables**.
3. In the top navigation bar, select the region to which the route tables belong.
4. On the **Route Tables** page, find the route table and click **Manage** in the **Actions** column.
5. On the **Route Entry List** tab, find the target route entry and then click **Delete** in the **Actions** column.
6. In the **Delete Route Entry** dialog box, click **OK**.

22.6. HAVIPs

22.6.1. Overview

A high-availability virtual IP address (HAVIP) is a private IP address that can be created and released as an independent resource. After you associate an HAVIP with an Elastic Compute Service (ECS) instance, the ECS instance can advertise this HAVIP by sending Address Resolution Protocol (ARP) messages.

Features

HAVIPs have the following features:

- Each HAVIP can be associated with a maximum of two ECS instances. After the association, the ECS instance can advertise the associated HAVIP by sending ARP messages.
- Each ECS instance can claim more than one private IP address by advertising multiple HAVIPs.
- By allowing ECS instances to advertise HAVIPs with ARP messages, a Virtual Router Redundancy Protocol (VRRP) configuration is implemented to ensure high availability of your services.
- Each HAVIP can be associated with an elastic IP address (EIP), which means that after an HAVIP is disassociated from the previous ECS instance and reassociated with a different ECS instance, the traffic sent to the EIP can also be delivered to the current ECS instance.

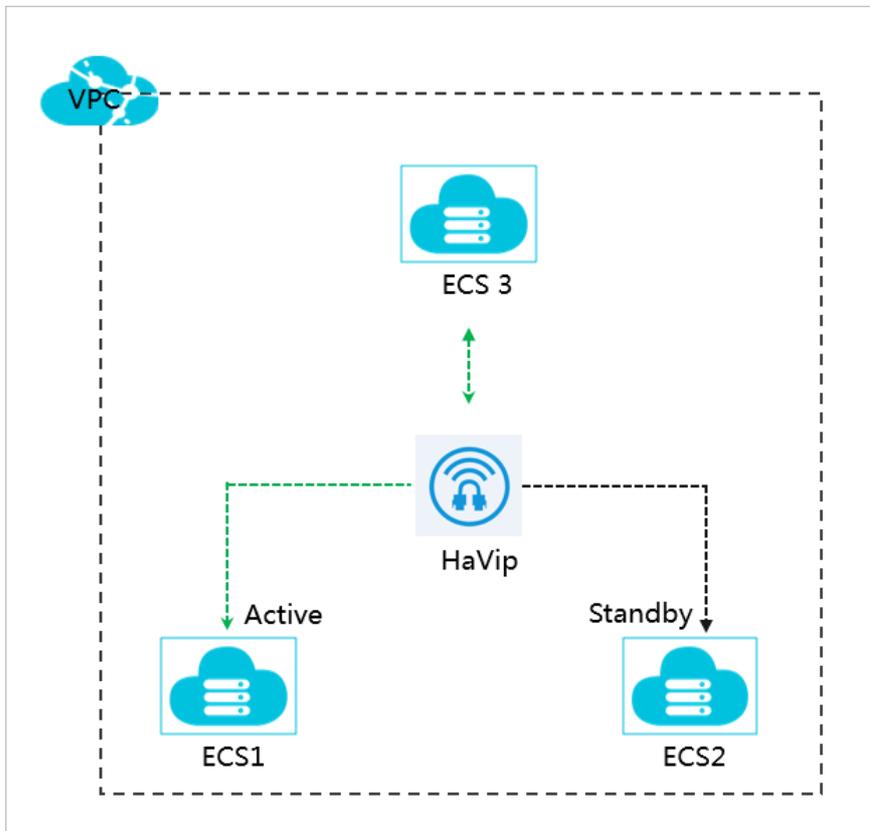
Scenarios

HAVIPs support flexible configurations in the following scenarios.

- Scenario 1: High availability services for private networks

High-availability solutions are provided by using open-source programs such as Keepalived and Heartbeat.

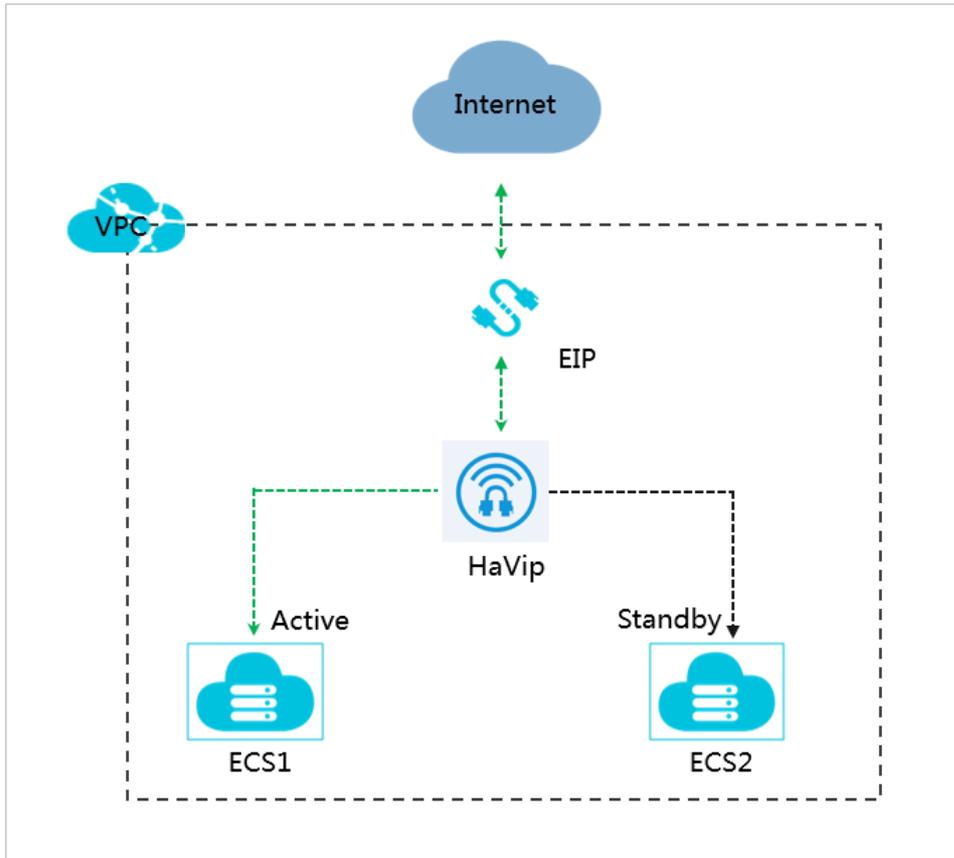
In the following figure, Keepalived integrates the HAVIP and two ECS instances (ECS 1 and ECS 2) into a highly available solution designed for the private network. The cloud instances in the same virtual private cloud (VPC) can access this service over the private network from the IP address of this HAVIP, which is also the endpoint. The high availability of this solution makes sure that ECS 2 is always ready to serve on the same IP address in case ECS 1 fails.



- Scenario 2: High availability services on the Internet

High-availability solutions are provided by using open-source programs such as Keepalived and Heartbeat.

In the following figure, ECS 1 and ECS 2 are associated with an HAVIP, which is configured with an EIP. In this case, the service can be accessed over the Internet from its endpoint, that is, the IP address of the HAVIP. Keepalived is used to implement failover, where ECS 2 takes over to keep the service available on the same IP address in case ECS 1 fails.



Limits

Before you use HAVIPs, note the following limits:

- HAVIPs can only be deployed in VPCs.
- HAVIPs do not support multicasting or broadcasting.
- Each account can hold a maximum of five HAVIPs.
- A maximum of five HAVIPs can exist in a VPC.
- Each ECS instance can be associated with a maximum of five HAVIPs.
- Each HAVIP can be associated with a maximum of two ECS instances.
- Each VPC can have a maximum of five route entries destined for HAVIPs.

22.6.2. Create an HAVIP

A high-availability virtual IP address (HAVIP) is a private IP address that can be created and released as an independent resource. This topic describes how to create an HAVIP in the console.

Procedure

1. [Log on to the VPC console.](#)
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Click **Create HaVip Address**.
4. Configure the HAVIP and then click **Submit**. The following table describes the parameters for creating an HAVIP.

Parameter	Description
Organization	Select the organization to which the HAVIP belongs.

Parameter	Description
Resource Set	Select the resource set to which the HAVIP belongs.
Region	Select a region to deploy the HAVIP.
VPC	Select the VPC to which the HAVIP belongs.
VSwitch	Select the VSwitch to which the HAVIP belongs.
Private IP Address	Specify a private IP address for the HAVIP. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 5px;"> 🔊 Notice You must select an available private IP address in the CIDR range of the VSwitch. </div>

22.6.3. Associate an HAVIP with an ECS instance

This topic describes how to associate a high-availability virtual IP address (HAVIP) with an Elastic Compute Service (ECS) instance deployed in a virtual private cloud (VPC). After the association, the ECS instance can advertise this HAVIP by sending Address Resolution Protocol (ARP) messages. Each HAVIP can be associated with a maximum of two ECS instances.

Prerequisites

An ECS instance is created and placed in the same region as the HAVIP to be associated.

For more information, see [Create an instance](#) under [Quick start](#) in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HAVIP.
4. Find the target HAVIP and then click **Manage** in the **Actions** column.

5. In the **Resources** section, click the  icon.



6. On the page that appears, select the ECS instance to associate and click **OK**.
7. On the **HaVip Details** page, click **Refresh** to check the ECS associations of this HAVIP.



After you associate an HA VIP with an ECS instance, you must configure the HA VIP on the network interface of the ECS instance.

22.6.4. Associate an HA VIP with an EIP

This topic describes how to associate a high-availability virtual IP address (HA VIP) with an elastic IP address (EIP). After the association, you can use the HA VIP to provide your instances with public network access.

Prerequisites

You have purchased an EIP. For more information, see [Create an elastic IP address](#) under [Quick start](#) in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. **Log on to the VPC console.**
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HA VIP.
4. Find the target HA VIP instance and choose **More > Bind EIP Address** in the **Actions** column.
5. On the page that appears, select the EIP to associate and click **OK**.

22.6.5. Disassociate an HA VIP from an ECS instance

This topic describes how to disassociate a high-availability virtual IP address (HA VIP) from an Elastic Compute Service (ECS) instance. After the disassociation, the ECS instance can no longer announce this HA VIP by sending Address Resolution Protocol (ARP) messages.

Procedure

1. **Log on to the VPC console.**
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HA VIP.
4. On the **HaVip Addresses** page, find the target HA VIP and then click **Manage** in the **Actions** column.
5. In the **Resources** section, find the target ECS instance and then click **Unbind**.



6. In the dialog box that appears, click **OK**.

22.6.6. Disassociate an EIP from an HAVIP

This topic describes how to disassociate a high-availability virtual IP address (HAVIP) from an elastic IP address (EIP). After the disassociation, you can no longer use the HAVIP to provide your instances with Internet access.

Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HAVIP.
4. On the **HaVip Addresses** page, find the target HAVIP and then choose **More > Unbind with EIP** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

22.6.7. Delete an HAVIP

This topic describes how to delete a high-availability virtual IP address (HAVIP).

Prerequisites

- The HAVIP is not associated with any elastic IP address (EIP).
If the HAVIP is associated with an EIP, before you delete the HAVIP, you must disassociate it from the EIP. For more information, see [Disassociate an EIP from an HAVIP](#).
- The HAVIP is not associated with any Elastic Compute Service (ECS) instance.
If the HAVIP is associated with one or more ECS instances, before you delete the HAVIP, you must disassociate it from the ECS instances. For more information, see [Disassociate an HAVIP from an ECS instance](#).

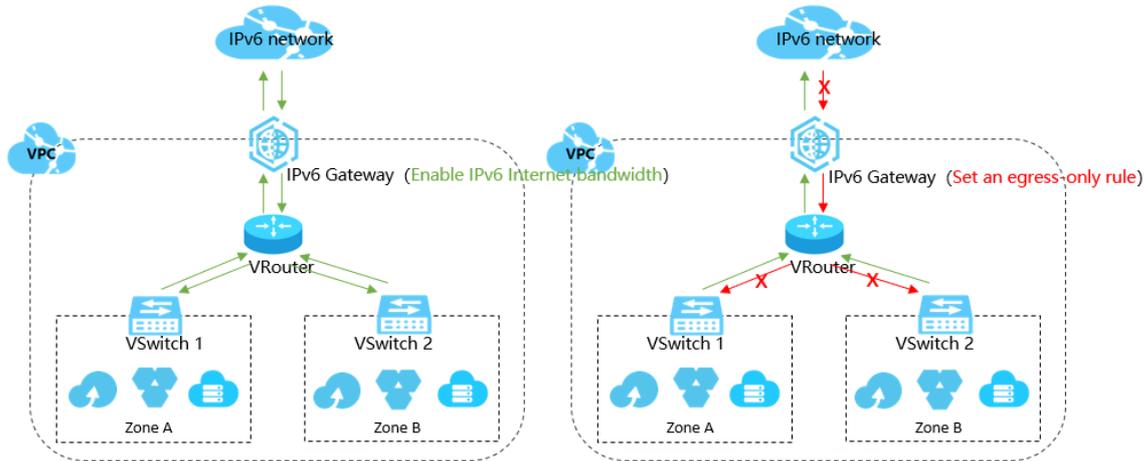
Procedure

1. [Log on to the VPC console](#).
2. In the left-side navigation pane, click **HaVip Addresses**.
3. Select the region of the HAVIP.
4. On the **HaVip Addresses** page, find the target HAVIP and then choose **More > Delete** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

23. IPv6 Gateway

23.1. What is an IPv6 Gateway?

This topic provides an overview of the IPv6 Gateways of Virtual Private Cloud (VPC). An IPv6 Gateway functions as an IPv6 traffic gateway for a VPC. You can configure the IPv6 Internet bandwidth and egress-only rules to manage the inbound and outbound IPv6 traffic.



Functions

The functions of an IPv6 gateway are as follows:

- **IPv6 internal network communication**

By default, an IPv6 address in a VPC is allocated with an Internet bandwidth of 0 Mbit/s and only supports communication over the internal network. Specifically, the cloud instances in a VPC can only access other IPv6 addresses in the same VPC through the IPv6 address. The resources cannot access the Internet with these IPv6 addresses or be accessed by IPv6 clients over the Internet.

- **IPv6 public network communication**

You can purchase an Internet bandwidth for the IPv6 address for which you have applied. In this way, the resources in the VPC can access the Internet through the IPv6 address and be accessed by IPv6 clients over the Internet.

You can set the Internet bandwidth to 0 Mbit/s at any time to deny the IPv6 address Internet access. After this configuration, the IPv6 address can only communicate over the internal network.

- **IPv6 public network communication with an egress-only rule**

You can set an egress-only rule for an IPv6 Gateway. In this way, the IPv6 address can access the Internet, but IPv6 clients are denied access to your cloud resources in the VPC over the Internet.

You can delete the egress-only rule at any time. After the rule is deleted, your resources in the VPC can access the Internet through the IPv6 address for which you have purchased Internet bandwidth, and IPv6 clients can access the resources in the VPC over the Internet.

The network access capability of IPv6 addresses is dependent on the settings of the network type, Internet bandwidth, and egress-only rule, as shown in the following table.

IPv6 network type	Enable IPv6 Internet bandwidth?	Set an egress-only rule?	IPv6 network access capability
Internal network	No	No	Internal network communication

IPv6 network type	Enable IPv6 Internet bandwidth?	Set an egress-only rule?	IPv6 network access capability
Public network	Yes	No	Internal network communication Public network communication
		Yes	Internal network communication Public network communication when access is initiated by VPCs

Benefits

IPv6 Gateway provides the following benefits:

- **High availability**
IPv6 Gateways provide cross-zone high availability and stable IPv6 Internet gateway services.
- **High performance**
A single IPv6 Gateway provides a 10-gigabit level throughput.
- **Flexible management of public network communication**
You can manage the Internet communication capability of an IPv6 Gateway by adjusting its Internet bandwidth and setting an egress-only rule.

23.2. Log on to the IPv6 Gateway console

This topic provides an example of how to use Google Chrome to log on to the IPv6 Gateway console in the Apsara Stack Cloud Management (ASCM) console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.

4. In the top navigation bar, choose **Products > Networking > IPv6 Gateway**.

23.3. Quick start

23.3.1. Create an IPv6 VPC

This topic describes how to build a virtual private cloud (VPC) with an IPv6 Classless Inter-domain Routing (CIDR) block and create an Elastic Compute Service (ECS) instance assigned with an IPv6 address in the VPC.

Step 1: Create a VPC and a VSwitch

Before you deploy cloud resources in a VPC, you must create a VPC and a VSwitch.

Perform the following steps to create a VPC and a VSwitch:

1. Log on to the VPC console.
2. On the VPCs page, click **Create VPC**.
3. On the **Create VPC** page, configure the VPC and click **Submit**. The following table describes the parameters for creating a VPC.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, the administrators of sub-organizations can create resources in the VPC. In this tutorial, select No.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> . In this tutorial, enter <code>VPCTest</code> .
IPv4 CIDR Block	Select an IPv4 CIDR block for the VPC. The following setting methods are supported: <ul style="list-style-type: none"> ◦ Recommended CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8. ◦ Custom CIDR Block: Enter 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or a subset of these CIDR blocks. The subnet mask must be 8 to 28 bits in length. For example, enter 192.168.0.0/16. In this tutorial, select Recommended CIDR Block and then select 192.168.0.0/16. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note After you create a VPC, you cannot change its IPv4 CIDR block. </div>
IPv6 CIDR Block	Specify whether to assign an IPv6 CIDR block to the VPC. <ul style="list-style-type: none"> ◦ Do Not Assign: No IPv6 CIDR block will be assigned to the VPC. ◦ Assign: An IPv6 CIDR block will be automatically assigned to the VPC. In this tutorial, select Assign.

Parameter	Description
Description	<p>Enter a description for the VPC.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>

4. Click **Back to Console**. In the left navigation pane, click **VSwitches**.
5. On the **VSwitches** page, click **Create VSwitch**.
6. On the **VSwitch** page, configure the VSwitch and click **Submit**. The following table describes the parameters for creating a VSwitch.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.
Zone	<p>Select a zone to deploy the VSwitch.</p> <p>Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches that reside in different zones to achieve zone-disaster recovery.</p> <p> Note Each cloud resource can be deployed in only one VSwitch.</p>
VPC	<p>Select the VPC for which you want to create the VSwitch.</p> <p>In this tutorial, select <code>VPCtest</code>.</p>
Dedicated for Off-Cloud Servers	<p>Specify whether the VSwitch is dedicated for off-cloud servers.</p> <p>For more information, see the Features of off-cloud servers for VPC section in the <i>ECS Bare Metal Instance User Guide</i>.</p> <p>In this tutorial, select <code>No</code>.</p>
VSwitch Name	<p>Enter a name for the VSwitch.</p> <p>The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code>.</p>
IPv4 CIDR Block	<p>Enter an IPv4 CIDR block for the VSwitch.</p> <p>This tutorial uses the default IPv4 CIDR block.</p>
IPv6 CIDR Block	<p>Enter an IPv6 CIDR block for the VSwitch.</p> <p>This tutorial uses the default IPv6 CIDR block.</p>

Parameter	Description
Description	<p>Enter a description for the VSwitch.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> or <code>/</code> or <code>.</code></p>

Step 2: Create a security group

Perform the following steps to create a security group:

1. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
2. Choose **Networks and Security > Security Groups**.
3. On the **Security Groups** page, click **Create Security Group**.
4. On the **Create Security Group** page, configure the security group and click **Submit**. The following table describes the parameters for creating a security group.

Parameter	Description
Organization	Select the organization to which the security group belongs.
Resource Set	Select the resource set to which the security group belongs.
Region	<p>Select the region that will use the security group.</p> <p>The security group and the VPC must belong to the same region.</p>
Zone	Select the zone that will use the security group.
VPC	Select the VPC to which the security group belongs.
Security Group Name	<p>Enter a name for the security group.</p> <p>The name must be 2 to 128 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> or <code>/</code> or <code>.</code></p>
Description	<p>Enter a description for the security group.</p> <p>The description must be 2 to 256 characters in length and can contain letters, Chinese characters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or a Chinese character and cannot start with <code>http://</code> or <code>https://</code> or <code>/</code> or <code>.</code></p>

Step 3: Create and configure an ECS instance

After you create a VPC and a VSwitch, you must create an ECS instance and assign an IPv6 address to the ECS instance, and then associate this IPv6 address with the network interface controller (NIC) of the ECS instance.

Perform the following steps to create and configure an ECS instance:

1. In the top navigation bar, choose **Products > Networking > Virtual Private Cloud**.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where your VSwitch resides.
4. On the **VSwitches** page, find the target VSwitch and choose **Purchase > ECS Instance** in the **Actions** column.

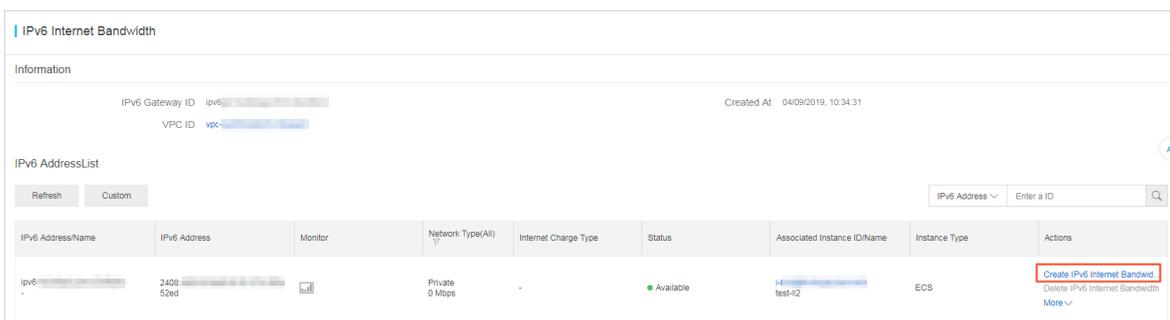
5. On the **Create ECS Instance** page, configure the ECS instance and click **Submit**. In this tutorial, select **Assign** to assign an IPv6 address to the ECS instance. For more information about other parameters, see **Create an instance** under **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.
6. Return to the **Instances** page, and click the instance ID to view the assigned IPv6 address.
7. Configure a static IPv6 address.
 - If the image of your ECS instance supports DHCPv6, you do not need to manually configure the static IPv6 address. DHCPv6 enables automatic configuration of IPv6 addresses. Therefore, if your ECS instance image supports DHCPv6, the ECS instance can use the assigned IPv6 address to communicate in the private network directly after being created.
The following images support DHCPv6:
 - Linux images:
 - CentOS 7.6 IPV6 64Bit
 - CentOS 6.10 64Bit
 - SUSE Linux Enterprise Server 12 SP4 64Bit
 - Windows Server images
 - If the image of your ECS instance does not support DHCPv6, you must manually configure an IPv6 address for the ECS instance. We recommend that you refer to the related documentation for each image for configuration guidance.

Step 4: Purchase an IPv6 Internet bandwidth plan

By default, IPv6 addresses are only used for communication inside private networks. If you want to allow an instance assigned with an IPv6 address to access the Internet or be accessed by IPv6 clients over the Internet, you must purchase an Internet bandwidth plan for the IPv6 address.

Perform the following steps to purchase an Internet bandwidth plan for the IPv6 address:

1. In the top navigation bar, choose **Products > Networking > IPv6 Gateway**.
2. Select the region where the IPv6 Gateway resides.
3. On the **IPv6 Gateway** page, find the target IPv6 Gateway and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the **IPv6 Internet Bandwidth** page, find the target IPv6 address and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Select a bandwidth plan and then click **Submit**. The maximum IPv6 Internet bandwidth for an IPv6 Gateway of the Free, Enterprise, or Enhanced Edition is 2 Gbit/s.

Step 5: Configure the security group rules

IPv4 and IPv6 addresses are independent of each other. If the current security group rules cannot serve your IPv6 services, you must configure security group rules for IPv6 traffic.

For more information, see **Add security group rules** under **Security groups** in the *Apsara Stack Elastic Compute Service User Guide*.

Step 6: Test the network connectivity

Log on to the ECS instance and run the ping command to test the network connectivity.

```
[root@iZhp3aehva ~]# ping6 ipv6.baidu.com
PING ipv6.baidu.com(2400:da00:2::29 (2400:da00:2::29)) 56 data bytes
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=1 ttl=45 time=77.1 ms
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=2 ttl=45 time=77.1 ms
64 bytes from 2400:da00:2::29 (2400:da00:2::29): icmp_seq=3 ttl=45 time=77.0 ms
^C
--- ipv6.baidu.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 77.070/77.101/77.127/0.227 ms
[root@iZhp3aehva ~]#
```

23.4. Enable IPv6 for VPCs

23.4.1. Create an IPv4 and IPv6 dual-stack VPC

This topic describes how to configure both IPv4 and IPv6 CIDR blocks for a VPC when you create the VPC. By default, all VPCs are associated with IPv4 CIDR blocks which cannot be deleted. However, you can choose whether to allocate an IPv6 CIDR blocks to a VPC. After you choose to allocate an IPv6 CIDR block to a VPC, the system creates an IPv6 gateway of the Free Edition for the VPC for you to provision IPv6 bandwidth and manage IPv6 traffic.

Procedure

1. Log on to the VPC console.
2. On the top of the page, select a region to deploy your VPC.
3. On the VPCs page, click **Create VPC**.
4. On the **Create VPC** page, configure the VPC network and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the VPC belongs.
Resource Set	Select the resource set to which the VPC belongs.
Region	Select a region to deploy the VPC.
Share with Sub-organizations	Specify whether to share the VPC. If you select Yes, administrators of sub-organizations can create resources in the VPC network.
VPC Name	Enter a name for the VPC. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code> .

Parameter	Description
IPv4 CIDR Block	<p>Specify an IPv4 CIDR block for the VPC. The following setting methods are supported:</p> <ul style="list-style-type: none"> ◦ Recommended CIDR Block: Use 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8 as the IPv4 CIDR block of the VPC. ◦ Custom CIDR Block: Use 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or one of their subnets as the IPv4 CIDR block of the VPC. The subnet mask must be 8 to 24 bits in length. For example, enter 192.168.0.0/16. <p>Note After you create a VPC, you cannot modify its IPv4 CIDR block.</p>
IPv6 CIDR Block	<p>Specify whether to assign an IPv6 CIDR block to the VPC. In this example, select Assign.</p> <p>If you set this parameter to Assign, the system automatically creates an IPv6 gateway of the Free Edition for your VPC and assigns an IPv6 CIDR block with the subnet mask /61, such as 2xx1:db8::/61. By default, IPv6 addresses can only be used for communication within private networks. If you need to enable an IPv6 address to access and be accessed over the Internet, you must purchase IPv6 Internet bandwidth for the IPv6 address. For more information, see Enable Internet connectivity for an IPv6 address.</p> <p>Note After you create a VPC, you cannot modify its IPv6 CIDR block.</p>
Description	<p>Enter a description for the VPC.</p> <p>The description must be 2 to 256 characters in length and can contain letters, digits, underscores (<code>_</code>), hyphens (<code>-</code>), periods (<code>.</code>), colons (<code>:</code>), and commas (<code>,</code>). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.</p>

23.4.2. Enable an IPv6 CIDR block for a VPC network

This topic describes how to enable an IPv6 CIDR block for a Virtual Private Cloud (VPC) network. After the IPv6 CIDR block is enabled, the system automatically creates an IPv6 gateway free of charge for the VPC network. You can use the IPv6 Gateway to manage the IPv6 Internet bandwidth and set egress-only rules.

Procedure

1. Log on to the VPC console.
2. Select the region where your VPC network is deployed.
3. On the **VPCs** page, find the target VPC network and click **Enable IPv6 CIDR Block** in the **IPv6 CIDR Block** column.

Instance ID/Name	CIDR	IPv6 CIDR Block	Status	Default VPC	Route Tables	VSwitches	Organization	Resource Set	Actions
vpc-9dbr1	192.168.0.0/16	Enable IPv6 CIDR Block	Available	No	1	1	yundun	ResourceSet(yundun)	Manage Delete
vpc-9dbr1	172.16.0.0/16	Enable IPv6 CIDR Block	Available	No	1	1	yundun	ResourceSet(yundun)	Manage Delete

4. In the **Enable IPv6 CIDR Block** dialog box, select **Enable IPv6 CIDR Block of all VSwitches in VPC**, and then click **OK**. If you do not select **Enable IPv6 CIDR Block of all VSwitches in VPC**, you must enable IPv6 CIDR Block for each VSwitch. For more information, see [Enable an IPv6 CIDR block for a VSwitch](#).

23.5. Enable IPv6 for VSwitches

23.5.1. Create an IPv4 and IPv6 dual-stack VSwitch

This topic describes how to assign an IPv6 CIDR block to a VSwitch when you create the VSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where you want to deploy the VSwitch.
4. On the **VSwitches** page, click **Create VSwitch**.
5. On the **VSwitch** page, configure the VSwitch and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the VSwitch belongs.
Resource Set	Select the resource set to which the VSwitch belongs.
Region	Select a region to deploy the VSwitch.
Zone	<p>Select a zone to deploy the VSwitch.</p> <p>Each VSwitch must reside entirely within one zone and cannot span multiple zones. However, you can deploy cloud resources in VSwitches of different zones to achieve cross-zone disaster recovery.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Each cloud resource can be deployed in only one VSwitch.</p> </div>
VPC	Select the VPC for which you want to create the VSwitch.
Dedicated for Off-Cloud Servers	<p>Specify whether the VSwitch is dedicated for off-cloud servers.</p> <p>For more information, see <i>the Features of off-cloud servers for VPC</i> topic in the Apsara Stack Bare Metal Server User Guide.</p>
VSwitch Name	<p>Enter a name for the VSwitch.</p> <p>The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.</p>
IPv4 CIDR Block	<p>Specify an IPv4 CIDR block for the VSwitch.</p> <ul style="list-style-type: none"> ○ You must specify the IP address range for the VSwitch in the form of a CIDR block. The IPv4 CIDR block size for a VSwitch is between a 16-bit mask and a 29-bit mask. It means that 8 to 65,536 IP addresses can be provided. ○ The IPv4 CIDR block of a VSwitch must be a subset of the IPv4 CIDR block of the VPC this VSwitch resides in. ○ The first and the last three IP addresses of each VSwitch IPv4 CIDR block are reserved. For example, if the VSwitch IPv4 CIDR block is 192.168.1.0/24, the IP addresses 192.168.1.0, 192.168.1.253, 192.168.1.254, and 192.168.1.255 are reserved. ○ The IPv4 CIDR block of a VSwitch must be more specific than the CIDR range of a route in any of the VPC route tables. ○ After you create a VSwitch, you cannot modify its IPv4 CIDR block.

Parameter	Description
IPv6 CIDR Block	<p>Specify an IPv6 CIDR block for the VSwitch.</p> <p>The default subnet mask for the IPv6 CIDR block of a VSwitch is /64. You can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block.</p> <p>For example, if the IPv6 CIDR block of the VPC that contains the VSwitch is 2xx1:db8::/64, you can enter 255 (FF in hexadecimal notation) in this field to define the IPv6 CIDR block of the VSwitch as 2xx1:db8:ff::/64.</p>
Description	<p>Enter a description for the VSwitch.</p> <p>The description must be 2 to 256 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter and cannot start with <code>http://</code> or <code>https://</code>.</p>

23.5.2. Enable IPv6 for a VSwitch

This topic describes how to allocate an IPv6 CIDR block to a VSwitch.

Procedure

1. Log on to the VPC console.
2. In the left-side navigation pane, click **VSwitches**.
3. Select the region where your VSwitch resides.
4. On the **VSwitches** page, find the target VSwitch, and click **Enable IPv6 CIDR Block** in the **IPv6 CIDR Block** column.

Instance ID/Name	VPC	Status	IPv4 CIDR Block	Number of Available Private IPs	IPv6 CIDR Block	Default VSwitch
vsw-2wff	vpc-9-szw	Available	172.31.0.0/16	65532	Enable IPv6 CIDR Block	No
vsw-iyay	vpc-9-ka	Available	172.16.0.0/26	59	1007:1100:0:58::/64	No

5. (Optional) In the **Enable IPv6 CIDR Block** dialog box, click **OK**.

Note This operation is required only when IPv6 is not enabled for the VPC to which the VSwitch belongs.

6. Specify an IPv6 CIDR block and click **OK**.

The default subnet mask for the IPv6 CIDR block of a VSwitch is /64. You can enter a decimal number ranging from 0 to 255 to define the last 8 bits of the IPv6 CIDR block.

For example, if the IPv6 CIDR block of the VPC that contains the VSwitch is 2xx1:db8::/64, you can enter 255 (FF in hexadecimal notation) in this field to define the IPv6 CIDR block of the VSwitch as 2xx1:db8:ff/64.

23.6. Manage IPv6 Gateways

23.6.1. Editions of IPv6 gateways

This topic describes the different editions of IPv6 gateways. Different quotas and limits apply to IPv6 gateways of different editions, such as the maximum forwarding bandwidth, maximum IPv6 bandwidth per IPv6 address, and maximum number of egress-only rules.

IPv6 gateway edition	Maximum forwarding bandwidth	Maximum IPv6 bandwidth per IPv6 address	Maximum number of egress-only rules
Free Edition	10 Gbit/s	2 Gbit/s	0
Enterprise Edition	20 Gbit/s	2 Gbit/s	50
Enhanced Enterprise Edition	50 Gbit/s	2 Gbit/s	200

23.6.2. Create an IPv6 gateway

This topic describes how to create an IPv6 gateway for a VPC. After you create an IPv6 gateway, you can purchase IPv6 Internet bandwidth and set egress-only rules for the IPv6 gateway.

Prerequisites

Make sure that IPv6 is enabled for the VPC before you create an IPv6 gateway for the VPC. For more information, see [Allocate an IPv6 CIDR block when you create a VPC](#) and [Enable an IPv6 CIDR block for a VPC network](#).

Procedure

1. [Log on to the IPv6 Gateway console](#).
2. On the IPv6 Gateway page, click **Create IPv6 Gateway**.
3. Configure the IPv6 gateway and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the IPv6 gateway belongs.
Resource Set	Select the resource set to which the IPv6 gateway belongs.
Region	Select a region to deploy the IPv6 gateway. The IPv6 gateway must belong to the same region as the VPC for which you want to create the IPv6 gateway.
VPC	Select the VPC for which you want to create an IPv6 gateway. The target VPC may not be available due to the following reasons: <ul style="list-style-type: none"> ○ Only one IPv6 gateway can be created for each VPC. The VPC already has an IPv6 gateway. ○ The VPC has a custom route with the destination CIDR block set to <code>::/0</code>. If this happens, you must delete this custom route before you can create an IPv6 gateway for the VPC. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note After an IPv6 gateway is created, you cannot change the VPC that is associated with the IPv6 gateway.</p> </div>

Parameter	Description
Edition	<p>Select the edition of the IPv6 gateway. IPv6 gateways are available in the following editions:</p> <ul style="list-style-type: none"> ◦ Free Edition ◦ Enterprise Edition ◦ Enhanced Enterprise Edition <p>Different quotas and limits apply to IPv6 gateways of different editions, such as the maximum forwarding bandwidth, maximum IPv6 bandwidth per IPv6 address, and maximum number of egress-only rules. For more information, see Editions of IPv6 gateways.</p>

23.6.3. Modify an IPv6 gateway

This topic describes how to modify the name and description of an IPv6 gateway.

Procedure

1. [Log on to the IPv6 Gateway console](#).
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. On the IPv6 Gateway Details page, click **Edit** next to **Name**. In the dialog box that appears, enter a new name for the IPv6 gateway, and click **OK**. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter.
5. Click **Edit** next to **Description**. In the dialog box that appears, enter a new description for the IPv6 gateway and click **OK**. The description must be 2 to 256 characters in length and cannot start with `http://` or `https://`.

23.6.4. Delete an IPv6 gateway

This topic describes how to delete an IPv6 gateway. If a VPC no longer needs to access or be accessed by IPv6 clients, you can delete the IPv6 gateway associated with the VPC.

Prerequisites

Before you delete an IPv6 gateway of the enterprise edition or enhanced enterprise edition, you must delete the egress-only rules of the IPv6 gateway. For more information, see [Delete an egress-only rule](#).

Procedure

1. [Log on to the IPv6 Gateway console](#).
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Delete** in the **Actions** column.
4. In the dialog box that appears, click **OK**.

23.7. Manage IPv6 Internet bandwidth

23.7.1. Enable Internet connectivity for an IPv6 address

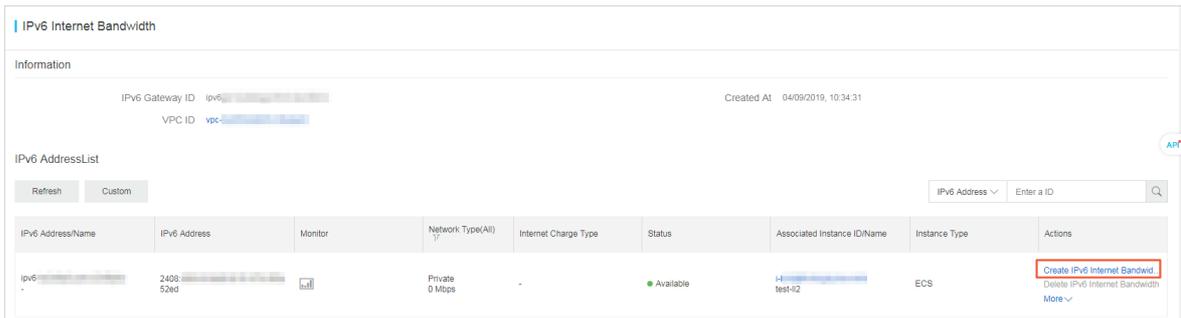
This topic describes how to enable Internet bandwidth for an IPv6 address. After Internet bandwidth is enabled for an IPv6 address, the IPv6 address can be used to communicate over the Internet.

Prerequisites

An ECS instance is created in the VPC associated with the corresponding IPv6 gateway and is configured with an IPv6 address.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. On the IPv6 Internet Bandwidth page, find the target IPv6 address, and click **Enable IPv6 Internet Bandwidth** in the **Actions** column.



6. Specify a bandwidth and click **Submit**. An IPv6 gateway of the Free Edition, Enterprise Edition, or Enhanced Enterprise Edition supports a 2 Gbit/s maximum bandwidth per IPv6 address.

23.7.2. Modify the maximum bandwidth of an IPv6 address

This topic describes how to modify the maximum bandwidth of an IPv6 address. The modification takes effect immediately.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. In the IPv6 AddressList section, find the target IPv6 address, and choose **More > Modify IPv6 Internet Bandwidth** in the **Actions** column.
6. Specify a bandwidth and click **Submit**.

23.7.3. Disable Internet connectivity for an IPv6 address

This topic describes how to delete the Internet bandwidth of an IPv6 address that is no longer needed for Internet communication.

Procedure

1. Log on to the IPv6 Gateway console.
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **IPv6 Internet Bandwidth**.
5. In the IPv6 AddressList section, find the target IPv6 address, and click **Delete IPv6 Internet Bandwidth** in the **Actions** column.
6. In the dialog box that appears, click **OK**.

23.8. Manage egress-only rules

23.8.1. Create an egress-only rule

This topic describes how to create an egress-only rule. If you want an instance in the VPC associated with an IPv6 gateway to be able to access the Internet with an IPv6 address, while resources on the Internet cannot initiate communication with this instance, you can create an egress-only rule for the instance.

Prerequisites

IPv6 Internet bandwidth is enabled for the IPv6 address configured for the instance. For more information, see [Enable Internet connectivity for an IPv6 address](#).

Context

IPv6 gateways of the Free Edition do not support egress-only rules. IPv6 gateways of the enterprise edition and enterprise enhanced edition support a maximum of 50 and 200 egress-only rules respectively.

Procedure

1. [Log on to the IPv6 Gateway console](#).
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Egress-only Rule**.
5. On the Egress-only Rule page, click **Create Egress-only Rule**.
6. In the **Create Egress-only Rule** dialog box, select the ECS instance that uses the IPv6 address to communicate with the Internet, and click **OK**.

23.8.2. Delete an egress-only rule

This topic describes how to delete an egress-only rule. After the rule is deleted, the IPv6 address with Internet bandwidth can access the Internet, and the instance associated with the IPv6 address can be accessed over the Internet.

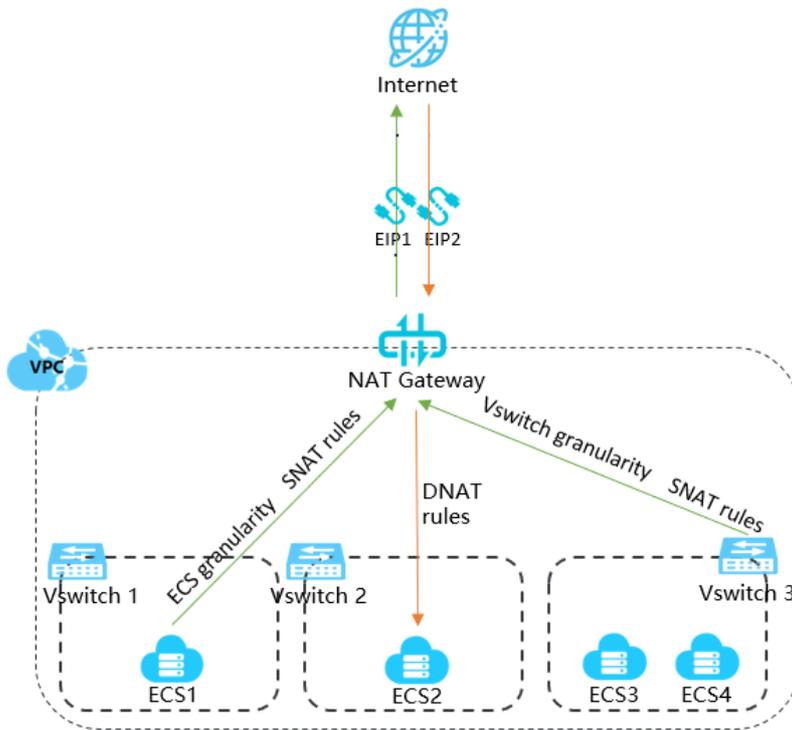
Procedure

1. [Log on to the IPv6 Gateway console](#).
2. Select the region where the IPv6 gateway resides.
3. On the IPv6 Gateway page, find the target IPv6 gateway, and click **Manage** in the **Actions** column.
4. In the left-side navigation pane, click **Egress-only Rule**.
5. On the Egress-only Rule page, find the egress-only rule that you want to delete, and click **Delete** in the **Actions** column.
6. In the dialog box that appears, click **OK**.

24.NAT Gateway

24.1. What is NAT Gateway?

A NAT gateway is an enterprise-grade Internet gateway. NAT Gateway provides source network address translation (SNAT) and destination network address translation (DNAT) features, a maximum forwarding capacity of 10 Gbit/s, and support for cross-zone disaster recovery.



Features

NAT gateways must be associated with public IP addresses. After you create a NAT gateway, you can associate it with one or more elastic IP addresses (EIPs).

NAT Gateway supports SNAT and DNAT.

- SNAT allows Elastic Compute Service (ECS) instances that are deployed in a virtual private cloud (VPC) and not associated with public IP addresses to access the Internet.
- DNAT maps public IP addresses of a NAT gateway to ECS instances so that the ECS instances can be accessible from the Internet.

24.2. Log on to the NAT Gateway console

This topic describes how to use Google Chrome to log on to the NAT Gateway console in Apsara Stack Cloud Management (ASCM).

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Network > NAT Gateway**.

24.3. Quick Start

24.3.1. Overview

This topic describes how to configure Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT). You can configure SNAT and DNAT to enable ECS instances in a Virtual Private Cloud (VPC) network to communicate with the Internet through a NAT gateway.

Prerequisites

Before you start, make sure that the following conditions are met:

- A VPC network is created. For more information, see *VPC User Guide Create a VPC network in the Quick Start chapter in the VPC User Guide*.
- An ECS instance is created in the VPC network. For more information, see *ECS User Guide Create an instance in the Quick Start chapter in the ECS User Guide*.
- An elastic IP address (EIP) is created. For more information, see *EIP User Guide Create an EIP in the Quick Start chapter in the EIP User Guide*.

Procedure

In this topic, an ECS instance that is not associated with any public IP addresses in a VPC network is used as an example. The following flowchart shows how to associate an EIP with a NAT gateway:



Create a NAT Gateway Associate an EIP Create a DNAT entry Create an SNAT entry

- Region
- VPC
- Specification
- Select an EIP
- Public IP address
- Private IP address
- Port settings
- VSwitch granularity
- ECS granularity

1. Create a NAT gateway

A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

For more information, see [Create a NAT Gateway](#).

2. Associate an EIP to a NAT gateway

A NAT gateway functions as expected only after it is associated with a public IP address. After you create a NAT gateway, you can associate it with an EIP.

For more information, see [Associate an EIP with a NAT Gateway](#).

3. Create a DNAT entry

This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps public IP addresses to Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

For more information, see [Create a DNAT entry](#).

4. Create an SNAT entry

This topic describes how to create a Source Network Address Translation (SNAT) entry. SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

For more information, see [Create a SNAT entry](#).

24.3.2. Create a NAT gateway

A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

Prerequisites

A VPC network is created. For more information, see *VPC User Guide* [Create a VPC network](#) in the **Quick Start** chapter in the VPC User Guide.

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateway** page, click **Create NAT Gateway**.
3. Set the parameters for the NAT gateway based on the following information, and then click **Submit**.

Parameter	Description
Organization	The organization to which the NAT gateway belongs.
Resource set	The resource set to which the NAT gateway belongs.
Region	The region where the NAT gateway is deployed.
VPC	<p>The VPC network to which the NAT gateway belongs.</p> <p>If you cannot find the target VPC network in the list, perform the following operations:</p> <ul style="list-style-type: none"> ◦ Check whether the VPC network is already associated with a NAT gateway. Each VPC network can be associated with only one NAT gateway. ◦ Check whether the VPC network has a custom route entry with the destination CIDR block set to 0.0.0.0/0. If such a custom route entry exists, delete it. ◦ Check whether the RAM user is authorized to access the VPC network. If the RAM user is not authorized, contact your Alibaba Cloud account owner to grant permissions.

Parameter	Description
Specification	<p>Select the size of the NAT gateway. Valid values:</p> <ul style="list-style-type: none"> ○ Small: supports up to 10,000 SNAT connections. ○ Medium: supports up to 50,000 SNAT connections. ○ Large: supports up to 200,000 SNAT connections. ○ Super Large: supports up to 1,000,000 SNAT connections. <p> Note The size of a NAT gateway determines the maximum number of SNAT connections, but it does not affect the maximum number of DNAT connections.</p>
Parameter	<p>Enter a name for the NAT gateway.</p> <p>The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character but cannot start with <code>http://</code> or <code>https://</code>.</p>

24.3.3. Associate an EIP with the a NAT gateway

A NAT gateway functions as expected only after it is associated with a public IP address. After you create a NAT gateway, you can associate it with an elastic IP address (EIP).

Context

You can associate an EIP or a NAT service plan with a NAT gateway. However, you can choose only one of them for the NAT gateway. If you want to associate a NAT service plan with a NAT gateway, you must purchase a NAT service plan first. Then, you can configure the SNAT or DNAT feature for the NAT gateway. For more information, see [Create a NAT service plan](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateways page, find the NAT gateway, and choose  > **Bind Elastic IP Address** in the Actions column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters, and then click **OK**.

Parameter	Description
Usable EIP list	Select the EIP that is used to access the Internet.
VSwitch	<p>Select the VSwitch for which you want to add SNAT entries.</p> <p>After a VSwitch is selected, the system automatically adds an SNAT entry so that Alibaba cloud services in the VSwitch can access the Internet. You can also skip this step and manually add SNAT entries after you associate an EIP with the NAT gateway. For more information, see Create a SNAT entry.</p>

24.3.4. Create a DNAT entry

This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps public IP addresses to Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateways page, find the target NAT gateway, and click **Configure DNAT** in the Actions column.
4. On the DNAT Table page, click **Create DNAT Entry**.
5. On the Create DNAT Entry page, set the following parameters, and then click **OK**.

Parameter	Description
Public IP address	<p>Select an available public IP address.</p> <p> Note If a public IP address is already used in a SNAT entry, it cannot be used to create a DNAT entry.</p>
Private IP address	<p>Select the ECS instance that uses the DNAT entry to receive requests from the Internet. You can specify the private IP address of the target ECS instance in the following ways:</p> <ul style="list-style-type: none"> ◦ Auto Fill: select an ECS instance from the ECS instance or Elastic Network Interface (ENI) list. ◦ Manually Input: enter the private IP address of the target ECS instance. <p> Note The private IP address that you enter must fall in the range of the CIDR block or belong to an in-use ECS instance.</p>
Port Settings	<p>Select a DNAT mapping method:</p> <ul style="list-style-type: none"> ◦ All: This method uses IP mapping. All requests destined for the public IP address are forwarded to the target ECS instance. ◦ Specific Port: This method uses port mapping. The NAT gateway forwards requests from the specified protocol and port to the specified port of the target ECS instance. <p>After you select a specific port, specify Public Port (the external port for port mapping), Private Port (the internal port for port mapping), and IP Protocol (the protocol of the ports).</p>
Entry Name	<p>Enter a name for the DNAT entry.</p> <p>The name must be 2 to 128 characters in length and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.</p>

24.3.5. Create an SNAT entry

This topic describes how to create a Source Network Address Translation (SNAT) entry. You can create SNAT entries on the NAT gateway of a Virtual Private Cloud (VPC) network. Then, the Elastic Compute Service (ECS) instances without public IP addresses assigned in the VPC network can use the SNAT entries to access the Internet.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateways page, find the target NAT gateway and click **Configure SNAT** in the Actions column.
4. On the SNAT Table page, click **Create SNAT Entry**.

5. In the Create SNAT Entry dialog box, set the following parameters, and then click OK.

Parameter	Description
VSwitch Granularity	
VSwitch	<p>Select the VSwitch for which you want to create the SNAT entry in the associated VPC network. All ECS instances attached to the specified VSwitch can access the Internet by using the SNAT entry.</p> <p> Note SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned static public IP address, associated with an elastic IP address (EIP) or has a Destination Network Address Translation (DNAT) IP mapping configured. These ECS instances use the public IP addresses instead of the SNAT entries to access the Internet.</p>
VSwitch CIDR block	The CIDR block of the selected VSwitch.
Public IP address	<p>Select the public IP address that is used to access the Internet.</p> <p>You can select multiple public IP addresses to create an SNAT IP address pool.</p> <p> Note A public IP address that is already used in a DNAT entry cannot be used to create an SNAT entry.</p>
Entry name	<p>Enter a name for the SNAT entry.</p> <p>The name must be 2 to 128 characters in length and can contain digits, underscores (_), and hyphens (-). It must start with a letter or a Chinese character.</p>
ECS Granularity	
Available ECS Instances	<p>Select the ECS instance for which you want to create the SNAT entry in the associated VPC network.</p> <p>The selected ECS instance can access the Internet by using the specified public IP address. Make sure that the following conditions are met:</p> <ul style="list-style-type: none"> ◦ The ECS instance is in the running state. ◦ The ECS instance is not associated with an EIP or assigned a static public IP address.
ECS CIDR Block	Displays the CIDR block of the ECS instance.

Parameter	Description
Public IP address	<p>Select the public IP address that is used to access the Internet.</p> <p>You can select multiple public IP addresses to create an SNAT IP address pool.</p> <p>The maximum bandwidth for each public IP address in an SNAT IP address pool is 200 Mbit/s. To fully utilize the EIP bandwidth plan and avoid port conflicts caused by insufficient public IP addresses, note the following limits when you add public IP addresses to an SNAT entry:</p> <ul style="list-style-type: none"> ◦ If the maximum bandwidth of the EIP bandwidth plan is 1024 Mbit/s, add at least five public IP addresses to the SNAT entry. ◦ For each additional 200 Mbit/s of the peak bandwidth to the EIP bandwidth plan, at least one public IP address must be added to the SNAT entry. <p> Note A public IP address that is used in a DNAT entry cannot be used to create an SNAT entry.</p>
Entry Name	<p>Enter a name for the SNAT entry.</p> <p>The name must be 2 to 128 characters in length and can contain numbers, underscores (_), and hyphens (-). The name must start with a letter or Chinese character.</p>

24.4. Manage a NAT gateway

24.4.1. Sizes of NAT gateways

This topic describes the available sizes of Network Address Translation (NAT) gateways. The available sizes are Small, Middle, Large, and Super Large. The maximum number of SNAT connections and the number of SNAT connections per second (CPS) supported by a NAT gateway are determined by the size of NAT gateway. However, the size of a NAT gateway does not affect the performance of Destination Network Address Translation (DNAT).

Compare NAT gateway sizes

The following table lists different sizes of NAT gateways.

Size	Maximum number of SNAT connections	Number of SNAT CPS
Small	10,000	1,000
Middle	50,000	5,000
Large	200,000	10,000
Super Large	1,000,000	30,000

Limits

When you select a size for a NAT gateway, note the following limits:

- Cloud Monitor monitors only the maximum number of SNAT connections for NAT gateways. It does not monitor the number of new SNAT connections per second.
- The timeout of SNAT connections in a NAT gateway is 900 seconds.
- To avoid the timeout of SNAT connections caused by network congestion and Internet instability, make sure

that your applications support automatic reconnection. This ensures higher availability.

- NAT gateways do not support packet fragmentation.
- For the same destination public IP address and port, the number of EIPs associated with a NAT gateway determines the maximum number of connections. Each EIP associated with a NAT gateway supports up to 55,000 connections. If N EIPs are associated with the NAT gateway, the maximum number of connections that the NAT gateway supports is $N \times 55,000$.

24.4.2. Create a NAT gateway

A NAT gateway is an enterprise-class gateway that provides NAT proxy services. Before you configure SNAT and DNAT entries, you must create a NAT gateway.

Prerequisites

A VPC network is created. For more information, see *VPC User Guide* Create a VPC network in the Quick Start chapter in the VPC User Guide.

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the NAT Gateway page, click **Create NAT Gateway**.
3. Set the parameters for the NAT gateway based on the following information, and then click **Submit**.

Parameter	Description
Organization	The organization to which the NAT gateway belongs.
Resource set	The resource set to which the NAT gateway belongs.
Region	The region where the NAT gateway is deployed.
VPC	<p>The VPC network to which the NAT gateway belongs.</p> <p>If you cannot find the target VPC network in the list, perform the following operations:</p> <ul style="list-style-type: none"> ◦ Check whether the VPC network is already associated with a NAT gateway. Each VPC network can be associated with only one NAT gateway. ◦ Check whether the VPC network has a custom route entry with the destination CIDR block set to 0.0.0.0/0. If such a custom route entry exists, delete it. ◦ Check whether the RAM user is authorized to access the VPC network. If the RAM user is not authorized, contact your Alibaba Cloud account owner to grant permissions.
Specification	<p>Select the size of the NAT gateway. Valid values:</p> <ul style="list-style-type: none"> ◦ Small: supports up to 10,000 SNAT connections. ◦ Medium: supports up to 50,000 SNAT connections. ◦ Large: supports up to 200,000 SNAT connections. ◦ Super Large: supports up to 1,000,000 SNAT connections. <p> Note The size of a NAT gateway determines the maximum number of SNAT connections, but it does not affect the maximum number of DNAT connections.</p>

Parameter	Description
Parameter	<p>Enter a name for the NAT gateway.</p> <p>The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), hyphens (-), periods (.), colons (:), and commas (,). It must start with a letter or Chinese character but cannot start with <code>http://</code> or <code>https://</code>.</p>

24.4.3. Modify a NAT gateway

This topic describes how to modify the name and description of a NAT gateway.

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateway page, find the NAT gateway that you want to manage, and then click **Manage** in the **Actions** column.
4. On the **NAT Gateway Details** tab, click **Edit** next to the name. In the dialog box that appears, enter a new name for the NAT gateway, and then click **OK**. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
5. Click **Edit** next to the description. In the dialog box that appears, enter a new description, and then click **OK**. The description must be 2 to 256 characters in length. It cannot start with `http://` or `https://`.

24.4.4. Delete a NAT gateway

You can delete NAT gateways that are billed on a pay-as-you-go basis. You cannot delete subscription NAT gateways.

Prerequisites

Before you delete a NAT gateway, make sure that the following conditions are met:

- The NAT gateway is not associated with an EIP. If the NAT gateway is associated with an EIP, disassociate the EIP first. For more information, see [Disassociate EIPs from a NAT gateway](#).
- The DNAT table is empty. If the DNAT table contains DNAT entries, delete these entries first. For more information, see [Delete a DNAT entry](#).
- The SNAT table is empty. If the DNAT table contains SNAT entries, delete these entries first. For more information, see [Delete a SNAT entry](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click  > **Delete** in the **Actions** column.
4. In the dialog box that appears, click **OK**.

 **Note** If you select **Delete (Delete NAT gateway and resources)**, the DNAT and SNAT entries of the NAT gateway are deleted automatically. The EIP associated with the NAT gateway is also disassociated.

24.5. Manage EIPs

24.5.1. Associate an EIP with a NAT gateway

This topic describes how to associate an EIP with a NAT gateway. NAT gateways must be associated with EIPs so that they can work as expected. After you create a NAT gateway, you can associate it with an EIP.

Prerequisites

Before you associate an EIP with a NAT gateway, make sure that the following conditions are met:

- A NAT gateway is created. For more information, see [Create a NAT Gateway](#).
- An EIP is purchased. For more information, see the [Create an Elastic IP address](#) topic of the [Quick start](#) document in the *EIP user guide*.

Context

You can associate an EIP or a NAT service plan with a NAT gateway. However, you can choose only one of them for the NAT gateway. If you want to associate a NAT service plan with a NAT gateway, you must purchase a NAT service plan first. Then, you can configure the SNAT or DNAT feature for the NAT gateway. For more information, see [Create a NAT service plan](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the NAT gateway, and choose  > **Bind Elastic IP Address** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters, and then click **OK**.

Parameter	Description
Usable EIP list	Select the EIP that is used to access the Internet.
VSwitch	Select the VSwitch for which you want to add SNAT entries. After a VSwitch is selected, the system automatically adds an SNAT entry so that Alibaba cloud services in the VSwitch can access the Internet. You can also skip this step and manually add SNAT entries after you associate an EIP with the NAT gateway. For more information, see Create a SNAT entry .

24.5.2. Disassociate an EIP from a NAT gateway

If your NAT gateway does not need to communicate with the Internet, you can disassociate the EIP from the NAT gateway.

Prerequisites

The EIP that you want to disassociate is not used in any SNAT or DNAT entries.

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the NAT gateway, and choose  > **Unbind Elastic IP Address** in the **Actions** column.
4. In the **Unbind Elastic IP Address** dialog box, select the EIP that you want to disassociate, and click **OK**.

24.6. Manage a DNAT table

24.6.1. DNAT table overview

NAT Gateway supports the Destination Network Address Translation (DNAT) feature. You can create DNAT entries to map public IP addresses to ECS instances in a Virtual Private Cloud (VPC) network. This way, the ECS instances can receive requests from the Internet.

DNAT entries

You can configure port mapping when you create a DNAT entry. After the DNAT entry is created, requests destined for the specified public IP address are forwarded to the ECS instances within a VPC network based on the port mapping rule.

Each DNAT entry consists of the following elements:

- **Public IP address:** the EIP associated with the NAT gateway.
- **Private IP address:** the private IP address assigned to the ECS instance in the VPC network.
- **Public Port:** the external port where requests from the Internet are received.
- **Private Port:** the internal port to which the requests received on the external port are forwarded.
- **Protocol Type:** the protocol used by the ports.

Port mapping and IP mapping

The DNAT feature supports port mapping and IP mapping:

- **Port mapping**

After port mapping is configured, a NAT gateway forwards requests destined for a public IP address to the specified ECS instance based on the specified protocol and ports.

DNAT entry	Public IP address	Public port	Private IP address	Private port	Protocol type
Entry 1	139.224.xx.xx	80	192.168.x.x	80	TCP
Entry 2	139.224.xx.xx	8080	192.168.x.x	8000	UDP

Entry 1: The NAT gateway forwards requests destined for TCP port 80 of ECS instance 139.244.xx.xx to TCP port 80 of ECS instance 192.168.x.x.

Entry 2: The NAT gateway forwards requests destined for UDP port 8080 of ECS instance 139.224.xx.xx to UDP port 8000 of ECS instance 192.168.x.x.

- **IP mapping**

After IP mapping is configured, a NAT gateway forwards all requests destined for a public IP address to the specified ECS instance.

DNAT entry	Public IP address	Public port	Private IP address	Private port	Protocol type
Entry 3	139.224.xx.xx	Any	192.168.x.x	Any	Any

Entry 3: The NAT gateway forwards requests destined for ECS instance 139.224.xx.xx to ECS instance 192.168.x.x.

24.6.2. Create a DNAT entry

This topic describes how to create a Destination Network Address Translation (DNAT) entry. DNAT maps a public IP address to an Elastic Compute Service (ECS) instance in a Virtual Private Cloud (VPC) network. This allows the ECS instance to receive requests sent over the Internet. DNAT supports port mapping and IP mapping.

Prerequisites

A NAT gateway is created and associated with an elastic IP address (EIP). For more information, see [Create a NAT Gateway](#) and [Associate an EIP with a NAT Gateway](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT Table** page, click **Create DNAT Entry**.
5. On the **Create DNAT Entry** page, set the following parameters, and then click **OK**.

Parameter	Description
Public IP address	<p>Select an available public IP address.</p> <p> Note If a public IP address is already used in a SNAT entry, it cannot be used to create a DNAT entry.</p>
Private IP address	<p>Select the ECS instance that uses the DNAT entry to receive requests from the Internet. You can specify the private IP address of the target ECS instance in the following ways:</p> <ul style="list-style-type: none"> ◦ Auto Fill: select an ECS instance from the ECS instance or Elastic Network Interface (ENI) list. ◦ Manually Input: enter the private IP address of the target ECS instance. <p> Note The private IP address that you enter must fall in the range of the CIDR block or belong to an in-use ECS instance.</p>
Port Settings	<p>Select a DNAT mapping method:</p> <ul style="list-style-type: none"> ◦ All: This method uses IP mapping. All requests destined for the public IP address are forwarded to the target ECS instance. ◦ Specific Port: This method uses port mapping. The NAT gateway forwards requests from the specified protocol and port to the specified port of the target ECS instance. <p>After you select a specific port, specify Public Port (the external port for port mapping), Private Port (the internal port for port mapping), and IP Protocol (the protocol of the ports).</p>
Entry Name	<p>Enter a name for the DNAT entry.</p> <p>The name must be 2 to 128 characters in length and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.</p>

24.6.3. Modify a DNAT entry

This topic describes how to modify a Destination Network Address Translation (DNAT) entry. After you create a DNAT entry, you can modify the public IP address, private IP address, ports, and name of the DNAT entry.

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT table** page, find the target DNAT entry, and click **Edit** in the **Actions** column.

5. In the **Edit DNAT Entry** dialog box, change the public IP address, private IP address, ports, and name of the DNAT entry, and then click **OK**.

24.6.4. Delete a DNAT entry

This topic describes how to delete a Destination Network Address Translation (DNAT) entry. If you no longer need an Elastic Compute Service (ECS) instance to receive requests sent over the Internet, you can delete the DNAT entry of the ECS instance.

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the **NAT Gateways** page, find the target NAT gateway, and click **Configure DNAT** in the **Actions** column.
4. On the **DNAT** table page, find the target DNAT entry, and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

24.7. Manage an SNAT table

24.7.1. SNAT table overview

NAT Gateway supports Source Network Address Translation (SNAT). SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

SNAT entries

You can create SNAT entries in an SNAT table to allow ECS instances to access the Internet.

An SNAT entry consists of the following elements:

- **VSwitch or ECS instance:** the VSwitch or ECS instance that requires the SNAT proxy service.
- **Public IP address:** the public IP address used to access the Internet.

VSwitch granularity and ECS granularity

SNAT entries can be created based on the following granularity to enable ECS instances in a VPC network to access the Internet.

- **VSwitch granularity**

You can select the VSwitch granularity to create an SNAT entry. The NAT gateway provides proxy service for an ECS instance attached to the specified VSwitch by using a specified public IP address when the instance sends requests to the Internet. By default, all ECS instances attached to the VSwitch can use the specified public IP address to access the Internet.

 **Note** SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned static public IP address, associated with an elastic IP address (EIP) or has a Destination Network Address Translation (DNAT) IP mapping configured. These ECS instances use the public IP addresses instead of the SNAT entries to access the Internet.

- **ECS granularity**

If you select the ECS granularity to create an SNAT entry, the specified ECS instance uses the specified public IP address to access the Internet. The NAT gateway provides proxy service (SNAT) for a specified ECS instance by using a specified public IP address when the instance sends requests to the Internet.

24.7.2. Create an SNAT entry

This topic describes how to create a Source Network Address Translation (SNAT) entry. SNAT allows Elastic Compute Service (ECS) instances in a Virtual Private Cloud (VPC) network to access the Internet without using public IP addresses.

Prerequisites

Before you create an SNAT entry, make sure that the following requirements are met:

- A NAT gateway is created and associated with an elastic IP address (EIP). For more information, see [Create a NAT Gateway](#) and [Associate an EIP with a NAT Gateway](#).
- To create an SNAT entry with VSwitch granularity, make sure that the VSwitch is created and associated with the NAT gateway in a VPC network.
- To create an SNAT entry with ECS granularity, make sure that the ECS instance is created and associated with the NAT gateway in a VPC network.

Procedure

1. [Log on to the NAT Gateway console](#).
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateway page, find the target NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the **SNAT Table** page, click **Create SNAT Entry**.
5. In the **Create SNAT Entry** dialog box, set the following parameters, and click **OK**.

Parameter	Description
VSwitch Granularity	
VSwitch	<p>Select the VSwitch for which you want to create the SNAT entry in the associated VPC network. All ECS instances attached to the VSwitch can access the Internet by using the SNAT entry.</p> <p> Note SNAT entries do not take effect on ECS instances that are assigned public IP addresses. For example, an ECS instance may be assigned a static public IP address, associated with an EIP, or configured with a Destination Network Address Translation (DNAT) IP mapping. Such an ECS instance uses the public IP address instead of the SNAT entry to access the Internet.</p>
VSwitch CIDR Block	The CIDR block of the selected VSwitch.
Public IP Address	<p>Select the public IP address that is used to access the Internet.</p> <p>You can select more than one public IP address to form a SNAT IP address pool.</p> <p> Note A public IP address that is already used in a DNAT entry cannot be used to create a SNAT entry.</p>
Entry Name	<p>Enter a name for the SNAT entry.</p> <p>The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.</p>
ECS Granularity	

Parameter	Description
Available ECS Instances	<p>Select the ECS instance for which you want to create the SNAT entry in the associated VPC network.</p> <p>The selected ECS instance can access the Internet by using the specified public IP address. Make sure that the following conditions are met:</p> <ul style="list-style-type: none"> ◦ The ECS instance is in the Running state. ◦ The ECS instance is not associated with an EIP or assigned a static public IP address.
ECS CIDR Block	The CIDR block of the ECS instance.
Public IP Address	<p>Select the public IP address that is used to access the Internet.</p> <p>You can select more than one public IP address to form a SNAT IP address pool.</p> <p>The maximum bandwidth for each public IP address in a SNAT IP address pool is 200 Mbit/s. To fully utilize the EIP bandwidth plan and avoid port conflicts caused by insufficient public IP addresses, add public IP addresses to the SNAT IP address pool based on the following rules:</p> <ul style="list-style-type: none"> ◦ If the maximum bandwidth of the EIP bandwidth plan is 1,024 Mbit/s, add at least five public IP addresses to the SNAT IP address pool. ◦ For each additional 200 Mbit/s added to the maximum bandwidth of the EIP bandwidth plan, add at least one public IP address to the SNAT IP address pool. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note A public IP address that is already used in a DNAT entry cannot be used to create a SNAT entry.</p> </div>
Entry Name	<p>Enter a name for the SNAT entry.</p> <p>The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.</p>

24.7.3. Modify an SNAT entry

This topic describes how to modify a Source Network Address Translation (SNAT) entry. After you create an SNAT entry, you can modify the public IP address and name of the SNAT entry.

Procedure

1. Log on to the NAT Gateway console.
2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateway page, find the target NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the SNAT table page, find the target SNAT entry, and click **Edit** in the **Actions** column.
5. In the **Edit SNAT Entry** dialog box, change the public IP address and name of the SNAT entry, and then click **OK**.

24.7.4. Delete a SNAT entry

This topic describes how to delete a Source Network Address Translation (SNAT) entry. You can delete the SNAT entry if the Elastic Compute Service (ECS) instances without public IP addresses in a Virtual Private Cloud (VPC) network no longer need the SNAT service to access the Internet.

Procedure

1. Log on to the NAT Gateway console.

2. In the top navigation bar, select the region where the NAT gateway is deployed.
3. On the NAT Gateway page, find the target NAT gateway and click **Configure SNAT** in the **Actions** column.
4. On the SNAT table page, find the target SNAT entry, and click **Remove** in the **Actions** column.
5. In the dialog box that appears, click **OK**.

24.8. NAT service plan

24.8.1. Create a NAT service plan

You can associate an elastic IP address (EIP) or a NAT service plan to a NAT gateway. However, you can choose only one of them for the NAT gateway. If you want to associate a NAT service plan with the NAT gateway, you must create a NAT service plan first. Then, you can configure SNAT or DNAT for the NAT gateway. A NAT service plan consists of public IP addresses and Internet bandwidth.

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateways** page, find the target NAT gateway and choose **Purchase NAT Bandwidth Package** in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click **Purchase**.
4. On the **NAT Bandwidth Package** page, set the following parameters, and click **Submit**.

Parameter	Description
Region	Indicates the region for which the NAT service plan is purchased.
Billing methods	Select the billing method of the NAT service plan. Only By Bandwidth is supported.
Bandwidth (Mbit/s)	Enter a bandwidth value for the NAT service plan that you want to purchase. The maximum value is 5000 Mbit/s.
Name	Enter a name for the NAT service plan. The name must be 2 to 128 characters in length and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
Description	Enter a description for the NAT service plan. The description must be 2 to 256 characters in length and cannot start with <code>http://</code> or <code>https://</code> .
Quantity	Enter the number of NAT bandwidth plans that you want to purchase.

24.8.2. Modify the bandwidth of a NAT service plan

This topic describes how to modify the bandwidth of a NAT bandwidth plan. The modification takes effect immediately.

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.

3. On the **Bandwidth Package Details** page, click the target NAT service plan, and then choose **Modify Bandwidth**.
4. On the **Modify Bandwidth** page, modify the bandwidth, and then click **Submit**. Each NAT bandwidth plan supports a maximum of 5,000 Mbit/s in bandwidth.

24.8.3. Add an IP address

This topic describes how to add IP addresses to a NAT service plan. The added IP addresses can be used to create Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) rules.

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click the target NAT service plan, and then choose **Add IP Address**.
4. On the **Modify IP Addresses** page, enter the number of IP addresses to be added, and then click **Submit**.

24.8.4. Release an IP address

This topic describes how to release IP addresses in a NAT service plan. The NAT service plan must contain at least one IP address.

Prerequisites

Before you release an IP address in the NAT service plan, make sure that the IP address is not used in Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) entries. If the IP address is used in an SNAT or DNAT entry, delete the SNAT or DNAT entry first. For more information, see [Delete a DNAT entry](#) and [Delete a SNAT entry](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.
3. On the **Bandwidth Package Details** page, click the target NAT service plan.
4. In the **Public IP List** section, find the target IP address, and click **Release** in the **Actions** column.
5. In the **Release IP** dialog box, click **OK**.

24.8.5. Delete a NAT service plan

This topic describes how to delete a service plan.

Prerequisites

Before you start, make sure that the following requirements are met:

- Delete the IP addresses that are used in Destination Network Address Translation (DNAT) entries. For more information, see [Delete a DNAT entry](#).
- Delete the IP addresses that are used for Source Network Address Translation (SNAT) entries. For more information, see [Delete a SNAT entry](#).

Procedure

1. [Log on to the NAT Gateway console](#).
2. On the **NAT Gateways** page, find the target NAT gateway and click the ID of the NAT service plan in the **Internet Shared Bandwidth** column.

3. On the **Bandwidth Package Details** page, find the target NAT service plan and click **Delete**.
4. In the **Delete Shared Internet Shared Bandwidth** dialog box, click **OK**.

24.9. Anti-DDoS Basic

Distributed Denial of Service (DDoS) attack is a malicious network attack against the target system, which can make the attacked network inaccessible. Alibaba Cloud provides up to 5 Gbit/s of basic anti-DDoS protection for NAT Gateway, which can efficiently prevent DDoS attack.

How Anti-DDoS Basic works

After you enable Anti-DDoS Basic, all traffic from the Internet must first pass through Alibaba Cloud Security before arriving at NAT Gateway. Anti-DDoS Basic scrubs and filters common DDoS attacks at Alibaba Cloud Security. Anti-DDoS Basic protects your services against attacks such as SYN flood, UDP flood, ACK flood, ICMP flood, and DNS Query flood.

Anti-DDoS Basic sets the scrubbing threshold and black hole triggering threshold based on the EIP bandwidth of NAT Gateway. When the inbound traffic reaches the threshold, scrubbing or blackholing is triggered:

- **Scrubbing:** When the attack traffic from the Internet exceeds the scrubbing threshold or matches certain attack traffic pattern, Alibaba Cloud Security starts scrubbing the attack traffic. The scrubbing includes packet filtering, bandwidth capping, and traffic throttling.
- **Blackholing:** When the attack traffic from the Internet exceeds the black hole triggering threshold, blackholing is triggered and all inbound traffic is dropped.

Scrubbing threshold

The thresholds for triggering traffic scrubbing and blackholing on NAT Gateway are calculated as described in the following table:

EIP bandwidth	Traffic scrubbing threshold (bits/s)	Traffic scrubbing threshold (packets/s)	Default black hole triggering threshold
Lower than or equal to 800 Mbit/s	800Mbps	120,000	1.5 Gbps
Higher than 800 Mbit/s	Predefined bandwidth	Predefined bandwidth × 150	Predefined bandwidth × 2

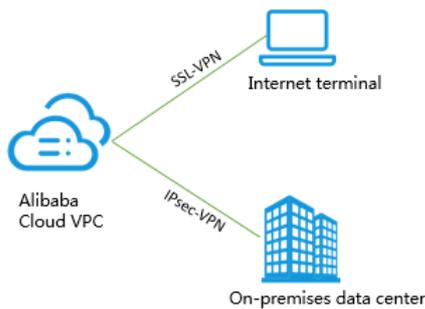
If the EIP bandwidth is 1,000 Mbit/s, the traffic scrubbing threshold (bits/s) is 1,000 Mbit/s, the traffic scrubbing threshold (packets/s) is 150,000 and the default blackholing threshold is 2 Gbit/s.

25.VPN Gateway

25.1. What is VPN Gateway?

VPN Gateway is an Internet-based service that allows you to connect enterprise data centers, office networks, or Internet-facing terminals to Alibaba Cloud Virtual Private Cloud (VPC) networks through secure and reliable connections. VPN Gateway supports both IPsec-VPN connections and SSL-VPN connections.

 **Note** The Alibaba Cloud VPN Gateway service complies with the local regulations and policies. VPN Gateway does not provide Internet access services.



Features

VPN Gateway supports the following features:

- IPsec-VPN

Route-based IPsec-VPN allows you to route network traffic in multiple ways, and also facilitates the configuration and maintenance of VPN policies.

You can use IPsec-VPN to connect an on-premises data center to a VPC network or connect two VPC networks. IPsec-VPN supports the IKEv1 and IKEv2 protocols. Any devices that support these two protocols can connect to Alibaba Cloud VPN Gateway, such as devices manufactured by Huawei, H3C, Hillstone, Sangfor, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.

- SSL-VPN

SSL-VPN is implemented based on the OpenVPN framework. You can create an SSL-VPN connection to connect a remote client to applications and services deployed in a VPC network. After you deploy your applications or services, you only need to import the certificate to the client to initiate a connection.

Benefits

VPN Gateway offers the following benefits:

- **High security:** You can use the IKE and IPsec protocols to encrypt data for secure and reliable data transmission.
- **High availability:** VPN Gateway adopts the hot-standby architecture to achieve failover within a few seconds, session persistence, and zero service downtime.
- **Cost-effectiveness:** The encrypted Internet connections provided by VPN Gateway are more cost-effective than leased lines.
- **Ease of use:** VPN Gateway is a ready-to-use service. VPN gateways start to work immediately after they are deployed.

25.2. Log on to the VPN Gateway console

This topic describes how to use Google Chrome to log on to the VPN Gateway console in Apsara Stack Cloud Management (ASCM).

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > VPN Gateway**.

25.3. Get started with IPsec-VPN

25.3.1. Connect on-premises data centers to VPC networks

This topic describes how to create IPsec-VPN connections on VPN gateways to connect an on-premises data center to a Virtual Private Cloud (VPC) network.

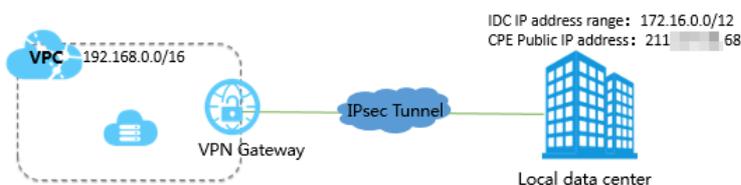
Prerequisites

Before you start, make sure that the following requirements are met:

- Check the gateway device in the on-premises data center. Alibaba Cloud VPN gateways support the standard IKEv1 and IKEv2 protocols. Any gateway device that supports these two protocols can connect to Alibaba Cloud VPN gateways, such as gateway devices manufactured by Huawei, H3C, Hillstone, Sangfor, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.
- Make sure that you have set a static public IP address for the gateway device in the on-premises data center.
- The CIDR block of the on-premises data center must not overlap with that of the VPC network.

Context

For example, a company creates a VPC network on Alibaba Cloud. The CIDR block of the VPC network is 192.168.0.0/16. The CIDR block of the on-premises data center is 172.16.0.0/12. The static public IP address for the gateway device in the on-premises data center is 211.xx.xx.68. To meet business requirements, the company needs to connect the on-premises data center to the VPC network.



The preceding figure displays that the on-premises data center is connected to the VPC network through IPsec-VPN. Cloud resources can be shared with on-premises data centers.

Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway:

1. **Log on to the VPN Gateway console.**
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, set the following parameters for the VPN gateway, and then click **Submit**.
 - **Organization:** Select the organization to which the VPN gateway belongs.
 - **Resource Set:** Select the resource set to which the VPN gateway belongs.
 - **Region:** Select the region where you want to deploy the VPN gateway.

 **Note** Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

- **Name:** Enter a name for the VPN gateway.
The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
- **VPC:** Select the VPC network to be associated with the VPN gateway.
- **Bandwidth:** Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
- **IPsec-VPN:** Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Enable**.
After IPsec-VPN is enabled, you can create IPsec-VPN connections between an on-premises data center and a VPC network, or between two VPC networks.
- **SSL-VPN:** Specify whether to enable SSL-VPN. In this example, select **Disable**.
SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a customer gateway.
- **SSL Connections:** Specify the maximum number of concurrent SSL connections that the VPN gateway supports.

 **Note** This parameter is available only after SSL-VPN is enabled.

5. Go to the **VPN Gateways** page to view the newly created VPN gateway. The newly created VPN gateway is in the **Preparing** state. Its status changes to **Normal** after about two minutes. The **Normal** state indicates that the VPN gateway is initialized and ready for use.

 **Note** It takes about one to five minutes to create a VPN gateway.

Step 2: Create a customer gateway

Take the following steps to create a customer gateway.

1. In the left-side navigation pane, choose **VPN > Customer Gateways**.
2. Select the region where you want to deploy the customer gateway
3. On the **Customer Gateways** page, click **Create Customer Gateway**.
4. On the **Create Customer Gateway** page, set the following parameters, and click **Submit**.
 - **Organization:** Select the organization to which the customer gateway belongs.

- **Resource Set:** Select the resource set to which the customer gateway belongs.
- **Region:** Select the region where you want to deploy the customer gateway.

 **Note** Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.

- **Zone:** Select the zone where you want to deploy the customer gateway.
- **Name:** Enter a name for the customer gateway.
The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with `http://` or `https://`.
- **IP Address:** Enter the public IP address of the gateway device in the on-premises data center that is to be connected to the VPC network. In this example, enter `211.xx.xx.68`.
- **Description:** Enter a description for the customer gateway.
The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`.

Step 3: Create an IPsec-VPN connection

Take the following steps to create an IPsec-VPN connection:

1. In the left-side navigation pane, choose **VPN > IPsec Connections**.
2. Select the region where you want to create an IPsec-VPN connection.
3. On the IPsec Connections page, click **Create IPsec Connection**.
4. On the **Create IPsec Connection** page, set the following parameters for the IPsec-VPN connection, and click **Submit**.
 - **Organization:** Select the organization to which the IPsec-VPN connection belongs.
 - **Resource Set:** Select the resource set to which the IPsec-VPN connection belongs.
 - **Region:** Select the region where the IPsec-VPN connection is established.
 - **Zone:** Select the zone where the IPsec-VPN connection is established.
 - **Name:** Enter a name for the IPsec-VPN connection.
 - **VPN Gateway:** Select a VPN gateway.
 - **Customer Gateway:** Select the customer gateway to be connected through the IPsec-VPN connection.
 - **Source CIDR Block:** Enter the CIDR block of the VPC network with which the selected VPN gateway is associated. In this example, enter `192.168.0.0/16`.
 - **Destination CIDR Block:** Enter the CIDR block of the on-premises data center. In this example, enter `172.16.0.0/12`.
 - **Immediate Effect:** Specify whether to start connection negotiations immediately.
 - **Yes:** negotiate immediately after the configuration is complete.
 - **No:** negotiate when traffic is detected in the IPsec-VPN connection.
 - **Pre-shared Key:** Enter the pre-shared key. The pre-shared key must be the same as that of the gateway device deployed in the on-premises data center.
Use the default settings for other parameters.

Step 4: Load the configurations of the IPsec-VPN connection to the customer gateway device

Take the following steps to load the configurations of the IPsec-VPN connection to the customer gateway device:

1. In the left-side navigation pane, choose **VPN > IPsec Connections**.

2. Select the region where the IPsec-VPN connection is established.
3. On the IPsec Connections page, find the target IPsec-VPN connection, and then choose **More > Download Configuration** in the **Actions** column.
4. Load the configurations of the IPsec-VPN connection to the customer gateway device by following the instructions described in . For more information about how to configure customer gateways, consult the manufacturers of the gateway devices. **RemotSubnet** and **LocalSubnet** in the downloaded configurations are opposite to **RemotSubnet** and **LocalSubnet** that you specify when you create an IPsec-VPN connection. For a VPN gateway, **RemotSubnet** refers to the CIDR block of the on-premises data center and **LocalSubnet** refers to the CIDR block of the VPC network. For a customer gateway, **LocalSubnet** refers to the CIDR block of the on-premises data center and **RemoteSubnet** refers to the CIDR block of the VPC network.

Step 5: Configure routes for the VPN gateway

Take the following steps to configure routes for the VPN gateway:

1. In the left-side navigation pane, choose **VPN > VPN Gateways**.
2. Select the region where the VPN gateway is deployed.
3. On the **VPN Gateways** page, find the target VPN gateway, and then click the instance ID in the **Instance ID/Name** column.
4. In the **Destination-based routing** tab, click **Add Route Entry**.
5. In the **Add Route Entry** dialog box, set the following parameters and click **OK**.
 - **Destination CIDR Block:** Enter the CIDR block of the on-premises data center. In this example, enter **172.16.0.0/12**.
 - **Next Hop Type:** Select **IPsec Connection**.
 - **Next Hop:** Select an IPsec instance.
 - **Publish to VPC:** Specify whether to automatically publish new route entries to the VPC route table. In this example, select **Yes**.
 - **Weight:** Select a weight. In this example, select **100**.

Step 6: Verify the settings

Log on to an Elastic Compute Service (ECS) instance that is not assigned a public IP address in the VPC network. Run the **ping** command to ping the private IP address of a server that resides in the on-premises data center, and test the connectivity.

25.4. Get started with SSL-VPN

25.4.1. Initiate a connection from a Linux client

This topic describes how to use SSL-VPN to connect a Linux client to a Virtual Private Cloud (VPC) network.

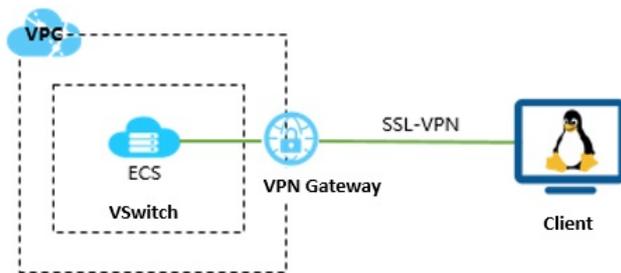
Prerequisites

Before you start, make sure that the following requirements are met:

- The CIDR block of the VPC network must not overlap with that of the client. Otherwise, the client cannot communicate with the VPC network.
- The client must be able to access the Internet.

Context

The following scenario is used as an example.



Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway.

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, specify the following parameters for the VPN gateway, and then click **Submit**.

- **Organization:** Select the organization to which the VPN gateway belongs.
- **Resource Set:** Select the resource set to which the VPN gateway belongs.
- **Region:** Select the region where the VPN gateway is deployed.

Note Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

- **Instance Name:** Enter a name for the VPN gateway.
The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character, and cannot start with `http://` or `https://`.
 - **VPC:** Select the VPC network to be associated with the VPN gateway.
 - **Bandwidth:** Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
 - **IPsec-VPN:** Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Disable**.
 - **SSL-VPN:** Specify whether to enable SSL-VPN for the VPN gateway. In this example, select **Enable**.
SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a gateway for the client.
 - **SSL Connections:** Specify the maximum number of concurrent SSL connections that the VPN gateway supports.
5. Go to the **VPN Gateways** page to view the newly created VPN gateway. The newly created VPN gateway is in the **Preparing** state. Its status changes to **Normal** after about two minutes. The **Normal** status indicates that the VPN gateway is initialized and ready for use.

Note It takes about one to five minutes to create a VPN gateway.

Step 2: Create an SSL server

Take the following steps to create an SSL server.

1. In the left-side navigation pane, choose **VPN > SSL Servers**.
2. Select the region where you want to create the SSL server.

3. On the **SSL Servers** page, click **Create SSL Server**.
4. On the **Create SSL Server** page, set the following parameters for the SSL server, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL server belongs.
 - **Resource Set:** Select the resource set to which the SSL server belongs.
 - **Region:** Select the region where you want to deploy the SSL server.
 - **Zone:** Select the zone where you want to deploy the SSL server.
 - **Name:** Enter a name for the SSL server.
 - **VPN Gateway:** Select a VPN gateway from the drop-down list.
 - **Source CIDR Block:** Enter the CIDR block of the VPC network. Click **+Add Local Network** to add more CIDR blocks. You can add the CIDR block of a VPC network, a VSwitch, and a local network.
 - **Client CIDR Block:** Enter the CIDR block of the client. The client connects to the SSL server from the specified CIDR block.
 - **Advanced Settings:** Specify whether to customize the advanced settings. In this topic, the default settings are used.

Step 3: Create and download an SSL client certificate

Take the following steps to create and download an SSL client certificate.

1. In the left-side navigation pane, choose **VPN > SSL Clients**.
2. Select the region where the client is created.
3. On the **SSL Clients** page, click **Create Client Certificate**.
4. On the **Create SSL Client Certificate** page, set the following parameters for the SSL client, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL client certificate belongs.
 - **Resource Set:** Select the resource set to which the SSL client certificate belongs.
 - **Region:** Select the region where you want to create the SSL client certificate.
 - **Zone:** Select the zone where you want to create the SSL client certificate.
 - **Name:** Enter a name for the SSL client certificate.
 - **VPN Gateway:** Select the VPN gateway to be associated with the SSL client certificate.
 - **SSL Server:** Select the SSL server to which you want to import the SSL client certificate.
5. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

Step 4: Configure the client

Take the following steps to configure the Linux client:

1. Run the following command to install the OpenVPN:

```
yum install -y openvpn
```

2. Decompress the client certificate package that you downloaded in Step 3 and copy the client certificate file to the `/etc/openvpn/conf/` folder where OpenVPN is installed.
3. Run the following command to launch OpenVPN:

```
openvpn --config /etc/openvpn/conf/config.ovpn --daemon
```

Step 5: Test the connectivity

To test the connectivity, run the `ping` command to ping the connected ECS instance in the VPC network.

Note Make sure that the security group rules of the ECS instance allow remote access from Linux clients.

25.4.2. Initiate a connection from a Windows client

This topic describes how to use SSL-VPN to connect a Windows client to a Virtual Private Cloud (VPC) network.

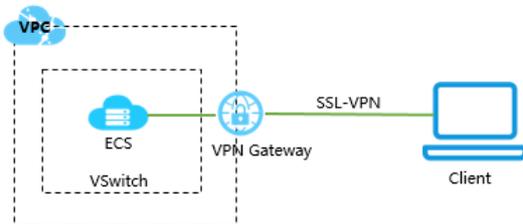
Prerequisites

Before you start, make sure that the following requirements are met:

- The CIDR block of the VPC network must not overlap with that of the client. Otherwise, the client cannot communicate with the VPC network.
- The client must be able to access the Internet.

Context

The following scenario is used as an example.



Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway.

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, specify the following parameters for the VPN gateway, and then click **Submit**.
 - **Organization:** Select the organization to which the VPN gateway belongs.
 - **Resource Set:** Select the resource set to which the VPN gateway belongs.
 - **Region:** Select the region where the VPN gateway is deployed.

Note Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

- **Instance Name:** Enter a name for the VPN gateway.

The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character, and cannot start with `http://` or `https://`.

- **VPC:** Select the VPC network to be associated with the VPN gateway.
- **Bandwidth:** Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
- **IPsec-VPN:** Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Disable**.
- **SSL-VPN:** Specify whether to enable SSL-VPN for the VPN gateway. In this example, select **Enable**.

SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a gateway for the client.

- **SSL Connections:** Specify the maximum number of concurrent SSL connections that the VPN gateway supports.
- 5. Go to the VPN Gateways page to view the newly created VPN gateway. The newly created VPN gateway is in the Preparing state. Its status changes to Normal after about two minutes. The Normal status indicates that the VPN gateway is initialized and ready for use.

 **Note** It takes about one to five minutes to create a VPN gateway.

Step 2: Create an SSL server

Take the following steps to create an SSL server.

1. In the left-side navigation pane, choose **VPN > SSL Servers**.
2. Select the region where you want to create the SSL server.
3. On the **SSL Servers** page, click **Create SSL Server**.
4. On the **Create SSL Server** page, set the following parameters for the SSL server, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL server belongs.
 - **Resource Set:** Select the resource set to which the SSL server belongs.
 - **Region:** Select the region where you want to deploy the SSL server.
 - **Zone:** Select the zone where you want to deploy the SSL server.
 - **Name:** Enter a name for the SSL server.
 - **VPN Gateway:** Select a VPN gateway from the drop-down list.
 - **Source CIDR Block:** Enter the CIDR block of the VPC network. Click **+Add Local Network** to add more CIDR blocks. You can add the CIDR block of a VPC network, a VSwitch, and a local network.
 - **Client CIDR Block:** Enter the CIDR block of the client. The client connects to the SSL server from the specified CIDR block.
 - **Advanced Settings:** Specify whether to customize the advanced settings. In this topic, the default settings are used.

Step 3: Create and download an SSL client certificate

Take the following steps to create and download an SSL client certificate.

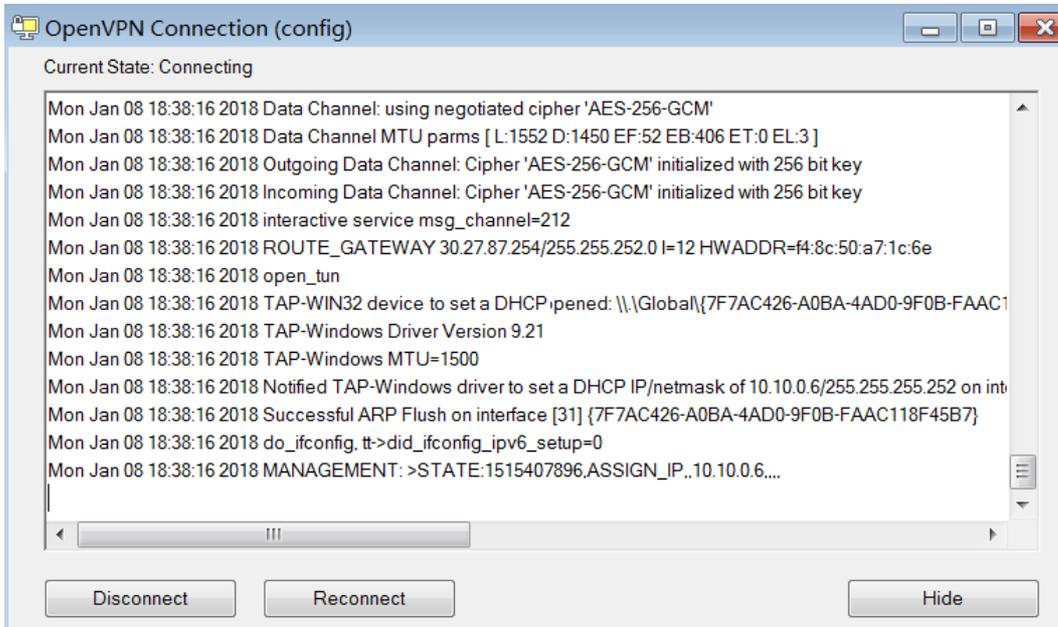
1. In the left-side navigation pane, choose **VPN > SSL Clients**.
2. Select the region where the client is created.
3. On the **SSL Clients** page, click **Create Client Certificate**.
4. On the **Create SSL Client Certificate** page, set the following parameters for the SSL client, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL client certificate belongs.
 - **Resource Set:** Select the resource set to which the SSL client certificate belongs.
 - **Region:** Select the region where you want to create the SSL client certificate.
 - **Zone:** Select the zone where you want to create the SSL client certificate.
 - **Name:** Enter a name for the SSL client certificate.
 - **VPN Gateway:** Select the VPN gateway to be associated with the SSL client certificate.
 - **SSL Server:** Select the SSL server to which you want to import the SSL client certificate.
5. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

Step 4: Configure the client

Take the following steps to configure the Windows client:

Notice You must run the client as an administrator.

1. Download and install OpenVPN.
2. Decompress the client certificate package that you downloaded in Step 3 and copy the client certificate file to the *config* folder where OpenVPN is installed.
3. Click **Connect** to initiate a connection.



Step 5: Test the connectivity

To test the connectivity, run the ping command to ping the connected ECS instance in the VPC network.

Note Make sure that the security group rules of the ECS instance allow remote access from Linux clients.

25.4.3. Initiate a connection from a macOS client

This topic describes how to use SSL-VPN to connect a macOS client to a Virtual Private Cloud (VPC) network.

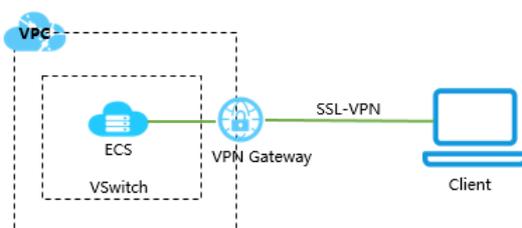
Prerequisites

Before you start, make sure that the following requirements are met:

- The CIDR block of the VPC network must not overlap with that of the client. Otherwise, the client cannot communicate with the VPC network.
- The client must be able to access the Internet.

Context

The following scenario is used as an example.



Step 1: Create a VPN gateway

Take the following steps to create a VPN gateway.

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN Gateway** page, specify the following parameters for the VPN gateway, and then click **Submit**.
 - **Organization:** Select the organization to which the VPN gateway belongs.
 - **Resource Set:** Select the resource set to which the VPN gateway belongs.
 - **Region:** Select the region where the VPN gateway is deployed.

 **Note** Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.

- **Instance Name:** Enter a name for the VPN gateway.
The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character, and cannot start with `http://` or `https://`.
 - **VPC:** Select the VPC network to be associated with the VPN gateway.
 - **Bandwidth:** Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
 - **IPsec-VPN:** Specify whether to enable IPsec-VPN for the VPN gateway. In this example, select **Disable**.
 - **SSL-VPN:** Specify whether to enable SSL-VPN for the VPN gateway. In this example, select **Enable**.
SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a gateway for the client.
 - **SSL Connections:** Specify the maximum number of concurrent SSL connections that the VPN gateway supports.
5. Go to the **VPN Gateways** page to view the newly created VPN gateway. The newly created VPN gateway is in the **Preparing** state. Its status changes to **Normal** after about two minutes. The **Normal** status indicates that the VPN gateway is initialized and ready for use.

 **Note** It takes about one to five minutes to create a VPN gateway.

Step 2: Create an SSL server

Take the following steps to create an SSL server.

1. In the left-side navigation pane, choose **VPN > SSL Servers**.
2. Select the region where you want to create the SSL server.
3. On the **SSL Servers** page, click **Create SSL Server**.
4. On the **Create SSL Server** page, set the following parameters for the SSL server, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL server belongs.
 - **Resource Set:** Select the resource set to which the SSL server belongs.
 - **Region:** Select the region where you want to deploy the SSL server.
 - **Zone:** Select the zone where you want to deploy the SSL server.
 - **Name:** Enter a name for the SSL server.
 - **VPN Gateway:** Select a VPN gateway from the drop-down list.
 - **Source CIDR Block:** Enter the CIDR block of the VPC network. Click **+Add Local Network** to add more CIDR

blocks. You can add the CIDR block of a VPC network, a VSwitch, and a local network.

- **Client CIDR Block:** Enter the CIDR block of the client. The client connects to the SSL server from the specified CIDR block.
- **Advanced Settings:** Specify whether to customize the advanced settings. In this topic, the default settings are used.

Step 3: Create and download an SSL client certificate

Take the following steps to create and download an SSL client certificate.

1. In the left-side navigation pane, choose **VPN > SSL Clients**.
2. Select the region where the client is created.
3. On the **SSL Clients** page, click **Create Client Certificate**.
4. On the **Create SSL Client Certificate** page, set the following parameters for the SSL client, and then click **Submit**.
 - **Organization:** Select the organization to which the SSL client certificate belongs.
 - **Resource Set:** Select the resource set to which the SSL client certificate belongs.
 - **Region:** Select the region where you want to create the SSL client certificate.
 - **Zone:** Select the zone where you want to create the SSL client certificate.
 - **Name:** Enter a name for the SSL client certificate.
 - **VPN Gateway:** Select the VPN gateway to be associated with the SSL client certificate.
 - **SSL Server:** Select the SSL server to which you want to import the SSL client certificate.
5. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

Step 4: Configure the client

Take the following steps to configure the macOS client:

1. Run the following command to install OpenVPN:

```
brew install openvpn
```

 **Note** If Homebrew is not installed, install Homebrew first.

2. Decompress the client certificate package that you downloaded in Step 3, copy the client certificate file to the folder where OpenVPN is installed, and initiate a connection.

- i. Back up the default configuration file.
- ii. Run the following command to delete the default configuration file:

```
rm /usr/local/etc/openvpn/*
```

- iii. Run the following command to copy the file to the configuration directory:

```
cp cert_location /usr/local/etc/openvpn/
```

In the preceding command, `cert_location` represents the path that stores the certificate downloaded in Step 3, for example, `/Users/example/Downloads/certs6.zip`.

- iv. Run the following command to decompress the client certificate package:

```
cd /usr/local/certificates
unzip /usr/local/etc/openvpn/certs6.zip
```

v. Run the following command to initiate a connection:

```
sudo /usr/local/opt/openssh/sbin/ssh-keygen --config /usr/local/etc/ssh/sshd_config
```

Step 5: Test the connectivity

To test the connectivity, run the ping command to ping the connected ECS instance in the VPC network.

 **Note** Make sure that the security group rules of the ECS instance allow remote access from Linux clients.

25.5. Manage a VPN Gateway

25.5.1. Create a VPN gateway

This topic describes how to create a VPN gateway. You must create a VPN gateway before you can use the IPsec-VPN and SSL-VPN services. After the VPN gateway is created, a public IP address is assigned to the VPN gateway.

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. On the **VPN Gateways** page, click **Create VPN Gateway**.
4. On the **Create VPN** page, specify the following parameters for the VPN gateway, and then click **Submit**.

Parameter	Description
Organization	Select the organization to which the VPN gateway belongs.
Resource Set	Select the resource set to which the VPN gateway belongs.
Region	Select the region where the VPN gateway is deployed. You can use IPsec-VPN to connect an on-premises data center to a VPC network or connect two VPC networks. Make sure that the VPC network and the VPN gateway associated with the VPC network are deployed in the same region.
Instance Name	Enter a name for the VPN gateway. The name must be 2 to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). It must start with a letter or Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
VPC	Select the VPC network to be associated with the VPN gateway.
Bandwidth	Specify the maximum bandwidth of the VPN gateway. The bandwidth is provided for data transfer over the Internet.
IPsec-VPN	Specify whether to enable IPsec-VPN for the VPN gateway. After IPsec-VPN is enabled, you can create IPsec-VPN connections between an on-premises data center and a VPC network, or between two VPC networks.
SSL-VPN	Specify whether to enable SSL-VPN for the VPN gateway. SSL-VPN connections are point-to-site connections. SSL-VPN allows you to connect a client to Alibaba Cloud without configuring a customer gateway.

Parameter	Description
SSL Connections	<p>Specify the maximum number of concurrent SSL connections that the VPN gateway supports.</p> <p> Note This parameter is available only after SSL-VPN is enabled.</p>

25.5.2. Modify a VPN gateway

This topic describes how to modify the name and description of a VPN gateway.

Prerequisites

A VPN gateway is created. For more information, see [Create a VPN gateway](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. Select the region where the VPN gateway is deployed.
4. On the **VPN Gateways** page, find the target VPN gateway, and click the  icon in the **Instance ID/Name** column. In the dialog box that appears, enter a new name and click **OK**. The name must be 2 to 100 characters in length, and can contain letters, digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
5. Click the  icon in the **Description** column. In the dialog box that appears, enter a new description and click **OK**. The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`.

25.5.3. Configure routes of a VPN Gateway

25.5.3.1. VPN Gateway route overview

After you create an IPsec-VPN connection, you must manually add a VPN Gateway route.

The route-based IPsec-VPN enables you to easily configure and maintain VPN policies, and provides flexible ways for routing traffic.

You can add the following two types of routes for a VPN Gateway:

- Policy-based routes.
- Destination-based routes.

Policy-based route

If a policy-based route is used, traffic is forwarded based on both the source IP address and the destination IP address.

For more information, see [Add policy-based routes](#).

 **Note** Policy-based routes take precedence over destination-based routes.

Destination-based route

If a destination-based route is used, traffic is forwarded based only on the destination IP address.

For more information, see [Add Destination-based routes](#).

25.5.3.2. Add a policy-based route entry

This topic describes how to add a policy-based route entry after an IPsec-VPN connection is created. Policy-based routing (PBR) is a technique that routes packets based on source and destination IP addresses.

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. Select the region where the VPN gateway is deployed.
4. On the **VPN Gateways** page, find the target VPN gateway and click the instance ID in the **Instance ID/Name** column.
5. Click the **Policy-based Routing** tab, and then click **Add Route Entry**.
6. In the **Add Route Entry** dialog box, set the following parameters and click **OK**.

Parameter	Description
Destination CIDR Block	The CIDR block that you want to access.
Source CIDR Block	The CIDR block of the VPC network.
Next Hop Type	Select IPsec Connection.
Next Hop	Select an IPsec instance to create an IPsec-VPN connection.
Publish to VPC	<p>Specify whether to automatically publish new route entries to the VPC route table.</p> <ul style="list-style-type: none"> ◦ Yes (Recommended): automatically publishes new route entries to the VPC route table. ◦ No: does not automatically publish new route entries to the VPC route table. <p> Note If you select No, you must manually publish new route entries to the VPC route table.</p>
Weight	<p>Select a weight. Valid values:</p> <ul style="list-style-type: none"> ◦ 100: indicates that the priority of the route entry is high. ◦ 0: indicates that the priority of the route entry is low. <p> Note If two policy-based route entries are configured with the same destination CIDR block, you cannot set the weights of both route entries to 100.</p>

25.5.3.3. Add a destination-based route entry

This topic describes how to manually add a destination-based route entry after an IPsec-VPN connection is created. Destination-based routing is a technique that routes packets to specified destination IP addresses.

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > VPN Gateways**.
3. Select the region where the VPN gateway is deployed.
4. On the **VPN Gateways** page, find the target VPN gateway and click the instance ID in the **Instance ID/Name** column.
5. In the **Destination-based routing** tab, click **Add Route Entry**.

6. In the Add Route Entry dialog box, set the following parameters and click **OK**.

Parameter	Description
Destination CIDR Block	The CIDR block that you want to access.
Next Hop Type	Select IPsec Connection.
Next Hop	Select an IPsec instance to create an IPsec-VPN connection.
Publish to VPC	Specify whether to automatically publish new route entries to the VPC route table. <ul style="list-style-type: none"> Yes (Recommended): automatically publishes new route entries to the VPC route table. No: does not automatically publish new route entries to the VPC route table. <p>Note If you select No, you must manually publish new route entries to the VPC route table.</p>
Weight	Select a weight. Valid values: <ul style="list-style-type: none"> 100: indicates that the priority of the route entry is high. 0: indicates that the priority of the route entry is low. <p>Note If two destination-based route entries are configure with the same destination CIDR block, you cannot set the weights of both route entries to 100.</p>

25.5.4. Delete a VPN gateway

This topic describes how to delete a VPN gateway. After you delete a VPN gateway, you can no longer use the VPN gateway to establish IPsec-VPN or SSL-VPN connections.

Context

Before you delete a VPN gateway, make sure that the following conditions are met:

- The IPsec-VPN connections on the VPN gateway are deleted. For more information, see [Delete an IPsec-VPN connection](#).
- The SSL server associated with the VPN gateway is deleted. For more information, see [Delete an SSL server](#).

Procedure

- Log on to the [VPN Gateway console](#).
- In the left-side navigation pane, choose **VPN > VPN Gateways**.
- Select the region where the VPN gateway is deployed.
- On the **VPN Gateways** page, find the target VPN gateway, and then click **Delete** in the **Actions** column.
- In the **Delete VPN Gateway** dialog box, click **OK**.

25.6. Manage a customer gateway

25.6.1. Create a customer gateway

This topic describes how to create a customer gateway when you use IPsec-VPN to connect a Virtual Private Cloud (VPC) network to an on-premises data center or connect two VPC networks. You can register and update the information of an on-premises gateway to Alibaba Cloud by creating a customer gateway, and then connect the customer gateway to the VPN gateway. A customer gateway can be connected to multiple VPN gateways.

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, click **VPN > Customer Gateways**.
3. Select the region where you want to deploy the customer gateway.

 **Note** Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.

4. On the **Customer Gateways** page, click **Create Customer Gateway**.
5. On the **Create Customer Gateway** page, set the following parameters, and click **Submit**.

Parameter	Description
Organization	Select the organization to which the customer gateway belongs.
Resource Set	Select the resource set to which the customer gateway belongs.
Region	Select the region where you want to deploy the customer gateway.  Note Make sure that the customer gateway and the VPN gateway to be connected are deployed in the same region.
Zone	Select the zone where you want to deploy the customer gateway
Name	Enter a name for the customer gateway. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
IP Address	Enter the static public IP address of the gateway device that is deployed in the on-premises data center.
Description	Enter a description for the customer gateway. The description must be 2 to 256 characters in length, and cannot start with <code>http://</code> / or <code>https://</code> .

25.6.2. Modify a customer gateway

This topic describes how to modify the name and description of a customer gateway.

Prerequisites

A customer gateway is created. For more information, see [Create a customer gateway](#).

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, click **VPN > Customer Gateways**.

3. Select the region where the customer gateway is deployed.
4. On the **Customer Gateways** page, find the target customer gateway, click the  icon in the **Instance ID** column. In the dialog box that appears, enter a name and click **OK**. The name must be 2 to 128 characters in length, and can contain digits, underscores (_), and hyphens (-). It must start with a letter or Chinese character.
5. Click the  icon in the **Description** column. In the dialog box that appears, enter a new description and click **OK**. The description must be 2 to 256 characters in length, and cannot start with `http://` or `https://`.

25.6.3. Delete a customer gateway

This topic describes how to delete a customer gateway.

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, click **VPN > Customer Gateways**.
3. Select the region where the customer gateway is deployed.
4. On the **Customer Gateways** page, find the target customer gateway, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

25.7. Configure SSL-VPN

25.7.1. Configuration overview

This topic describes how to use the SSL-VPN function to connect a remote client to a VPC.

Prerequisites

The following conditions must be met before you deploy a VPN Gateway:

- The client and the VPC are not using the same private CIDR block.
- The client is able to access the Internet.

Procedure

The following figure illustrates the work flow of how to connect a client to a VPC by using the SSL-VPN function.



1. **Create a VPN Gateway**
Create a VPN Gateway and enable the SSL-VPN function.
2. **Create an SSL server**
Specify the IP address range of the SSL server and the IP address range used by the client.
3. **Create a client certificate**

Create the client certificate according to server configurations, and then download the client certificate and configurations.

4. Configure the client

Download and install client VPN software in the client, load the client certificate and configurations, and initiate the connection.

5. Configure security groups

Make sure that the security group rules of ECS instances in the VPC allow remote access.

25.7.2. Manage an SSL server

25.7.2.1. Create an SSL Server

This topic describes how to create an SSL server. Before you use SSL-VPN to establish point-to-site connections, you must create an SSL server.

Prerequisites

A VPN gateway is created and SSL-VPN is enabled. For more information, see [Create a VPN gateway](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > SSL Servers**.
3. In the top navigation bar, select the region where you want to create the SSL server.
4. On the **SSL Servers** page, click **Create SSL Server**.
5. On the **Create SSL Server** page, configure the SSL server based on the following information, and then click **Submit**.

Parameter	Description
Organization	Select the organization to which the SSL server belongs.
Resource Set	Select the resource set to which the SSL server belongs.
Region	Select the region where the SSL server is to be deployed.
Zone	Select the zone where the SSL server is to be deployed.
Name	Enter a name for the SSL server. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
VPN Gateway	Select the VPN gateway to be associated with the SSL server. Make sure that SSL-VPN is enabled for the VPN gateway.

Parameter	Description
Source CIDR Block	<p>Enter the CIDR block that the client needs to access through the SSL-VPN connection. It can be the CIDR block of a VPC network, a VSwitch, an on-premises data center connected to a VPC through a leased line, or a cloud service such as ApsaraDB for RDS or Object Storage Service (OSS).</p> <p>Click + to add more CIDR blocks.</p> <p> Note The subnet mask of the specified server CIDR block must be 16 to 29 bits in length.</p>
Client CIDR Block	<p>Enter the CIDR block to be allocated to the virtual network interface of the client. Do not enter the CIDR block where the client resides. When the client accesses the server through an SSL-VPN connection, the VPN gateway selects an IP address from the specified CIDR block and assigns it to the client.</p> <p> Note Make sure the server CIDR block and the client CIDR block do not overlap.</p>
Advanced Settings	<p>Specify whether to customize the advanced settings.</p> <ul style="list-style-type: none"> ◦ Default: Use the default settings. ◦ Configure: Use custom settings. You can customize the following settings: <ul style="list-style-type: none"> ▪ Protocol: Select the protocol over which the SSL-VPN connection is established. Supported protocols are UDP and TCP. ▪ Port: Specify the port to which the SSL-VPN connection is established. <p>The following ports are not supported: 22, 2222, 22222, 9000, 9001, 9002, 7505, 80, 443, 53, 68, 123, 4510, 4560, 500, and 4500.</p> ▪ Encryption Algorithm: Select the encryption algorithm used by the SSL connection. Valid values: AES-128-CBC, AES-192-CBC, AES-256-CBC, and none. ▪ Compress: Specify whether to enable data compression.

25.7.2.2. Modify an SSL server

This topic describes how to modify the name, server CIDR block, client CIDR block, and advanced settings of an SSL server.

Prerequisites

An SSL server is created. For more information, see [Create an SSL server](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > SSL Servers**.
3. In the top navigation bar, select the region where you want to create the SSL server.
4. On the **SSL Servers** page, find the target SSL server, and then click **Edit** in the **Actions** column.
5. On the **Edit SSL Server** page, modify the name, server CIDR block, client CIDR block, and advanced settings of the SSL server, and then click **OK**.

25.7.2.3. Configure a routing group

This topic describes how to configure a routing group to control the inbound and outbound traffic of an Elastic Compute Service (ECS) instance after you create an IPsec-VPN connection.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose VPN > SSL Servers.
3. Select the region where the SSL server is deployed.
4. On the SSL Servers page, find the target SSL server, and then click **Configure Routing Group** in the **Actions** column.
5. In the **Configure Routing Group** dialog box, set the following parameters, and then click **Submit**.

Parameter	Description
Routing Group	Select the routing group to which you want to add the routing group rule.
Rule Direction	Select the direction in which the rule is applied. <ul style="list-style-type: none"> ◦ Outbound: from the ECS instances in the current routing group to other ECS instances on Alibaba Cloud or resources on the Internet. ◦ Inbound: from other ECS instances on Alibaba Cloud or resources on the Internet to the ECS instances in the current routing group.
Authorization Policy	Select an authorization policy. <ul style="list-style-type: none"> ◦ Allow: accept requests received on the specified ports. ◦ Deny: discard requests received on the specified ports without returning messages. If you specify different authorization policies for two routing group rules but the other settings are the same, the Deny rule prevails over the Allow rule.
Protocol Type	Select a protocol type.
Port Range	Select a port range for the routing group rule. The port range depends on the protocol type: <ul style="list-style-type: none"> ◦ When you set Protocol to All, this parameter displays -1/-1, which indicates all ports. You cannot specify a port range if you select this protocol type. ◦ When you set Protocol to TCP, you can specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 22/22 to indicate port 22. ◦ When Protocol is set to UDP, specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 3389/3389 to indicate port 3389. ◦ When Protocol is set to ICMP, this parameter displays -1/-1, which indicates all ports. You cannot set a port range if you select this protocol type. ◦ When Protocol is set to GRE, this parameter displays -1/-1, which indicates all ports. You cannot set a port range if you select this protocol type.
Priority	Set the priority of the rule. Valid values: 1 to 100. The default value is 1, which indicates the highest priority.
Authorization Type	Select the authorization type of the routing group rule. You can select only Address .
NIC type	Select a NIC type. <ul style="list-style-type: none"> ◦ Internal: Control inbound and outbound traffic within Alibaba Cloud. ◦ External: Control inbound and outbound traffic over the Internet.

Parameter	Description
Authorized IP Addresses	Select the CIDR blocks to be authorized. You can specify up to 10 CIDR blocks at a time.
Automatically Configure Routers	Specify whether to automatically configure routers.
Description	The description of the routing group rule. The description must be 2 to 256 characters in length, and cannot start with <code>http://</code> or <code>https://</code> . You can left this parameter empty.

25.7.2.4. Delete an SSL server

This topic describes how to delete an SSL server.

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > SSL Servers**.
3. In the top navigation bar, select the region where you want to create the SSL server.
4. On the **SSL Servers** page, find the target SSL server, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

25.7.3. Manage an SSL client certificate

25.7.3.1. Create an SSL client certificate

After you create an SSL server, you must create an SSL client certificate based on the configuration of the SSL server before you can establish an SSL-VPN connection.

Prerequisites

An SSL server is created. For more information, see [Create an SSL Server](#).

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > SSL Clients**.
3. Select the region where the SSL client is deployed.
4. On the **SSL Clients** page, click **Create Client Certificate**.
5. On the **Create SSL Client Certificate** page, configure the client certificate based on the following information, and then click **Submit**.

Parameter	Description
Organization	Select the organization to which the SSL client belongs.
Resource Set	Select the resource set to which the SSL client belongs.
Region	Select the region where the SSL client is deployed.
Zone	Select the zone where the SSL client is deployed.

Parameter	Description
Name	Enter a name for the SSL client certificate. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
VPN Gateway	Select the VPN gateway that is used to establish the SSL-VPN connection.
SSL Server	Select the SSL server to be associated with the SSL client.

25.7.3.2. Download an SSL client certificate

This topic describes how to download an SSL client certificate. Before you use an SSL client to initiate an SSL-VPN connection, you must import the SSL client certificate to the SSL client.

Prerequisites

An SSL client certificate is created. For information, see [Create an SSL client certificate](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > SSL Clients**.
3. Select the region where the SSL client is deployed.
4. On the **SSL Clients** page, find the target SSL client certificate, and then click **Download** in the **Actions** column.

25.7.3.3. Delete an SSL client certificate

This topic describes how to delete an SSL client certificate.

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > SSL Clients**.
3. Select the region where the SSL client is deployed.
4. On the **SSL Clients** page, find the target SSL client certificate, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

25.8. Configure IPsec-VPN connections

25.8.1. Configuration overview

This topic describes how to connect a VPC to an on-premises data center through IPsec-VPN.

Prerequisites

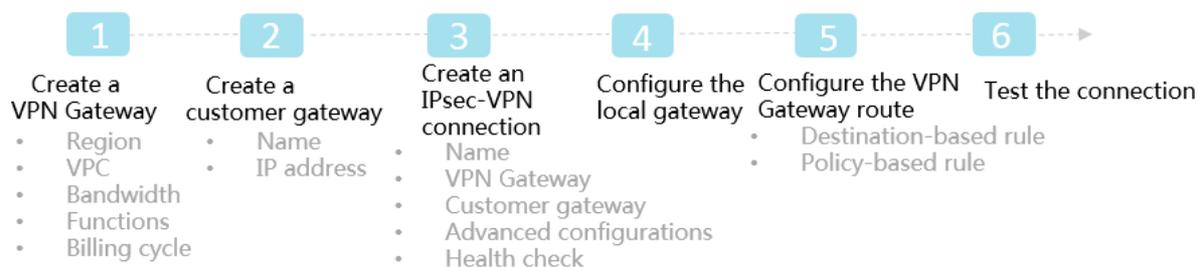
Before creating a site-to-site VPN connection, make sure the following conditions are met:

- The protocols IKEv1 and IKEv2 are supported by the gateway device of the on-premises data center.
IPsec-VPN supports IKEv1 and IKEv2 protocols. Devices that support these two protocols can connect to Alibaba Cloud VPN Gateway, including devices of Huawei, H3C, Hillstone, SANGFOR, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.
- A static public IP address is configured for the local gateway.

- The IP address ranges of the VPC and on-premises data center to be connected do not conflict with each other.

Procedure

The following figure shows the procedure of connecting a VPC to an on-premises data center through IPsec-VPN.



1. Create a VPN Gateway

Enable the IPsec-VPN function. Up to 10 IPsec-VPN connections can be established in a VPN Gateway.

2. Create a customer gateway

By creating a customer gateway, you can register the local gateway to Alibaba Cloud and connect the customer gateway to the VPN Gateway. A customer gateway can be connected to multiple VPN Gateways.

3. Create an IPsec connection

An IPsec connection is a VPN channel established between a VPN Gateway and a customer gateway. The encrypted communication between the VPN Gateway and the on-premises data center can be achieved only after the IPsec connection is established.

4. Configure the local gateway

You need to load the VPN Gateway configurations to the local gateway device. For more information, see [Local CPE configurations](#).

5. Configure the VPN Gateway route

You need to configure a route in the VPN Gateway and publish it to the VPC route table. For more information, see [VPN Gateway route overview](#).

6. Test the connection

Log on to an ECS instance (without a public IP address) in the connected VPC. ping the private IP address of a server in the on-premises data center to check whether the connection is established.

For more information, see [Establish a connection between a VPC and an on-premises data center](#).

25.8.2. Manage an IPsec-VPN connection

25.8.2.1. Create an IPsec-VPN connection

This topic describes how to create an IPsec-VPN connection. After you create a VPN gateway and a customer gateway, you can create an IPsec-VPN connection between the two gateways for encrypted data transmission.

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the **IPsec Connections** page, click **Create IPsec Connection**.
5. On the **Create IPsec Connection** page, configure the IPsec-VPN connection based on the following

information and click **Submit**.

Parameter	Description
Organization	Select the organization to which the IPsec-VPN connection belongs.
Resource Set	Select the resource set to which the IPsec-VPN connection belongs.
Region	Select the region where the IPsec-VPN connection is established.
Zone	Select the zone where the IPsec-VPN connection is established.
Name	Enter a name for the IPsec-VPN connection. The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), and underscores (_). It must start with a letter or Chinese character and cannot start with <code>http://</code> or <code>https://</code> .
VPN Gateway	Select the VPN gateway to be connected through the IPsec-VPN connection.
Customer Gateway	Select the customer gateway to be connected through the IPsec-VPN connection.
Source CIDR Block	Enter the CIDR block of the VPC network to be connected to the on-premises data center. The CIDR block is used during phase 2 negotiation. You can add more than one CIDR blocks only if IKEv2 is used.
Destination CIDR Block	Enter the CIDR block of the on-premises data center to be connected to the VPC network. This CIDR block is used during phase 2 negotiation. You can add more than one CIDR blocks only if IKEv2 is used.
Immediate Effect	Specify whether to start connection negotiations immediately. <ul style="list-style-type: none"> ◦ Yes: Negotiate immediately after the configuration is complete. ◦ No: Negotiate when traffic is detected in the IPsec-VPN tunnel.
Advanced Settings	Specify whether to customize the advanced settings. <ul style="list-style-type: none"> ◦ Default: Use default settings. ◦ Configure: Use custom settings.
Advanced configuration: IKE configuration	
Pre-shared Key	Enter the pre-shared key used for authentication between the VPN gateway and customer gateway. You can specify a key, or use the default key that is randomly generated by the system.
Version	The version of the IKE protocol. Select an IKE version. Compared with IKEv1, IKEv2 simplifies the process of Security Association (SA) negotiation and provides better support for scenarios where an SSL-VPN connection is established with multiple subnets. We recommend that you select IKEv2.
Negotiation Mode	The negotiation mode of IKEv1. <ul style="list-style-type: none"> ◦ Main: This mode offers higher security. ◦ Aggressive: This mode is faster than the main mode. Negotiations are more likely to succeed in this mode. <p>Connections negotiated in both modes ensure the same security level of data transmission.</p>

Parameter	Description
Encryption Algorithm	Select the encryption algorithm used during phase 1 negotiation. Supported algorithms are aes, aes192, aes256, des, and 3des.
Authentication Algorithm	Select the authentication algorithm used during phase 1 negotiation. Supported algorithms are sha1, and md5.
DH Group	Select the Diffie-Hellman key exchange algorithm used during phase 1 negotiation.
SA Life Cycle (Seconds)	Specify the lifecycle of the SA after phase 1 negotiation succeeds. Default value: 86,400.
LocalId	The ID of the VPN gateway used during phase 1 negotiation. The default value is the public IP address of the VPN gateway. If you set LocalId to a FQDN, we recommend that you set Negotiation Mode to Aggressive.
Remoteld	The ID of the customer gateway used during phase 1 negotiation. The default value is the public IP address of the customer gateway. If you set Remoteld to a FQDN, we recommend that you select set Negotiation Mode to Aggressive.
Advanced configuration: IPsec configuration	
Encryption Algorithm	Select the encryption algorithm used during phase 2 negotiation. Supported algorithms are aes, aes192, aes256, des, and 3des.
Authentication Algorithm	Select the authentication algorithm used during phase 2 negotiation. Supported algorithms are sha1, and md5.
DH Group	Select the Diffie-Hellman key exchange algorithm for phase 2 negotiations. <ul style="list-style-type: none"> ◦ If you select a Diffie-Hellman group, the perfect forward secrecy (PFS) feature is enabled and each negotiation requires a new key. Therefore, you must also enable PFS for the client. ◦ For clients that do not support PFS, select disabled.
SA Life Cycle (Seconds)	Specify the lifecycle of the SA after phase 2 negotiation succeeds. Default value: 86,400.

25.8.2.2. Modify an IPsec-VPN connection

This topic describes how to modify the name, advanced settings, and health check for an IPsec-VPN connection.

Prerequisites

An IPsec-VPN connection is created. For more information, see [Create an IPsec-VPN connection](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the IPsec Connections page, find the target IPsec-VPN connection and then click **Edit** in the **Actions** column.
5. In the **Modify IPsec Connections** dialog box, modify the name, advanced settings, and health check, and click **Submit**.

25.8.2.3. Download the configuration file of an IPsec-VPN connection

This topic describes how to download the configurations of an IPsec-VPN connection, and load the configurations to the customer gateway device after an IPsec-VPN connection is configured.

Prerequisites

An IPsec-VPN connection is created. For more information, see [Create an IPsec-VPN connection](#).

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > Download Configuration** in the **Actions** column.

 **Note** RemotSubnet and LocalSubnet in the downloaded configurations are opposite to RemotSubnet and LocalSubnet that you specify when you create an IPsec-VPN connection. For a VPN gateway, RemotSubnet refers to the CIDR block of the on-premises data center and LocalSubnet refers to the CIDR block of the VPC network. For a customer gateway, LocalSubnet refers to the CIDR block of the on-premises data center and RemoteSubnet refers to the CIDR block of the VPC network.

25.8.2.4. Configure a routing group

This topic describes how to configure routing groups to control the inbound and outbound traffic of ECS instances in a routing group after an IPsec-VPN connection is created.

Procedure

1. [Log on to the VPN Gateway console](#).
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the **IPsec Connections** page, find the target IPsec-VPN connection, and then choose **More > Configure Routing Group** in the **Actions** column.
5. In the **Configure Routing Group** dialog box, set the following parameters, and then click **Submit**.

Parameter	Description
Routing Group	Select the routing group to which you want to add the routing group rule.
Rule Direction	Select the direction in which the rule is applied. <ul style="list-style-type: none"> ◦ Outbound: from the ECS instances in the current routing group to other ECS instances on Alibaba Cloud or resources on the Internet. ◦ Inbound: from other ECS instances on Alibaba Cloud or resources on the Internet to the ECS instances in the current routing group.
Authorization Policy	Select an authorization policy. <ul style="list-style-type: none"> ◦ Allow: accept requests received on the specified ports. ◦ Deny: discard requests received on the specified ports without returning messages. If you specify different authorization policies for two routing group rules but the other settings are the same, the Deny rule prevails over the Allow rule.

Parameter	Description
Protocol Type	Select a protocol type.
Port Range	<p>Select a port range for the routing group rule. The port range depends on the protocol type:</p> <ul style="list-style-type: none"> When you set Protocol to All, this parameter displays -1/-1, which indicates all ports. You cannot specify a port range if you select this protocol type. When you set Protocol to TCP, you can specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 22/22 to indicate port 22. When Protocol is set to UDP, specify a port range in the <start port number>/<end port number> format. Valid port numbers: 1 to 65535. To specify a single port, set the start and end port numbers to the same value. For example, use 3389/3389 to indicate port 3389. When Protocol is set to ICMP, this parameter displays -1/-1, which indicates all ports. You cannot set a port range if you select this protocol type. When Protocol is set to GRE, this parameter displays -1/-1, which indicates all ports. You cannot set a port range if you select this protocol type.
Priority	Set the priority of the rule. Valid values: 1 to 100. The default value is 1, which indicates the highest priority.
Authorization Type	Select the authorization type of the routing group rule. You can select only Address.
NIC type	Select a NIC type. <ul style="list-style-type: none"> Internal: Control inbound and outbound traffic within Alibaba Cloud. External: Control inbound and outbound traffic over the Internet.
Authorized IP Addresses	Select the CIDR blocks to be authorized. You can specify up to 10 CIDR blocks at a time.
Automatically Configure Routers	Specify whether to automatically configure routers.
Description	The description of the routing group rule. The description must be 2 to 256 characters in length, and cannot start with <code>http://</code> or <code>https://</code> . You can left this parameter empty.

25.8.2.5. View IPsec-VPN connection logs

This topic describes how to view IPsec-VPN connection logs that are generated within the last 30 days to troubleshoot connection errors. You can query log data generated within 10 minutes.

Procedure

1. Log on to the VPN Gateway console.
2. In the left-side navigation pane, choose VPN > IPsec Connections.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the IPsec Connections page, find the target IPsec-VPN connection, and then choose More > View Logs in the Actions column.

5. In the IPsec Connection Logs dialog box, set the time range and query the logs.

25.8.2.6. Delete an IPsec-VPN connection

This topic describes how to delete an IPsec-VPN connection.

Procedure

1. [Log on to the VPN Gateway console.](#)
2. In the left-side navigation pane, choose **VPN > IPsec Connections**.
3. Select the region where you want to create the IPsec-VPN connection.
4. On the IPsec Connections page, find the target IPsec-VPN connection, and then click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

25.8.3. MTU notes

The maximum transmission unit (MTU) is the size (in bytes) of the largest packet supported by the network layer protocol (such as TCP), with headers and data included.

Network packets sent over IPsec tunnels are encrypted and then encapsulated in external packets for routing. Because an encapsulated internal packet itself must fit the MTU of the corresponding external packet, the MTU of the internal packet must be smaller.

Gateway MTU and system MTU

You must configure the MTU limit of the local VPN Gateway to not more than 1,400 bytes. We recommend that you set the MTU to 1,400 bytes.

For TCP traffic, the maximum length of data that can be carried by each packet segment can be negotiated by the sender and receiver when they communicate based on the maximum segment size (MSS).

26. Elastic IP Address

26.1. What is an EIP?

This topic provides an overview of Elastic IP Address. An elastic IP address (EIP) is a public IP address that you can purchase and hold as an independent resource. You can associate an EIP with an Elastic Compute Service (ECS) instance deployed in a virtual private cloud (VPC), an internal Server Load Balancer (SLB) instance deployed in a VPC, or a secondary elastic network interface (ENI) attached to a VPC. You can also associate an EIP with a NAT gateway, or a High-Availability Virtual IP Address (HAVIP).

An EIP is also a NAT IP address that is provisioned in a public-facing gateway of Alibaba Cloud and is mapped to the associated cloud resource with NAT. After an EIP is associated with a cloud resource, the cloud resource can connect to the Internet by using this EIP.

Differences between an EIP and a public IP address of an ECS instance

The following table compares the features of an EIP with those of a public IP address of an ECS instance.

Feature	EIP	ECS public IP address
Supported networks	VPC	VPC
Support for being held as an independent resource	Yes	No
Support for being associated with and disassociated from an ECS instance at any time	Yes	No
Support for being displayed in the ENI information of the associated ECS instance	The IP address is displayed in the ENI information if it is associated with the ENI in the cut-through mode	No

Benefits

EIPs have the following benefits:

- **Independent purchase and possession**
You can purchase and hold an EIP as an independent resource. EIPs are not bundled with other computing or storage resources.
- **Flexible association**
You can dissociate an EIP from a cloud resource and then release the EIP if the EIP is no longer needed.
- **Configurable network capabilities**
You can adjust the peak bandwidth of an EIP at any time. The bandwidth changes take effect immediately.

26.2. Log on to the EIP console

This topic provides an example of how to log on to the EIP console by using Google Chrome.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.

2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > Elastic IP Address**.

26.3. Quick start

26.3.1. Tutorial overview

This topic provides an overview of the tutorial that guides you in creating an EIP and associating an EIP with an ECS instance to allow the ECS instance to access the Internet.

This tutorial walks you through the following tasks:

1. **Create an Elastic IP address**

An EIP is a public IP address that you can purchase and hold as an independent resource. To get started, you must create an EIP.

2. **Bind an EIP to an ECS instance**

You can associate an EIP with an ECS instance deployed in a VPC to enable the ECS instance to connect to the Internet.

3. **Unbind an Elastic IP address from a cloud instance**

You can disassociate an ECS instance from an EIP when the ECS instance no longer requires access to the Internet.

4. **Release an Elastic IP address**

You can release an EIP if it is no longer needed.

26.3.2. Create a EIP

This topic describes how to create a EIP. An EIP is a public IP address that you can purchase and hold as an independent resource.

Procedure

1. **Log on to the EIP console.**
2. On the **Elastic IP Addresses** page, click **Create EIP**.
3. On the **Create Elastic IP Address** page, set the parameters and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the EIP belongs.
Resource Set	Select the resource set to which the EIP belongs.

Parameter	Description
Region	Select a region for the EIP. Make sure that the EIP and the cloud resource to be associated with the EIP are located in the same region.
Zone	Select the zone to which the EIP belongs.
Line Type	Select the line type of the EIP.
Network Type	Select the network type of communication for which the EIP will be used. <ul style="list-style-type: none"> Public Network: Use the EIP to enable communication with the Internet. Hybrid Cloud: Use the EIP to enable communication within a hybrid cloud. For example, if you need to establish network connections from an on-premises data center to the Internet by using source network address translation (SNAT) and destination network address translation (DNAT), you must select this type for your EIP.
Service IP	Specify a specific EIP. Make sure that the specified IP address is an IPv4 address and is not taken by another account. Otherwise, the EIP cannot be allocated. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note If you do not specify a service IP, a random EIP will be assigned to you by the system.</p> </div>
Peak Bandwidth	Specify the maximum bandwidth for the EIP. Unit: Mbit/s.

26.3.3. Associate an EIP with an ECS instance

This topic describes how to associate an EIP with an ECS instance deployed in a VPC. ECS instances that are associated with EIPs can communicate with the Internet.

Prerequisites

You have created an ECS instance. For more information, see the **Create an instance** topic of **Quick start** in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. [Log on to the EIP console](#).
2. Select the region of the EIP.
3. On the **Elastic IP Addresses** page, find the target EIP, and then click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select ECS Instance.

Parameter	Description
ECS Instance	<p>Select the ECS instance to be associated with the EIP.</p> <p>When you select an ECS instance, note the following points:</p> <ul style="list-style-type: none"> ◦ The ECS instance must be deployed in a VPC. ◦ The ECS instance must be running or stopped. ◦ An ECS instance can be associated with only one EIP. ◦ The ECS instance and the EIP must reside in the same region. ◦ The ECS instance is not associated with a public IP address or another EIP.

26.3.4. Disassociate an EIP from a cloud resource

This topic describes how to dissociate an EIP from a cloud resource when this cloud resource no longer needs to communicate with the Internet.

Procedure

1. [Log on to the EIP console.](#)
2. In the upper-left corner, select the region where your EIP is created.
3. On the Elastic IP Addresses page, find the target EIP, and click **Unbind** in the **Actions** column.
4. In the **Unbind Elastic IP Address** dialog box, click **OK**.

26.3.5. Release an EIP

This topic describes how to release an Elastic IP address (EIP).

Prerequisites

The EIP is disassociated from all instances. For more information, see [Disassociate an EIP from a cloud resource.](#)

Procedure

1. [Log on to the EIP console.](#)
2. In the upper-left corner, select the region where your EIP is created.
3. On the Elastic IP Addresses page, find the target EIP, move the mouse pointer over **More operations** in the **Actions** column, and then click **Release**.
4. In the **Release an EIP** dialog box, click **OK**.

26.4. Manage EIPs

26.4.1. Create a EIP

This topic describes how to create a EIP. An EIP is a public IP address that you can purchase and hold as an independent resource.

Procedure

1. [Log on to the EIP console.](#)
2. On the Elastic IP Addresses page, click **Create EIP**.
3. On the **Create Elastic IP Address** page, set the parameters and click **Submit**. The following table describes the configuration parameters.

Parameter	Description
Organization	Select the organization to which the EIP belongs.
Resource Set	Select the resource set to which the EIP belongs.
Region	Select a region for the EIP. Make sure that the EIP and the cloud resource to be associated with the EIP are located in the same region.
Zone	Select the zone to which the EIP belongs.
Line Type	Select the line type of the EIP.
Network Type	Select the network type of communication for which the EIP will be used. <ul style="list-style-type: none"> Public Network: Use the EIP to enable communication with the Internet. Hybrid Cloud: Use the EIP to enable communication within a hybrid cloud. For example, if you need to establish network connections from an on-premises data center to the Internet by using source network address translation (SNAT) and destination network address translation (DNAT), you must select this type for your EIP.
Service IP	Specify a specific EIP. Make sure that the specified IP address is an IPv4 address and is not taken by another account. Otherwise, the EIP cannot be allocated. <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note If you do not specify a service IP, a random EIP will be assigned to you by the system.</p> </div>
Peak Bandwidth	Specify the maximum bandwidth for the EIP. Unit: Mbit/s.

26.4.2. Bind an EIP to a cloud instance

26.4.2.1. Associate an EIP with an ECS instance

This topic describes how to associate an EIP with an ECS instance deployed in a VPC. ECS instances that are associated with EIPs can communicate with the Internet.

Prerequisites

You have created an ECS instance. For more information, see the [Create an instance](#) topic of *Quick start* in the *Apsara Stack Elastic Compute Service User Guide*.

Procedure

1. [Log on to the EIP console](#).
2. Select the region of the EIP.
3. On the **Elastic IP Addresses** page, find the target EIP, and then click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select ECS Instance.

Parameter	Description
ECS Instance	<p>Select the ECS instance to be associated with the EIP.</p> <p>When you select an ECS instance, note the following points:</p> <ul style="list-style-type: none"> ◦ The ECS instance must be deployed in a VPC. ◦ The ECS instance must be running or stopped. ◦ An ECS instance can be associated with only one EIP. ◦ The ECS instance and the EIP must reside in the same region. ◦ The ECS instance is not associated with a public IP address or another EIP.

26.4.2.2. Associate an EIP with an SLB instance

This topic describes how to associate an EIP with an SLB instance. After the association, the SLB instance can distribute requests from the Internet.

Prerequisites

You have created an SLB instance. For more information, see the **Create an SLB instance** topic of **Quick start** in the *Apsara Stack Server Load Balancer User Guide*.

Procedure

1. Log on to the [EIP console](#) [EIP console](#).
2. [Log on to the EIP console](#).
3. In the upper-left corner, select the region where your EIP is created.
4. On the **Elastic IP Addresses** page, find the EIP that you want to manage, and click **Bind** in the **Actions** column.
5. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select SLB Instance.
SLB Instance	<p>Select the SLB instance to be associated with the EIP.</p> <p>When you select an SLB instance, note the following points:</p> <ul style="list-style-type: none"> ◦ The SLB instance must be deployed in a VPC. ◦ The SLB instance and the EIP must reside in the same region. ◦ An SLB instance can be associated with only one EIP.

26.4.2.3. Associate an EIP with an HAVIP

This topic describes how to associate an EIP with an HAVIP. After the association, the HAVIP can be used to connect to the Internet.

Prerequisites

You have created an HAVIP. For more information, see the **Create an HAVIP** topic of **HAVIPs** in the *Apsara Stack Virtual Private Cloud User Guide*.

Procedure

1. Log on to the [EIP console](#) [EIP console](#).
2. [Log on to the EIP console](#).

- In the upper-left corner, select the region where your EIP is created.
- On the **Elastic IP Addresses** page, find the EIP that you want to manage, and click **Bind** in the **Actions** column.
- In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select HaVIP Address.
HaVIP Address	<p>Select the HAVIP to be associated with the EIP.</p> <p>When you select an HAVIP, note the following points:</p> <ul style="list-style-type: none"> The HAVIP and the EIP must reside in the same region. The HAVIP must be available or allocated. An HAVIP can be associated with only one EIP.

26.4.2.4. Associate an EIP with a NAT gateway

This topic describes how to associate an EIP with a NAT gateway. After the association, you can use the EIP to configure DNAT and SNAT entries.

Prerequisites

You have created a NAT gateway. For more information, see the **Create a NAT gateway** topic of **Quick start** in the *Apsara Stack NAT Gateway User Guide*.

Procedure

- Log on to the EIP console.**
- In the upper-left corner, select the region where your EIP is created.
- On the **Elastic IP Addresses** page, find the EIP that you want to manage, and click **Bind** in the **Actions** column.
- In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

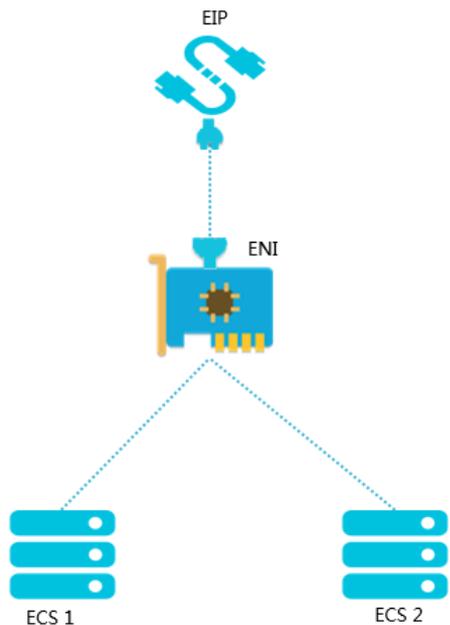
Parameter	Description
Instance Type	Select NAT Gateway.
NAT Gateway	<p>Select the NAT gateway to be associated with the EIP.</p> <p>When you select a NAT gateway, note the following points:</p> <ul style="list-style-type: none"> The NAT gateway and the EIP must reside in the same region. A NAT gateway can be associated with a maximum of 20 EIPs.

26.4.2.5. Bind an EIP to a secondary ENI

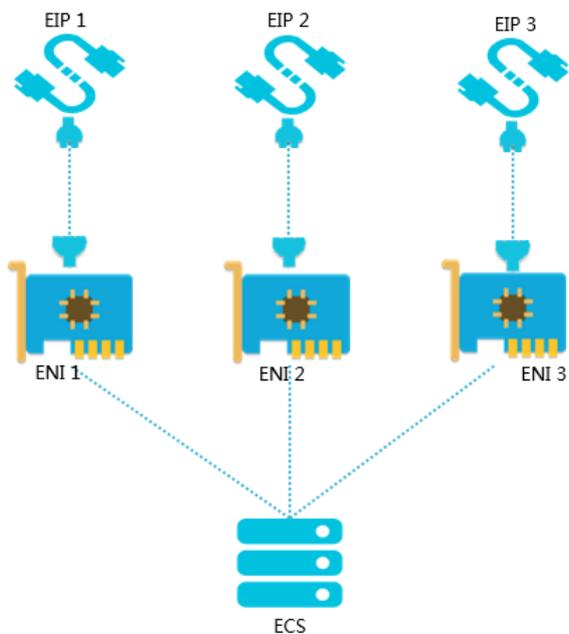
26.4.2.5.1. Overview

This topic provides an overview of EIP associations with ENIs. You can associate EIPs with ENIs to build a highly robust, flexible, and scalable IT solution. This also allows a cloud server to use multiple public IP addresses.

An ENI is assigned with a private IP address. After you associate an EIP with an ENI, the ENI has both a private IP address and a public IP address. When you migrate an ENI that is associated with an EIP from an ECS instance to another ECS instance, the public and private IP addresses are also migrated. This provides a highly reliable and available IP migration solution for cloud servers that use both public and private IP addresses.



To enable an ECS instance to use multiple public IP addresses, you can associate the ECS instance with multiple ENIs, and then associate each ENI with an EIP. You can use these public IP addresses along with security group rules to provide external services.



Association modes

You can associate an EIP with an ENI in one of the following two modes:

- NAT mode
- Cut-through mode

The following table lists the distinguishing features of each mode.

Feature	NAT mode	Cut-through mode
---------	----------	------------------

Feature	NAT mode	Cut-through mode
Support for displaying the associated EIPs in the ENI information	No	Yes  Note You can run the <code>ifconfig</code> or <code>ipconfig</code> command to query the EIP associated with the corresponding ENI.
Supported ENIs that can be associated with EIPs	Primary and secondary ENIs	Secondary ENIs
The maximum number of EIPs that can be associated with a primary ENI	1	EIP associations with primary ENIs are not supported.
The maximum number of EIPs that can be associated with a secondary ENI	Depends on the number of private IP addresses of the secondary ENI.  Note A one-to-one relationship is implemented between EIPs and the private IP addresses of a secondary ENI. For example, if a secondary ENI has 10 private IP addresses, a maximum of 10 EIPs can be associated with this ENI.	1  Note In the cut-through mode, an EIP can only be associated with the primary private IP address of a secondary ENI.
Support for the secondary ENI to deliver networking connectivity within the private network after the ENI is associated with an EIP	Yes	No
Supported protocols	EIPs that are associated in the NAT mode do not support the protocols that require connection management from NAT application-level gateway (ALG), such as H.323, Session Initiation Protocol (SIP), Domain Network System (DNS), and Real Time Streaming Protocol (RTSP).	All IP protocols, such as H.323, SIP, DNS, RTSP, File Transfer Protocol (FTP), and Trivial File Transfer Protocol (TFTP).

26.4.2.5.2. Associate an EIP with an ENI in the NAT mode

This topic describes how to associate an EIP with an ENI in the NAT mode. With this association, both the public and private IP addresses of the ENI can be used to provide networking connectivity. EIPs associated in the NAT mode cannot be displayed in the ENI information.

Prerequisites

Before you associate an EIP with an ENI in the NAT mode, make sure that the following conditions are met:

- You have created a secondary ENI that is attached to a VPC. The secondary ENI and the EIP to be associated reside in the same region. For more information, see the [Create an ENI](#) topic of *Elastic Network Interfaces in the Apsara Stack Elastic Compute Service User Guide*.
- The secondary ENI is not associated with an ECS instance.

If the secondary ENI is associated with an ECS instance, you must disassociate it from the ECS instance. You can re-associate the ENI with the ECS instance after you have associated the ENI with the intended EIP in the NAT mode. For more information, see the **Unbind a secondary ENI from an instance** topic of **Elastic Network Interfaces** in the *Apsara Stack Elastic Compute Service User Guide*.

Context

The NAT mode has the following characteristics:

- The number of EIPs with which a secondary ENI can be associated depends on the number of private IP addresses of this secondary ENI.
- When an EIP is associated with an ENI in the NAT mode, both the private and public IP addresses of this ENI are available.
- The EIP associated with an ENI cannot be viewed in the operating system. To query the EIP associated with an ENI, call the DescribeEipAddresses operation.
- EIPs that are associated in the NAT mode do not support the protocols that require connection management from NAT ALG, such as H.323, SIP, DNS, RTSP, and TFTP.

Procedure

1. **Log on to the EIP console.**
2. In the top navigation bar, select the region of the EIP.
3. On the **Elastic IP Addresses** page, find the target EIP and click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
IP Address	Displays the target EIP.
Instance Type	Select Secondary ENI.
Mode	Select NAT Mode.
Secondary ENI	Select the secondary ENI to be associated with the EIP.

26.4.2.5.3. Associate an EIP with an ENI in the cut-through mode

This topic describes how to associate an EIP with an ENI in the cut-through mode. After the association, the EIP replaces the private IP address of the secondary ENI, which changes the secondary ENI to an Internet interface. You can view the EIP in the ENI information.

Prerequisites

Before you associate an EIP with an ENI in the cut-through mode, make sure that the following conditions are met:

- The secondary ENI is attached to a VPC.
- The secondary ENI and the EIP reside in the same region.
- The secondary ENI is not associated with an ECS instance.

If the secondary ENI is associated with an ECS instance, you must disassociate it from the ECS instance. You can re-associate the ENI with the ECS instance after you have associated the ENI with the intended EIP in the cut-through mode. For more information, see the **Unbind a secondary ENI from an instance** topic of **Elastic Network Interfaces** in the *Apsara Stack Elastic Compute Service User Guide*.

- A secondary ENI can be associated with only one EIP.

Context

An EIP is essentially a NAT IP address. After you associate an EIP with an ENI in the NAT mode, the public IP address of the ENI is provisioned in a gateway rather than the ENI of the associated ECS instance. Therefore, the public IP address cannot be displayed in the operating system of the ECS instance, which only shows the private IP address in the ENI information. This complicates operations and maintenance because you must keep track of the mappings between network interfaces or servers and public IP addresses. In addition, EIPs that are associated with ENIs in the NAT mode do not support protocols such as H.323, SIP, DNS, or RTSP.

To solve the preceding problems, you can associate an EIP with an ENI in cut-through mode. In the cut-through mode:

- The EIP replaces the private IP address of the secondary ENI. This changes the secondary ENI to an Internet interface which can no longer deliver networking connectivity within the private network.
- You can view the EIP in the ENI information. To query the EIP associated with the corresponding ENI, you can call the `theifconfig oripconfig` operation.
- EIPs that are associated in the cut-through mode support all Internet protocols, such as H.323, SIP, DNS, RTSP, FTP, and TFTP.

Procedure

1. [Log on to the EIP console](#).
2. Select the region of the EIP.
3. On the **Elastic IP Addresses** page, find the EIP that you want to manage, and click **Bind** in the **Actions** column.
4. In the **Bind Elastic IP Address** dialog box, set the following parameters and click **OK**.

Parameter	Description
Instance Type	Select Secondary ENI .
Binding mode	Select Cut-Through Mode .
Secondary ENI	Select the secondary ENI to be associated with the EIP.

 **Notice** Make sure that the selected secondary ENI is not associated with an ECS instance.

5. Return to the **Elastic IP Addresses** page and click the ID of the associated ENI.
6. Find the target ENI from the list of ENIs. In the **Actions** column, click **Bind** to associate the ENI with an ECS instance.

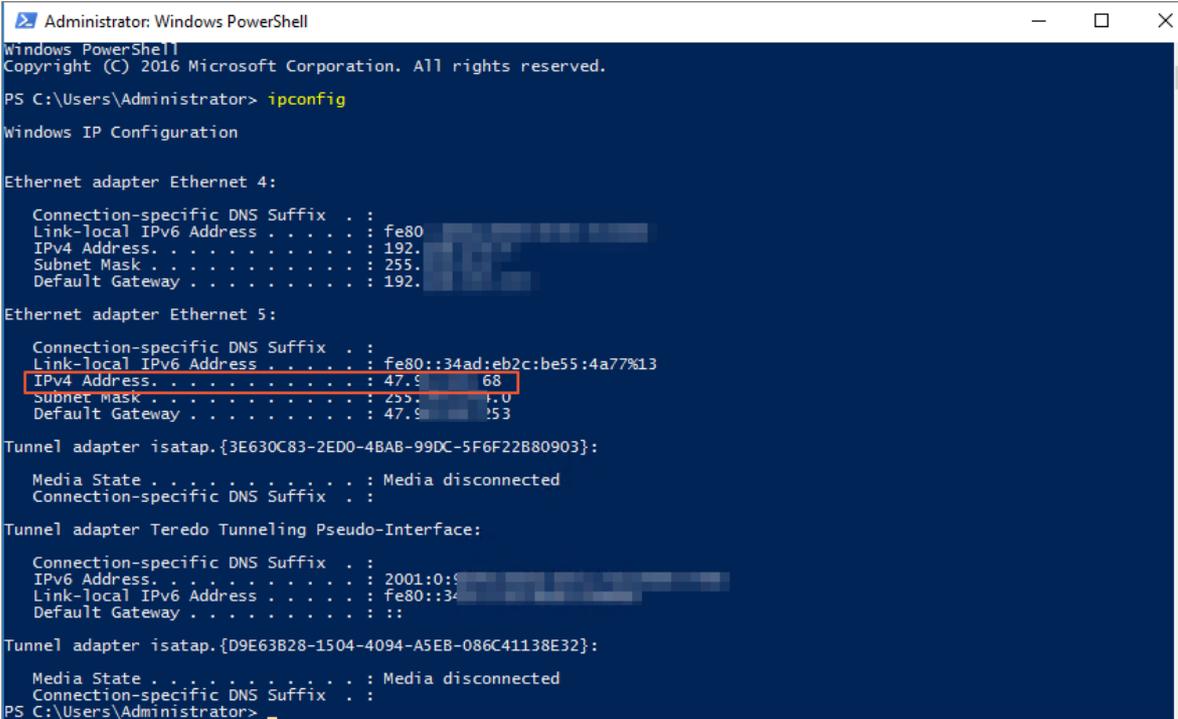
Note

- The maximum number of ENIs that can be associated with an ECS instance depends on the specification of the ECS instance. For more information, see the **Instance types** topic of *What is ECS* in the *Apsara Stack Elastic Compute Service User Guide*.
- After you associate a secondary ENI with an ECS instance, you must enable Dynamic Host Configuration Protocol (DHCP) for the secondary ENI and then restart the ENI. Otherwise, the configuration made for the cut-through mode will not take effect.
- After the configuration is complete, a route entry is automatically created for the ECS instance with the secondary ENI as the egress interface. The priority of this route entry is lower than that of the route with the primary ENI as the egress interface. Therefore, you can adjust the priority of these entries based on your service requirements.

7. Log on to the ECS instance by using the associated EIP to check the network configuration of the ECS instance.

Note Make sure that the security group rules of the ECS instance allow remote access.

As shown in the following figure, the IPv4 address of the ECS instance is changed to the associated EIP.



26.4.3. Upgrade a subscription EIP

This topic describes how to upgrade the bandwidth of EIP. Bandwidth upgrades take effect immediately.

Procedure

1. Log on to the EIP console.
2. In the upper-left corner, select the region where your EIP is created.
3. On the Elastic IP Addresses page, find the target EIP and choose More > Upgrade in the Actions column.
4. On the Change Specifications page, set a new peak bandwidth for the EIP, and then click Submit.

26.4.4. Disassociate an EIP from a cloud resource

This topic describes how to dissociate an EIP from a cloud resource when this cloud resource no longer needs to communicate with the Internet.

Procedure

1. Log on to the EIP console.
2. In the upper-left corner, select the region where your EIP is created.
3. On the Elastic IP Addresses page, find the target EIP, and click Unbind in the Actions column.
4. In the Unbind Elastic IP Address dialog box, click OK.

26.4.5. Release an EIP

This topic describes how to release an Elastic IP address (EIP).

Prerequisites

The EIP is disassociated from all instances. For more information, see [Disassociate an EIP from a cloud resource](#).

Procedure

1. [Log on to the EIP console](#).
2. In the upper-left corner, select the region where your EIP is created.
3. On the **Elastic IP Addresses** page, find the target EIP, move the mouse pointer over **More operations** in the **Actions** column, and then click **Release**.
4. In the **Release an EIP** dialog box, click **OK**.

27. Express Connect

27.1. What is Express Connect?

This topic provides an overview of Express Connect. Express Connect allows you to establish private connections to enable fast, stable, and secure communication between different networking environments. You can use Express Connect to ensure network stability and prevent data breaches.

Features

You can use a leased line provided by an Internet Service Provider (ISP) to establish a physical connection between your data center and an Alibaba Cloud access point. After the physical connection is established, you can create a virtual border router (VBR) to connect your data center with Alibaba Cloud to build a hybrid cloud.

The physical connections of Express Connect do not traverse the Internet, and therefore feature faster speeds, lower latency, greater security, and higher reliability compared with Internet connections.

Express Connect enables you to create a peering connection between two Virtual Private Clouds (VPCs) as a channel for private communication.

Benefits

Express Connect provides the following benefits:

- High-speed interconnections

Powered by the network virtualization technology of Alibaba Cloud, Express Connect allows networks to connect and exchange traffic at high speeds within internal networks without carrying traffic across the Internet. The impact of distance on network performance is minimized to ensure low-latency and high-bandwidth communication.

- Stability and reliability

Built on the state-of-the-art infrastructure of Alibaba Cloud, Express Connect guarantees stable and reliable communication between networks.

- Security

Express Connect implements cross-network communication at the network virtualization layer, where data is transmitted over separate and private channels within the infrastructure of Alibaba Cloud, mitigating the risks of data breaches.

- Buy-as-you-need service

Express Connect delivers connectivity with a wide range of bandwidth options. You can choose based on the needs of your business to get the best value for your purchase.

27.2. Log on to the Express Connect console

This topic provides an example of how to log on to the Apsara Stack Cloud Management (ASCM) console and then access Express Connect by using Google Chrome .

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

Note When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > Express Connect**.

27.3. Connect two VPCs

This topic describes how to create a peering connection to connect two VPCs.

Procedure

1. **Log on to the Express Connect console.**
2. In the left-side navigation pane, choose **VPC Peering Connections > VPC-to-VPC**.
3. On the **VPC Peering Connections** page, click **Create Peering Connection**.
4. Specify the following information to configure the peering connection.

Parameter	Description
Scenario	Select VPC-to-VPC to create a peering connection between two VPCs.
Source Configurations	
Organization	Select the organization to which the source VPC belongs.
Resource Set	Select the resource set to which the source VPC belongs.
Region	Select the region where the source VPC resides.

Parameter	Description
Router Type	Use the default VRouter setting.
Source VPC ID	Select the ID of the source VPC (initiator or acceptor of this peering connection).
Source Router	Select the router of the source VPC (initiator or acceptor of this peering connection).
Destination Configurations	
Organization	Select the organization to which the destination VPC belongs.
Resource Set	Select the resource set to which the destination VPC belongs.
Destination Region	Select the region where the destination VPC resides.
Destination Router Type	Use the default VRouter setting.
Destination VPC ID	Select the ID of the destination VPC (initiator or acceptor of this peering connection).
Destination Router	Select the router of the destination VPC (initiator or acceptor of this peering connection).
Basic Settings	
Bandwidth	Select a bandwidth for the peering connection.

5. Click **Submit**.

27.4. Delete a peering connection

This topic describes how to delete a peering connection.

Procedure

1. **Log on to the Express Connect console.**
2. In the left-side navigation pane, choose **VPC Peering Connections > VPC-to-VPC**.
3. Find the target peering connection and choose  > **Suspend Initiator** in the **Actions** column.
4. Find the target peering connection and choose  > **Suspend Acceptor** in the **Actions** column.
5. Find the target peering connection and choose  > **Delete** in the **Actions** column.
6. In the **Delete Peering Connection** dialog box, click **Confirm**.

28. Apsara Stack Security

28.1. What is Apsara Stack Security?

Apsara Stack Security is a solution that provides Apsara Stack with a full suite of security features, such as network security, server security, application security, data security, and security management.

Background

Traditional security solutions for IT services detect attacks on network perimeters. They use hardware products such as firewalls and intrusion prevention systems (IPSs) to protect networks against attacks.

However, with the development of cloud computing, an increasing number of enterprises and organizations now 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 secure cloud assets.

With the powerful data analysis capabilities and professional security operations team of Alibaba Cloud, Apsara Stack Security provides integrated security protection services at the network layer, application layer, and server layer.

Complete security solution

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services, and provides users with a comprehensive security solution.

Security domain	Service	Description
Security management	Threat Detection Service	Monitors traffic and overall security status to audit and centrally manage security.
Server security	Server Guard	Protects Elastic Compute Service (ECS) instances against intrusions and malicious code.
Application security	Web Application Firewall	Protects web applications against attacks and ensures that mobile and PC users can securely access web applications over the Internet.
Network security	Anti-DDoS	Ensures the availability of network links and improves business continuity.
Data security	Sensitive Data Discovery and Protection	Prevents data leaks and helps your business systems meet compliance requirements.
Security O&M service	On-premises security services	Help you establish and optimize your cloud security system to protect your business system against attacks by making full use of the security features of Apsara Stack Security and other Apsara Stack services.

28.2. Precautions

Before you log on to the Apsara Stack Security console, you must verify that your local PC meets the configuration requirements.

The configuration requirements for the local PC are listed in [Configuration requirements](#).

Configuration requirements

Item	Requirement
------	-------------

Item	Requirement
Browser	<ul style="list-style-type: none"> • Internet Explorer: V11 or later • Chrome (recommended): V42.0.0 or later • Firefox: V30 or later • Safari: V9.0.2 or later
Operating system	<ul style="list-style-type: none"> • Windows XP, Windows 7, or later • macOS

28.3. Quick start

28.3.1. User roles and permissions

This topic describes the user roles involved in Apsara Stack Security.

All roles in Apsara Stack Security Center are provided by default. You cannot add custom roles. Before you log on to Apsara Stack Security Center, make sure that your account is assigned the required role. For more information, see [Default roles in Apsara Stack Security](#).

Default roles in Apsara Stack Security

Role	Permission
System administrator of Apsara Stack Security Center	Manages and configures system settings for Apsara Stack Security Center. The system administrator has permissions to manage Apsara Stack accounts, synchronize data, configure alerts, and configure global settings.
Security administrator of Apsara Stack Security Center	<p>Monitors the security status across Apsara Stack and configures security policies for each functional module of Apsara Stack Security. The security administrator has permissions on all functions under Threat Detection, Network Security, Application Security, Server Security, PM Security, and Asset Management.</p> <p> Note The permissions on WAF and Cloud Firewall must be separately assigned.</p>
Department security administrator	<p>Monitors the security status of cloud product resources in the specified department and configures security policies for each functional module of Apsara Stack Security for this department. The department security administrator has permissions on all functions under Threat Detection, Network Security, Application Security, Server Security, PM Security, and Asset Management. In addition, the department security administrator can specify alert notification methods and alert recipients in the department.</p> <p> Note The permissions on WAF and Cloud Firewall must be separately assigned.</p>
Auditor of Apsara Stack Security Center	Conducts security audits across Apsara Stack. The auditor can view audit events and original logs, configure audit policies, and access all functions under Security Audit.

If you do not have a user that assume the required role, contact the administrator to create the account. For more information, see the [Create a user](#) topic in the *ASCM Console User Guide*

28.3.2. Log on to Apsara Stack Security Center

This topic describes how to log on to Apsara Stack Security Center.

Prerequisites

- Before logging on to the ASCM console, make sure that you have obtained the IP address or domain name of the ASCM console from the deployment personnel. The URL used to access the ASCM console is in the following format: `http://IP address or domain name of the ASCM console/manage`.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to access the ASCM console. Press Enter.
2. Enter your username and password.

The system has a default super administrator, whose username is `super`. The super administrator can create system administrators. A system administrator can create system users and notify the users of the default passwords by SMS or email.

 **Note** When you log on to the ASCM console for the first time, you must modify the password of your username as instructed. For security concerns, your password must meet the minimum complexity requirements: The password must be 8 to 20 characters in length and must contain at least two types of the following characters: letters, digits, and special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Security > Apsara Stack Security**.
5. On the **Apsara Stack Security Center** page, select **Region**.
6. Click **YD** to go to Apsara Stack Security Center.

28.4. Threat Detection Service

28.4.1. Threat Detection Service overview

This topic describes the basic concepts of Threat Detection Service (TDS).

TDS provides comprehensive protection for enterprises. It can monitor vulnerabilities, intrusions, web attacks, DDoS attacks, threat intelligence, and public opinions. TDS uses modeling and analysis to obtain key information based on traffic characteristics, host processes, and host operations logs. In addition, TDS identifies intrusions that cannot be detected by traffic inspection or file scan. You can use the input of cloud analysis models and intelligence data to discover sources and behavior of attacks and assess threats.

TDS provides the following features:

- **Overview:** provides a security situation overview and information about security screens.
- **Security Alerts:** displays security alerts that occur in the business system.
- **Attack Analysis:** displays application attacks and brute-force attacks that occur in the system.
- **Cloud Service Check:** checks whether the security configuration for cloud services has risks.
- **Application Whitelist:** provides information about application processes on servers that require protection based on intelligent learning. The application processes are identified as trusted, suspicious, or malicious. This prevents unauthorized processes.
- **Assets:** manages servers and cloud services on Apsara Stack.
- **Security Reports:** allows you to configure security report tasks on Apsara Stack.

28.4.2. Security overview

28.4.2.1. View security overview information

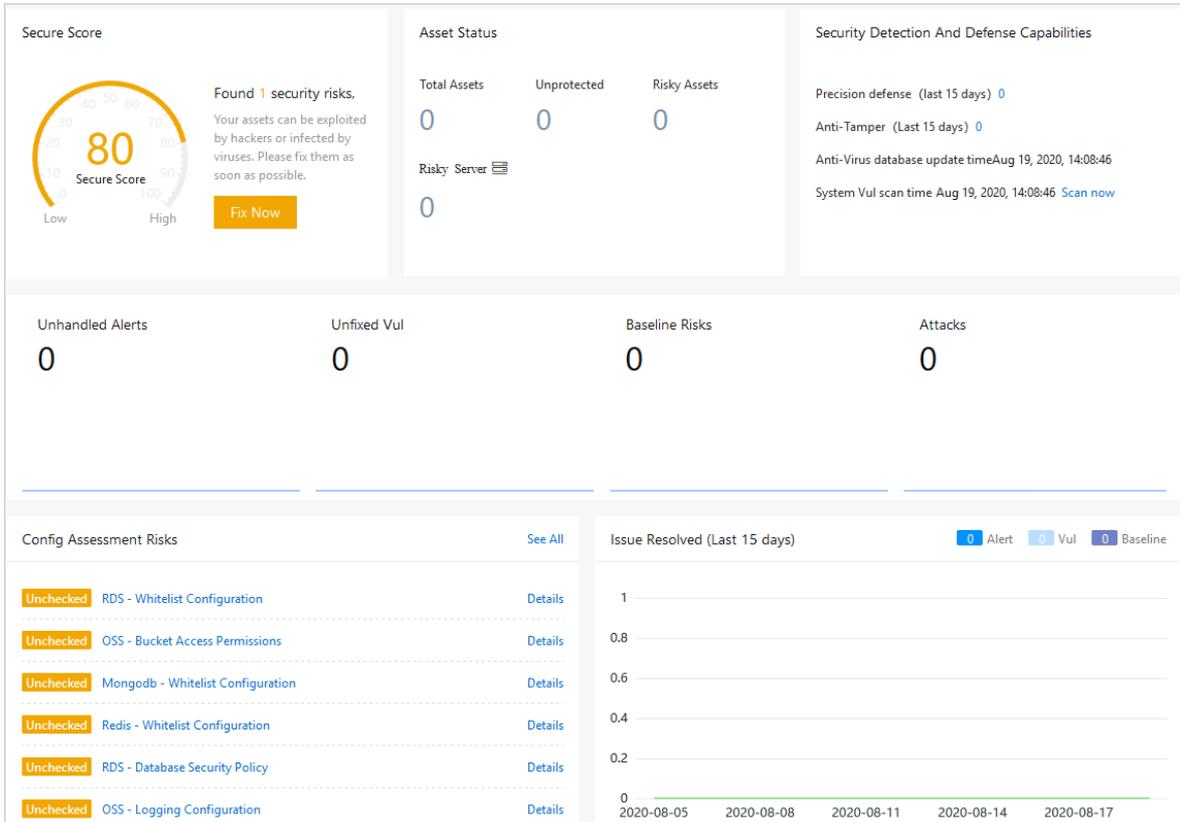
This topic describes how to view security statistics, attack trends, and network traffic information on Apsara Stack.

Context

The **Security Overview** tab provides an overview of detected security events, the latest threats, and inherent vulnerabilities in the system. A security administrator can view information on the **Security Overview** tab to have a comprehensive understanding of the system security situation.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Threat Detection > Overview > Security Overview.**
3. View the security situation of Apsara Stack.



Sections on the Security Overview tab

Section	Description
Security Score	The security score of assets and the number of detected security risks.
Asset Status	The total number of assets and the numbers of servers that are not protected, servers that are stopped, and servers that are at risk.
Security Detection And Defense Capabilities	The numbers of precise defense events and anti-tampering events in the last 15 days, the time when the anti-virus database was updated, and the time of vulnerability scanning. This allows you to obtain information about the defense situation and security status of your assets in real time.
Threat statistics	The numbers of alerts that are not handled, vulnerabilities that are not fixed, baseline risks, and attacks.
Config Assessment Risks	Risks in the baseline configurations of cloud services.
Issue Resolved	Statistics of alerts, vulnerabilities, and baseline risks that have been processed in the last 15 days. The statistics are displayed in a bar and trend chart.

28.4.3. Security alerts

28.4.3.1. View security alerts

This topic describes how to view security alerts on the Security Alerts page.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Threat Detection > Security Alerts**.
3. (Optional)Set filter conditions for security alerts.

 **Note** If you want to view all alerts, do not set the conditions.

Ur... X No... X War... X ▼ Unhandle... ▼ All ▼ Asset Group ▼ Alert/Asset Q

Filter condition	Description
Severity	The severity level. You can select one or more levels. Valid values: <ul style="list-style-type: none"> ○ Urgent ○ Warning ○ Notice
Alert status	The status of alerts. Valid values: <ul style="list-style-type: none"> ○ Unhandled Alerts ○ Handled
Alert type	The type of alerts. Select All or a specific type.
Affected asset group	The affected asset group. Select Asset Group or a specific group.
Search for alerts by name or asset	Enter an alert name or a keyword of affected assets to search for alerts.

4. View security alerts and their details in the list.

28.4.3.2. Manage quarantined files

This topic describes how to manage threat files that are quarantined by the system. The system deletes a quarantined file 30 days after the file is quarantined. You can restore the file before it is deleted.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Threat Detection > Security Alerts**.
3. In the upper-right corner of the **Alerts** page, click **Quarantine**.
4. On the **Quarantine** page, view information about a quarantined file, such as the host, path, status, and operation time.
5. (Optional)If a file is incorrectly quarantined, click **Restore** in the **Actions** to restore the file.

 **Notice** Before you restore a quarantined file, make sure that the file is normal and does not bring risks.

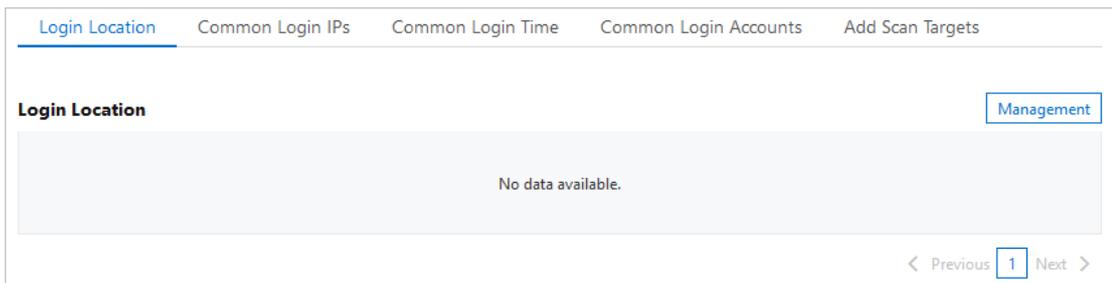
The restored file is removed from the quarantine and is displayed in the security alert list again.

28.4.3.3. Configure security alerts

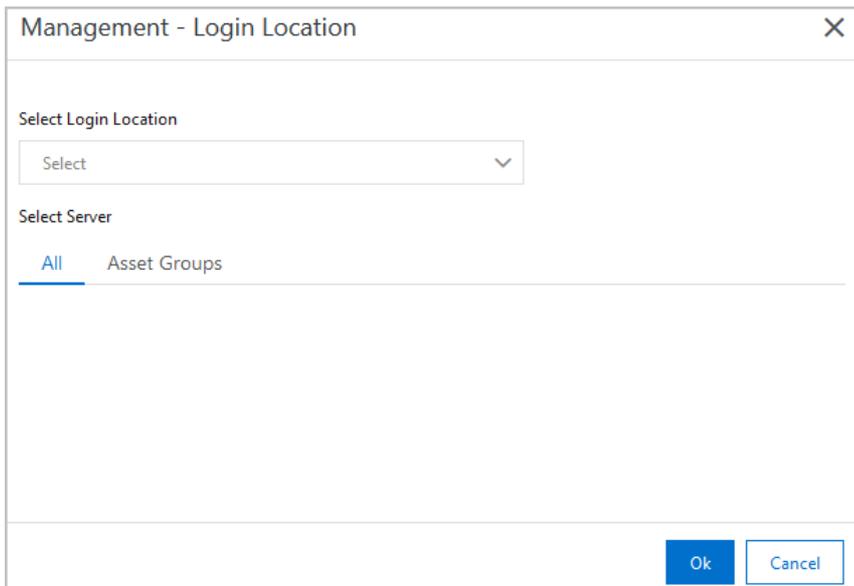
This topic describes how to configure logon settings and web directories that are scanned.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Security Alerts.**
3. In the upper-right corner of the page, click **Settings**. You can perform the following operations:
 - o **Manage a common logon location.**
 - a. On the right side of **Login Location**, click **Add**.



- b. Select a logon location and the servers that are allowed to be logged on to from the location.



- c. Click **OK**.

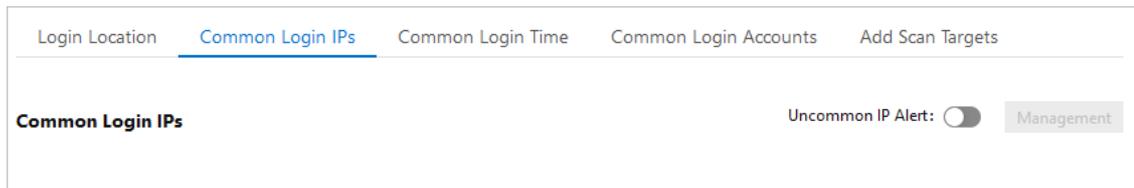
Threat Detection Service (TDS) allows you to edit and delete added logon locations.

- Find the target logon location and click **Edit** on the right side to change the servers that are allowed to be logged on to from this location.
- Find the target logon location and click **Delete** on the right side to delete the logon location.

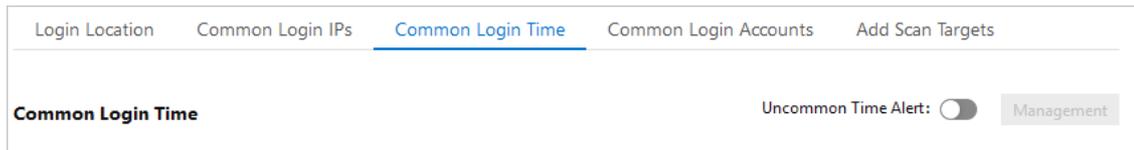
- o **Configure advanced logon settings.**

Note You can specify the IP addresses, accounts, and time periods that are allowed to log on to your assets. After you configure these settings, alerts are triggered if your assets receive logon requests that do not meet the requirements. The procedure to configure advanced logon settings is similar to that to configure common logon locations. You can follow the preceding procedure to add, edit, and delete advanced logon settings.

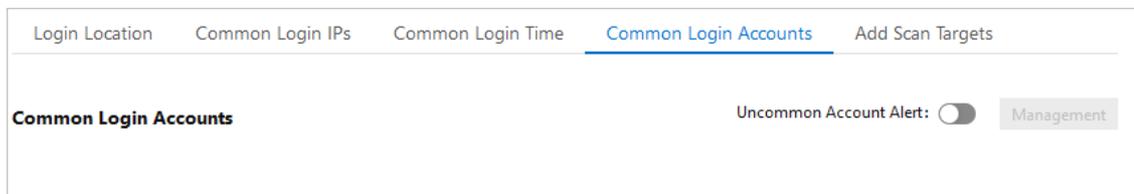
- On the right side of **Common Login IPs**, enable or disable **Uncommon IP Alert**. After this function is enabled, alerts are triggered if your assets receive logon requests from unauthorized IP addresses.



- On the right side of **Common Logon Time**, enable or disable **Uncommon Time Alert**. After this function is enabled, alerts are triggered if your assets receive logon requests at unauthorized time.

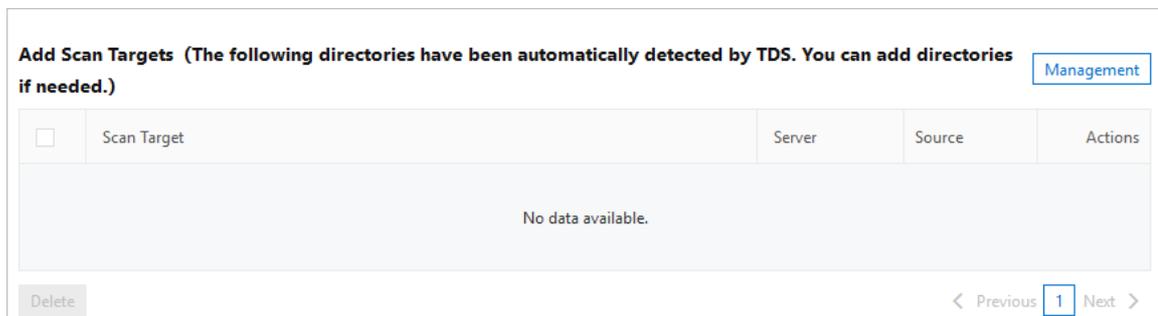


- On the right side of **Common Login Accounts**, enable or disable **Uncommon Account Alert**. After this function is enabled, alerts are triggered if your assets receive logon requests from unauthorized accounts.



- **Add a web directory as the scan target.**

TDS automatically detects web directories on your servers.



It performs dynamic and static scans on these directories. You can also manually add web directories for scanning.

- On the right side of **Add Scan Targets**, click **Add**.
- Enter a valid web directory and select the servers on which the directory is scanned. The web directory is added to the scan list.

Note To ensure performance and efficiency, do not enter a root directory.

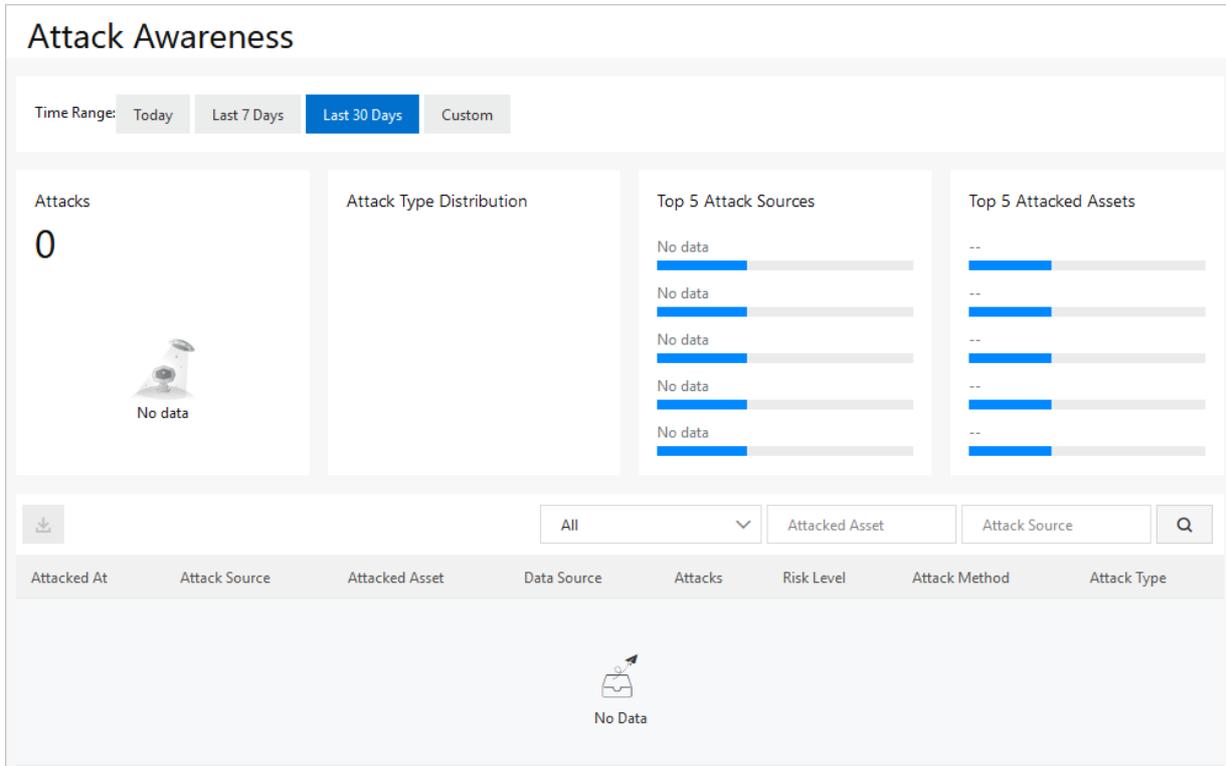
- Click **OK**.

28.4.4. Attack analysis

This topic describes the statistics provided by the attack analysis feature. These statistics include the total number of attacks, the distribution of attack types, top five attack sources, top five attacked assets, and the attack list.

Background information

The attack analysis feature provides basic attack detection and prevention capabilities in the Apsara Stack Security console. We recommend that you develop a more refined and in-depth defense system by optimizing firewalls and enhancing business security.



On the **Attack Awareness** page, you can specify a time period to view the following attack analysis statistics. You can view the attack analysis statistics of the current day, last 7 days, or last 30 days. You can also select **Custom** to view the statistics of a time period within the last 30 days.

- **Attacks**
- **Attack Type Distribution:** the attack types and the number of attacks of each type.
- **Top 5 Attack Sources:** the top five IP addresses where most attacks are initiated.
- **Top 5 Attacked Assets:** the top five assets that have encountered the most attacks.
- **Attack list:** The details of all attacks. These details include the attack time, source IP address, attacked asset, attack type, and attack status.

Note The attack list displays a maximum of 10,000 entries. You can specify **Time Range** to view details about attacks that occur over a specified time range.

Parameters in the attack list

Parameter	Description
Attacked At	The time when an attack is detected.
Attack Source	The source IP address of an attack.
Attacked Asset	The name, public IP address, and private IP address of an attacked asset.
Attack Method	The HTTP request method used to initiate an attack. Valid values: POST and GET.

Parameter	Description
Attack Type	The type of an attack. Supported attack types include SSH brute-force attacks and remote code execution attacks.

- **Search for an attack**

To view the details of a specific attack, you can specify search conditions in the upper-right corner of the attack list. Search conditions include the attack type, attacked asset, and source IP address.

- **View the details of an attacked asset**

To view the details of an attacked asset, you can place the pointer over the name of the **Attacked Asset**.

- **Export the attack list**

To export and save the attack list to your computer, you can click the  icon in the upper-left corner of the attack list. The asset records are exported to an Excel file.

28.4.5. Cloud service check

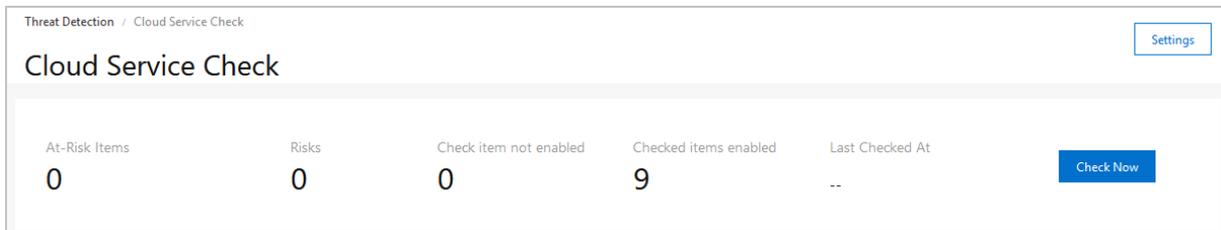
28.4.5.1. Overview

Threat Detection Service (TDS) provides the cloud service check feature. This feature allows you to check for security risks in the configurations of your Alibaba Cloud services. This topic describes the functions and check items that are supported by the cloud service check feature.

Background information

The cloud service check feature allows you to perform network access control and data security checks. The checks help you detect configuration risks of your Alibaba Cloud services and provide repair solutions.

You can view the number of **Checked items** enabled on the **Cloud Service Check** page.



At-Risk Items	Risks	Check item not enabled	Checked items enabled	Last Checked At
0	0	0	9	--

Cloud service check list

The following table lists the check items.

Type	Supported item	Description
	PolarDB - Backup configurations	Checks whether the automatic backup feature is enabled for PolarDB. Regular backups help you improve database security. You can restore data if an error occurs in your database. PolarDB supports automatic backup. We recommend that you enable automatic backup to create a backup on a daily basis.
	Container Registry - Repository permission configurations	Checks whether the status of Container Registry warehouses is set to private. Container Registry warehouses include public warehouses and private warehouses. Public warehouses allow all Internet users to anonymously download information. If an image contains sensitive information, we recommend that you set the value to private. Otherwise, ignore relevant alerts.

Type	Supported item	Description
Data security	OSS - Bucket server-side encryption	Checks whether the data encryption feature of Object Storage Service (OSS) buckets is enabled. OSS supports server-side encryption to secure data that is persistently stored in OSS. We recommend that you enable server-side encryption to protect sensitive data.
	OSS - Sensitive information leakage scans	Checks whether OSS sensitive files require access permissions.
	ApsaraDB for RDS - Cross-region backup configurations	Checks whether cross-region backup is enabled for ApsaraDB for RDS instances. ApsaraDB RDS for MySQL provides the cross-region backup feature that automatically synchronizes local backup files to OSS in another region. We recommend that you enable the cross-region backup feature.
	KVStore for Redis - Backup configurations	Checks whether the data backup feature is enabled for ApsaraDB for Redis instances.
	ApsaraDB for MongoDB - SSL encryption	Checks whether SSL encryption is enabled for ApsaraDB for MongoDB databases. We recommend that you enable the SSL encryption feature to improve the security of data links in ApsaraDB for MongoDB databases.
	ApsaraDB for MongoDB - Backup configurations	Checks whether the automatic backup feature is enabled for ApsaraDB for MongoDB databases. Regular backups help you improve database security. You can restore data if an error occurs in your database. ApsaraDB for MongoDB provides automatic backup policies. We recommend that you enable automatic backup to create a backup on a daily basis.
	ECS - Disk encryption	Checks whether encryption is enabled for disks on Elastic Compute Service (ECS) instances.
	ECS - Automatic snapshot policies	Checks whether the automatic snapshot feature is enabled for the disks on ECS instances. The automatic snapshot feature improves the security of ECS data and supports disaster recovery.
	OSS - Bucket permissions	Checks whether the OSS bucket permission is set to <i>private</i> .
	OSS - Logging configuration	Checks whether the logging feature is enabled for OSS.
	OSS - Cross-region replication configurations	Checks whether the cross-region replication feature is enabled for OSS.
	ApsaraDB for RDS - Database security policies	Checks whether the SQL audit, SSL encryption, and Transparent Data Encryption features are enabled for ApsaraDB for RDS databases.
	ApsaraDB for RDS - Backup configurations	Checks whether the data backup feature is enabled for ApsaraDB for RDS instances.
	SSL Certificates Service - Validity check	Checks whether your SSL certificate is expired. If your SSL certificate is expired, you are not allowed to use SSL Certificates Service.

Type	Supported item	Description
Network access control	ECS - Security group policies	Checks ECS security group policies. We recommend that you grant minimum permissions to users. If you set 0.0.0.0/0 for a service, it indicates that access from all IP addresses is allowed. For example, you can set 0.0.0.0/0 for ports 80, 443, 22, and 3389.
	OSS - Bucket hotlinking protection	Checks whether hotlinking protection is enabled for OSS buckets. The OSS hotlinking protection feature checks the Referer header to deny access from unauthorized users. We recommend that you enable this feature.
	VPC - DNAT management port mapping	Checks whether a port is mapped to the Internet. When you create a Destination Network Address Translation (DNAT) rule for a NAT Gateway that is deployed in a virtual private cloud (VPC), we recommend that you do not map internal management ports to the Internet. Do not map all ports or an important port, for example, ports 22, 3389, 1433, or 3306.
	Apsara Stack Security - Back-to-origin configuration checks	Checks whether Anti-DDoS Pro or Anti-DDoS Premium is configured to allow only Web Application Firewall (WAF) back-to-origin IP addresses. After you set up Anti-DDoS Pro, Anti-DDoS Premium, or WAF, you must hide the IP addresses of the backend servers to prevent attacks on the cloud assets.
	Apsara Stack Security - WAF back-to-origin configurations	Checks whether WAF only allows requests from WAF back-to-origin IP addresses. After you set up Anti-DDoS Pro, Anti-DDoS Premium, or WAF, you must hide the IP addresses of the backend servers to prevent attacks on the cloud assets.
	SLB - IP address whitelist configurations	Checks the access control configurations of Server Load Balancer (SLB) instances. Checks whether access control is enabled for HTTP and HTTPS services and checks whether 0.0.0.0/0 is added to the IP address whitelist.
	SLB - High-risk ports	Checks whether SLB opens ports to the Internet for forwarding unnecessary public services.
	ApsaraDB for RDS - IP address whitelist configurations	Checks whether the ApsaraDB for RDS access control policy is set to 0.0.0.0/0 (all IP addresses), which allows requests from all IP addresses. We recommend that you restrict the access scope to a specific range of IP addresses rather than expose database services to the Internet.
	KVStore for Redis - IP address whitelist configurations	Checks whether the KVStore for Redis access control policy is set to 0.0.0.0/0 (all IP addresses), which allows requests from all IP addresses. We recommend that you restrict the access scope to a specific range of IP addresses rather than expose database services to the Internet.
	AnalyticDB for PostgreSQL - IP address whitelist configurations	Checks whether the AnalyticDB for PostgreSQL access control policy is set to 0.0.0.0/0 (all IP addresses), which allows requests from all IP addresses. We recommend that you restrict the access scope to a specific range of IP addresses rather than expose database services to the Internet.
PolarDB - IP address whitelist configurations	Checks whether IP address whitelists of PolarDB are configured to allow requests from 0.0.0.0/0 (all IP addresses). To prevent security risks, we recommend that you configure IP address whitelists to allow only requests from specific IP addresses.	

Type	Supported item	Description
	ApsaraDB for MongoDB - IP address whitelist configurations	Checks whether the ApsaraDB for MongoDB access control policy is set to 0.0.0.0/0 (all IP addresses), which allows requests from all IP addresses. We recommend that you restrict the access scope to a specific range of IP addresses rather than expose database services to the Internet.

28.4.5.2. Run cloud service checks

Threat Detection Service (TDS) provides the cloud service check feature. This feature allows you to check for security risks in the configurations of your Alibaba Cloud services. This topic describes how to manually run cloud service checks on your Alibaba Cloud services. This topic also describes how to specify a detection interval to automate periodic checks.

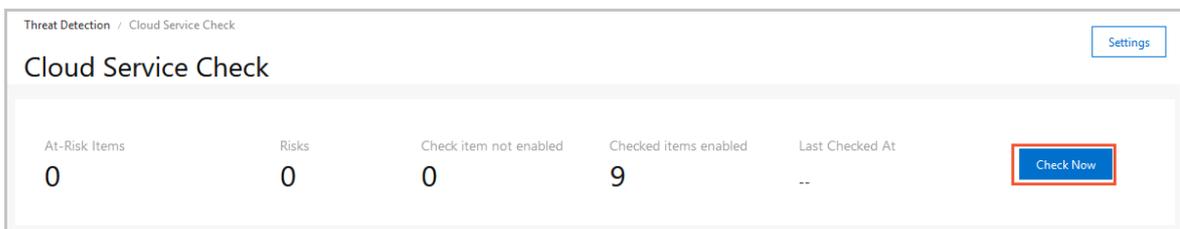
Context

Apsara Stack Security Center supports manual and automated periodic checks to scan for configuration risks in cloud services.

- **Manual checks:** On the **Cloud Service Check** page, click **Check Now** to check for security risks in the configurations of your Alibaba Cloud services
- **Automated periodic checks:** By default, Apsara Stack Security Center automatically runs configuration checks during the 00:00:00-06:00:00 time range every other day. You can also set a custom time period to automatically check whether the configurations of your Alibaba Cloud services contain risks. This helps you detect and manage configuration risks at the earliest opportunity.

Manual checks

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3. On the **Cloud Service Check** page, click **Check Now** to check whether the configurations of all your Alibaba Cloud services contain risks. After you run a check, the number of affected assets is displayed on this page.

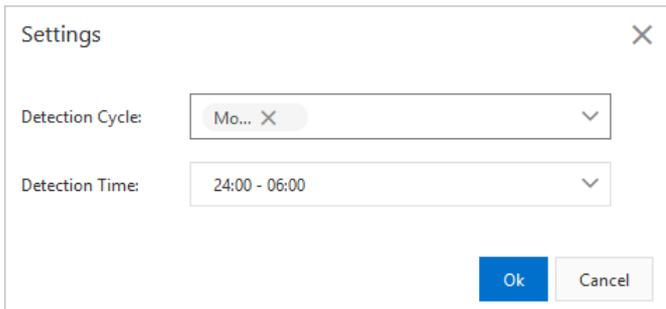


Note Do not perform other operations until the check is complete.

After the check is complete, the detected risks are listed based on their risk levels.

Automated checks

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3. In the upper-right corner of the **Config Assessment** page, click **Settings**.
4. In the **Settings** dialog box, specify **Detection Cycle** and **Detection Time**.



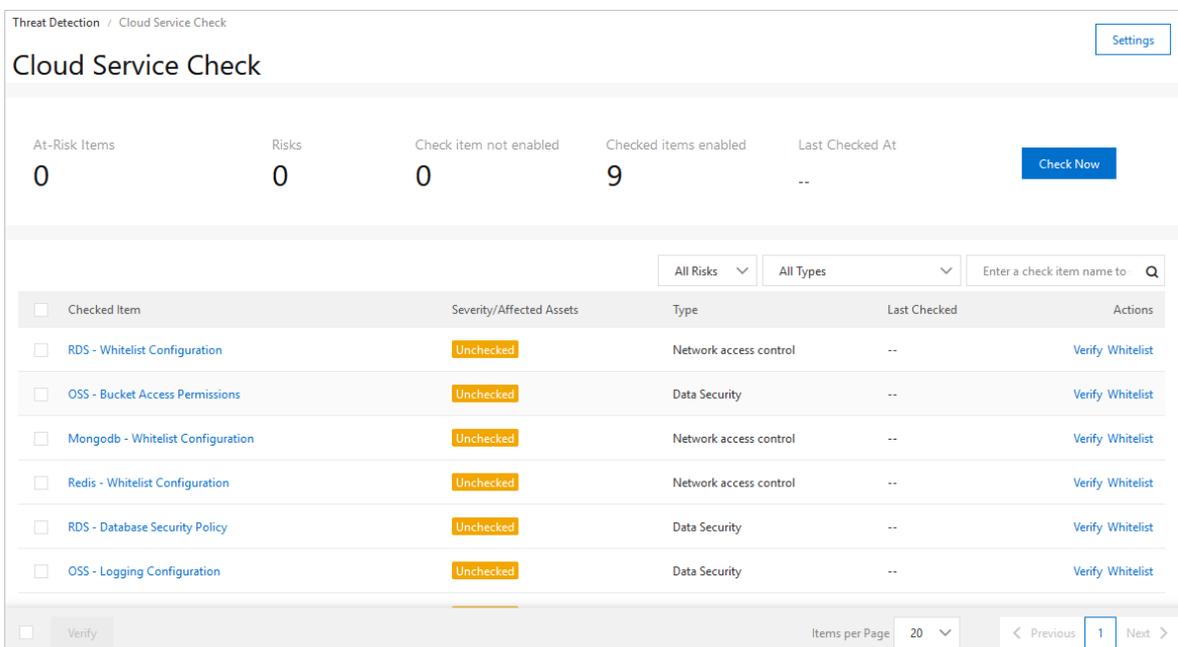
- **Detection Cycle:** supports Monday to Sunday. You can select multiple days.
 - **Detection Time:** Select 24:00:00 - 06:00:00 , 06:00:00 - 12:00:00 , 12:00:00 - 18:00:00 , or 18:00:00 - 24:00:00 .
5. Click **OK**.
During the selected period, Apsara Stack Security Center automatically runs checks based on all check items.

28.4.5.3. View and manage check results of Alibaba Cloud services

This topic describes how to view check results and manage configuration risks in Apsara Stack Security Center. Check results include check items, details of check items, potential impacts, and suggestions on how to manage configuration risks. You can manage configuration risks on the Cloud Service Check page.

View check results

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3. On the **Cloud Service Check** page, you can view the check results and details of the last configuration check.



- **View statistics on the result of the last check.**
Above the check item list, you can view the number of **At-Risk Items** (including the total number of items and the number of items at each risk level), the number of assets with risks (**Risks**), the number of enabled check items, the number of disabled check items, and the last check time.
- **View the check items.**

In the check item list, you can view information about the check items. The check items include the risk level, the type and number of assets affected by each check item, the type of each check item, and the time of the last check.

- View the check result details of a check item.

Click the name of the target checked item in the **Checked Item** column to go to the details page. You can view the check description, potential risks, and suggestions on how to manage the risks on this page.

Manage configuration risks

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Threat Detection > Cloud Service Check**.
3. On the **Cloud Service Check** page, manage the configuration risks.
 - Check whether the new settings contain risks.

If you have changed the configuration settings of an item, find the item on the check item list on the **Cloud Service Check** page and click **Verify** in the **Actions** column to check whether the new settings contain risks.

Threat Detection / Cloud Service Check

Cloud Service Check Settings

At-Risk Items: 0 Risks: 0 Check item not enabled: 0 Checked items enabled: 9 Last Checked At: -- Check Now

All Risks All Types Enter a check item name to:

<input type="checkbox"/> Checked Item	Severity/Affected Assets	Type	Last Checked	Actions
<input type="checkbox"/> RDS - Whitelist Configuration	Unchecked	Network access control	--	<input type="button" value="Verify"/> Whitelist
<input type="checkbox"/> OSS - Bucket Access Permissions	Unchecked	Data Security	--	Verify Whitelist
<input type="checkbox"/> MongoDB - Whitelist Configuration	Unchecked	Network access control	--	Verify Whitelist
<input type="checkbox"/> Redis - Whitelist Configuration	Unchecked	Network access control	--	Verify Whitelist
<input type="checkbox"/> RDS - Database Security Policy	Unchecked	Data Security	--	Verify Whitelist
<input type="checkbox"/> OSS - Logging Configuration	Unchecked	Data Security	--	Verify Whitelist

Verify Items per Page: 20 < Previous 1 Next >

- Add a check item to the whitelist.

If you want to ignore the detected risks of a checked item, find the item on the check item list and click **Whitelist** to add the item to the whitelist. The status of the item will change to **Ignored**. Ignored items are not counted as **At-Risk** items.

In the check item list, you can also click **Remove** to remove the **Ignored** items from the whitelist.

Note After you click **Whitelist**, the risk is ignored for this time only. If the risk is detected in the future, Apsara Stack Security Center will continue to report it.

28.4.6. Application whitelist

Application whitelists can prevent unauthorized programs from running on your servers and provide a trusted running environment for your assets.

Context

The application whitelist feature allows you to add servers and trusted applications to a whitelist. Applications that are not specified in the whitelist cannot run on your servers. This feature protects your servers from untrusted or malicious programs and improves resource utilization.

After you apply a whitelist policy to a server, Apsara Stack Security Center detects suspicious or malicious processes and generates alerts on the processes that are not specified in the whitelist.

Note An alert is triggered if a process that is not specified in the whitelist is detected. The detected process may be a normal process or a malicious process. If you trust the process that triggers an alert, we recommend that you add the process to the whitelist. A process that is added to the whitelist no longer triggers alerts when it restarts. If the process is malicious, we recommend that you remove this process immediately and check whether configuration files of scheduled tasks have been modified.

Step 1: Create an application whitelist policy

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Threat Detection > Application Whitelists**.
3. On the **App Control** page, click the **Policies** tab.
4. On the **App Control** page, click the **Policies** tab. Then click **Create Policy** in the upper-left corner.
5. In the **Create Policy** step of the **Create Whitelist Policy** pane, configure the following parameters:
 - **Policy Name:** Enter a whitelist policy name.
 - **Intelligent Learning Duration:** Select a duration for intelligent learning. Valid values: 1 Day, 3 Days, 7 Days, and 15 Days. The intelligent learning feature uses machine learning to automatically collect and categorize large amounts of alert data. Apsara Stack Security Center can identify suspicious or malicious processes based on the collected data.
 - **Servers for Intelligent Learning:** Select the servers that you want to add to the whitelist.
6. Click **Next** to create the whitelist policy.
After the whitelist policy is created, its details are automatically displayed in the policy list on the **Policies** tab.

The following table lists the parameters in the policy list.

Parameter	Description
Policy Name	The name of the created whitelist policy.
Servers	The number of servers to which the whitelist policy is applied.
Status	<p>The status of the policy. Valid values:</p> <ul style="list-style-type: none"> ○ Applied: Intelligent learning is completed. The policy is applied to servers. ○ Pending Confirmation: Intelligent learning is completed. The policy must be confirmed and enabled. After intelligent learning is completed, you also need to turn on the switch in the Policy Status column to enable this policy. The policy takes effect only after it is enabled. Apsara Stack Security Center automatically identifies the processes on your servers as trusted, suspicious, or malicious processes. ○ Paused: Intelligent learning is manually paused. You can click Continue in the Actions column to resume intelligent learning. ○ Progress: Intelligent learning is in progress. After a whitelist policy is created, Apsara Stack Security Center automatically performs intelligent learning based on the policy. The status of a newly created policy is Progress.
Applications	The numbers of processes, including Trusted , Suspicious , and Malicious processes, on all servers to which the policy is applied.

Parameter	Description
Actions	<p>The operations that you can perform on a policy. Valid values:</p> <ul style="list-style-type: none"> ○ Apply: Add or remove servers to which the policy is applied on the Apply Whitelist Policy pane. ○ Modify: Modify the policy on the Modify Whitelist Policy pane. You can change the values of Policy Name and Intelligent Learning Duration, and modify the servers that need to automatically perform intelligent learning. ○ Pause Learning: Pause intelligent learning. ○ Continue: Resume intelligent learning. <p>After you click Continue, the status of the policy changes to Progress. You can view the learning progress of the policy in the Status column.</p> <ul style="list-style-type: none"> ○ Delete: Delete the policy. <p>After the policy is deleted, the corresponding servers and processes are no longer protected by the policy.</p>

Step 2: Add servers to the application whitelist

1. **Log on to Apsara Stack Security Center.**
 2. In the left-side navigation pane, choose **Threat Detection > Application Whitelists**.
 3. On the **Servers** tab of the **App Control** page, click **Add Server** in the upper-left corner.
 4. In the **Add Server** pane, configure the following parameters:
 - **Whitelist Policy:** Select an existing whitelist policy from the drop-down list.
 - **Event Handling:** The default value is **Alert**, which indicates that Apsara Stack Security Center generates alerts when a suspicious process is detected.

When an unauthorized process starts on a server protected by the whitelist, an alert is automatically triggered. You can click the number in the **Suspicious Events** column to go to the **Alerts** tab and view the alert details.

 - **Servers:** Select the servers that you want to add to the whitelist. You can select multiple servers.

To search for a server, enter the server name in the search box of **Servers** and click the search icon. Fuzzy match is supported.
 5. Click **OK**.
- After you create an application whitelist, you can view the protected servers and the name of the whitelist policy applied to the servers in the server list on the **Servers** tab.

The following information of the added servers is displayed on the **Servers** tab:

- **Server Name/IP:** the name and IP address of the server to which the whitelist policy is applied.
- **Whitelist Policy:** the whitelist policy that is applied to the server.
- **Suspicious Events:** the number of unauthorized processes that are detected on the server. Apsara Stack Security Center generates alerts immediately when a suspicious process is detected.
- **Event Handling:** The default value is **Alert**, which indicates that Apsara Stack Security Center generates alerts when a suspicious process is detected.

When an unauthorized process starts on a server protected by the whitelist, an alert is automatically triggered. You can click the number in the **Suspicious Events** column to go to the **Alerts** tab and view the alert details.

- **Actions:** You can click **Delete** in the **Actions** column to remove a server from the application whitelist.
- After the server is removed from the whitelist, the application whitelist policy no longer protects the server. Apsara Stack Security Center generates alerts when a process starts on that server.

Add or remove a process to or from an application whitelist

After an application whitelist is configured for your servers, you can view the protected servers and the names of the whitelist policies applied to the servers in the server list on the Servers tab. You can click a policy name in the **Whitelist Policy** column to view the processes running on the required server. You can also view the trusted, suspicious, and malicious processes and their detailed information.

The following information about each process on the server is displayed:

- **Type:** the type of the process. Processes are classified as trusted, suspicious, or malicious processes.
- **Process Name:** the name of the process.
- **Hash:** the Hash function of the process. The Hash function is used to ensure that the process is unique and has not been forged.
- **Path:** the file path of the process on the server.
- **Degree of Trustability:** the degree of trustability for the process determined by Apsara Stack Security Center. Valid values: 0% (malicious process), 60% (suspicious process), and 100% (trusted process).

 **Note** We recommend that you focus on the processes of 0% trustability.

- **Actions:** the operations that can be performed on the process. You can determine whether to add the process to the whitelist based on the services deployed on your server. You can perform the following operations:
 - **Add to Whitelist:** If a process is trusted, add it to the whitelist.
 - **Remove from Whitelist:** After a process is removed from the whitelist, Apsara Stack Security Center identifies the process as untrusted and generates an alert when this process starts.

28.4.7. Assets

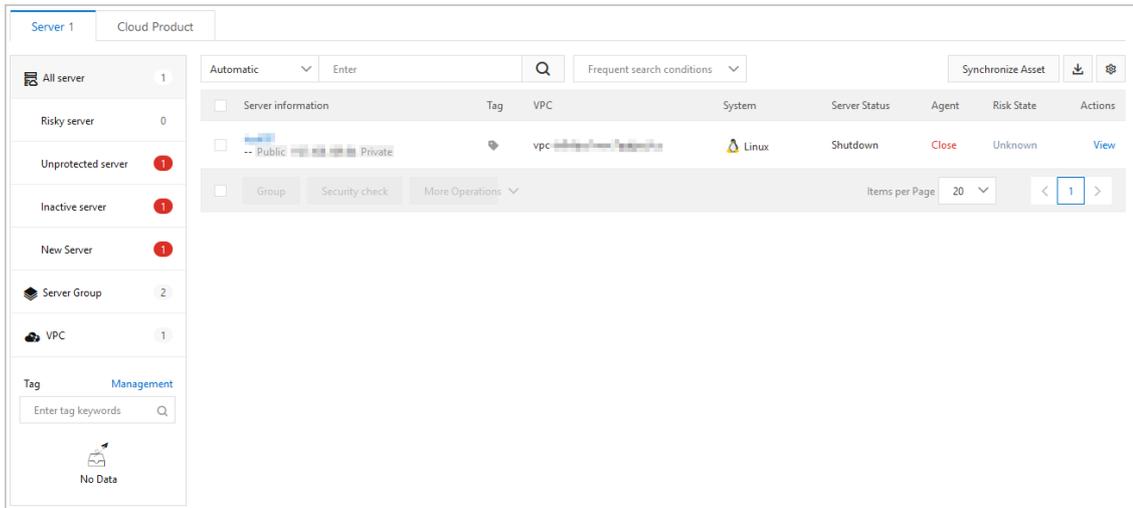
28.4.7.1. View the security status of a server

The Assets page displays security information about each protected server, including the virtual private cloud (VPC) that each server resides, server status, and risk status. This topic describes how to filter assets based on search conditions to view the security status of specific servers. This topic also describes how to specify search conditions and select the items that you want to display on the Assets page.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, view the security status of each server. To view the security status of each server, perform one of the following operations:
 - **Filter by server status**

- The All server section displays the numbers of all servers, risky servers, unprotected servers, inactive servers, and new servers.

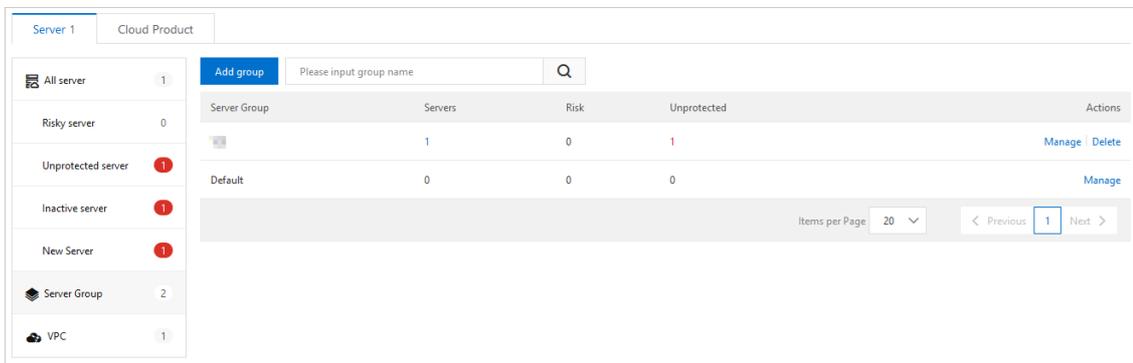


To view the details of a server, click the name of the server, or click Fix in the Actions column. For more information, see [View the details of a single asset](#).

- You can click Risky server, Unprotected server, Inactive server, or New server to view the security status of target servers.

○ Filter by server group

- You can click Server Group to view the total number of server groups and the numbers of servers, servers that are at risk, and unprotected servers in each server group.



You can Manage or Delete to manage a server group in Threat Detection Service (TDS). For more information, see [Manage asset groups](#).

- Find the target server group, click the number in the Servers, Risk, or Unprotected column to view the security status of specified servers in this group.

○ Filter by VPC instance ID

- You can click **VPC** to view the total number of VPCs and the numbers of servers, servers that are at risk, and unprotected servers that are deployed in each VPC.

VPC	Servers	Risk	Unprotected
vpc- <i>id</i>	1	0	1

- Find the target VPC, and click the number in the **Servers**, **Risk**, or **Unprotected** column to view the security status of specified servers in this VPC.

- **Filter by tag**

You can click a tag to the left of the asset list to view the security status of the servers to which this tag is bound.

- **Filter by search condition**

You can click **All server**, **Server Group**, **VPC**, or a **Tag**, and use the search bar on the **Assets** page to filter specific servers.

- Use multiple subconditions to search for specific assets:

Above the asset list of the **Assets** page, you can select a search condition, and select or enter a keyword in the search bar to filter specific servers. Supported search conditions include **Internet IP**, **Private IP**, **Instance name**, **System**, **Baseline problems**, **Vul problems**, **Alert problems**, **Risk Status**, **Online or Offline**, **Tag**, **Group name**, and **OS**.

Note

You can select multiple search conditions at a time and specify the Boolean operator between each search condition. The following content describes Boolean operators:

- Boolean operators:
 - **AND**: specifies the **AND** logical relation for the search conditions.
 - **OR**: specifies the **OR** logical relation for the search conditions.
- If you need to use one search condition and multiple keywords to search for specific servers, set the Boolean operator to **OR**.
- If the search condition requires you to enter a keyword, enter a keyword and then click the Search icon. Results are displayed only after you click the Search icon.

- Use multiple search conditions to search for specific assets:

Apply multiple search conditions to search for specific assets.

You can also select **Server Group**, **Region**, **VPC**, or a **Tag**, and use the search bar on the **Assets** page to filter specific servers.

- **Specify frequently used search conditions**

You can save applied search conditions as frequently used search conditions. Click **Save** and enter a name in the **Save condition** dialog box. Then, you can select the saved search conditions from the **Frequent search conditions** drop-down list on the right of the search bar.

- **Customize displayed items**

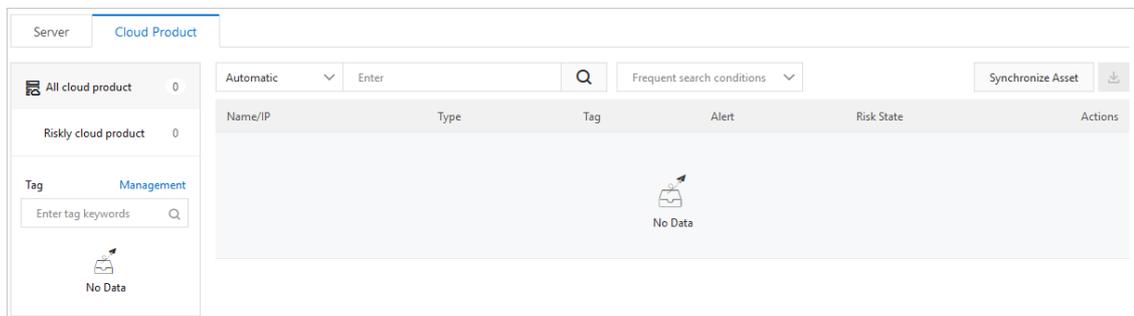
On the Assets page, click the  icon in the upper-right corner. You can then select the items that you want to display on the Assets page.

28.4.7.2. View the security status of cloud services

The Assets page displays the security information about each protected cloud service. The information includes the at-risk services and the types of services, for example, SLB and NAT Gateway. This topic describes how to configure search conditions to view the security status of cloud services.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click the **Cloud Product** tab to view the security status of cloud services. You can perform the following operations based on your business requirements:
 - **Search by asset status**
 - In the left-side pane on the **Cloud Product** tab, you can view the numbers of **All cloud product** and **Risky cloud product**. You can also view the security status of all cloud services.



- You can click **Risky cloud product** to view the cloud services that are at risk.

You can click the name of the target cloud service or click **View** in the **Actions** column that corresponds to a service to view detailed information. For more information, see [View the details of a single asset](#).

- **Search by asset type**

Cloud services are classified into two asset types:

- **SLB**
- **NAT Gateway**

In the left-side pane on the **Cloud Product** tab, you can view the number of cloud services of each asset type. You can click **SLB** or **NAT** to view the security status of the target cloud service.

- **Search by tag**

In the **Tag** section in the left-side pane of the **Cloud Product** tab, you can view the number of assets bound to each tag. You can click a tag on the left of the asset list to view the security status of the cloud services to which this tag is bound.

- **Filter by search condition**

You can click **All cloud product**, **SLB**, or **NAT** in the left-side pane of the **Cloud Product** tab and configure search conditions in the search box to search for specific assets.

For example, you can click **All cloud product** and configure search conditions to search for specific assets.

- Use multiple subconditions to search for specific assets:

Select a condition from the drop-down list of the search box on the **Cloud Product** tab, and select a subcondition or enter a keyword into the search box to search for specific assets. Supported search conditions are **Internet IP**, **Instance name**, **Alert problems**, **Risk Status**, **Tag**, and **Group name**.

 **Note**

You can select multiple search conditions at a time and specify the Boolean operator between each search condition. The following content describes Boolean operators:

- Boolean operators:
 - **AND**: specifies the **AND** logical relation for the search conditions.
 - **OR**: specifies the **OR** logical relation for the search conditions.
- If you need to use one search condition and multiple keywords to search for specific servers, set the Boolean operator to **OR**.
- If the search condition requires you to enter a keyword, enter a keyword and click the **Search** icon. Results are displayed only after you click the **Search** icon.

- Use multiple search conditions to search for specific assets:

Apply multiple search conditions.

- You can click **SLB**, **NAT**, or a tag specified in the **Tag** section and configure conditions in the search box on the **Cloud Product** tab to search for specific assets.
- You can also click **All cloud product**, **SLB**, or **NAT** and select a tag specified in the **Tag** section to search for specific assets.

- **Set frequently used search conditions**

You can save applied search conditions as frequently used search conditions. Click **Save** below the search box and enter a name in the **Save condition** dialog box. Then, you can select the saved search condition from the **Frequent search conditions** drop-down list on the right of the search box.

28.4.7.3. View the details of a single asset

The **Assets** page provides details about all assets. These details include basic information, alert management status, baseline check analysis, and asset fingerprints. This topic describes how to view details of a single server or Alibaba Cloud service.

Context

The basic information about assets is displayed on the **Assets** page. Based on the types of assets, servers or Alibaba Cloud services can be managed in different ways.

The following table lists the features that are supported by servers and Alibaba Cloud services on the **Assets** page. The following content describes the marks that are used to indicate whether a feature is supported by servers or Alibaba Cloud services:

- x: not supported by this edition.
- √: supported by this edition.

Feature	Description	Server	Cloud service
	Risk state: displays the number of risks of an asset. Risks can be divided into the following types: <ul style="list-style-type: none"> ● Vulnerability ● Alert ● Baseline risk 	√	√ (Only alerts can be processed.)

Feature	Description	Server	Cloud service
Basic information	Detail: displays the configuration and protection status of an asset. You can specify a group and a tag for the asset.	√	√ (You cannot specify a group for the asset.)
	Asset investigation: displays asset fingerprints, including ports, software, processes, and accounts.	√	X
	Vulnerability check: displays the supported types of vulnerability checks. You can enable or disable different types of vulnerability checks for an asset.	√	X
	Logon security setting: displays the common logon locations, IP addresses, time, and accounts of an asset. You can also enable or disable alerts for the asset.	√	X
Vulnerabilities	Displays the vulnerability check results of an asset.	√	X
Alerts	Displays the security alerts of an asset.	√	√
Baseline Risks	Displays the baseline check results of an asset.	√	X
Asset Fingerprints	Displays the fingerprints of an asset.	√	X

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Assets** page, click the **Server** or the **Cloud Product** tab.
4. On the **Server** or **Cloud Product** tab, find the target asset and click its name.
5. View the details of the target asset.

On the asset details page, click the **Basic Information**, **Vulnerabilities**, **Alerts**, **Baseline Risks**, or **Asset Fingerprints** tab to view relevant details.

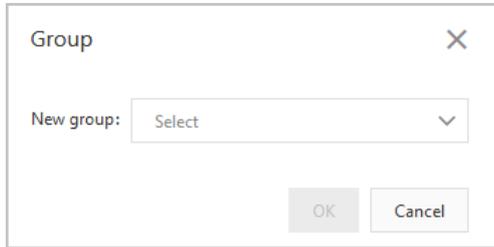
The following content displays the details of the target asset:

- **Basic Information:** This tab consists of sections where you can view asset details and manage an asset.
 - **Risk State:** This section displays information about vulnerabilities, alerts, and baseline risks of an asset. You can click the number under **Vulnerabilities**, **Alerts**, or **Baseline Risks** to view the details.

- **Detail:** This section displays information about the asset configuration and security protection settings, and allows you to manage asset tags and groups.

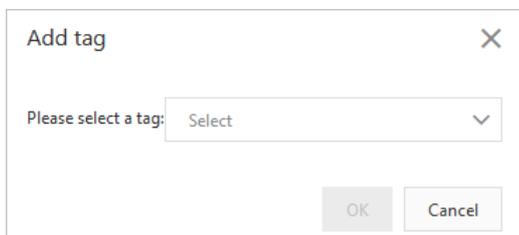
- **Change asset groups**

Click **Group**. In the **Group** dialog box, select a new group and click **OK**.



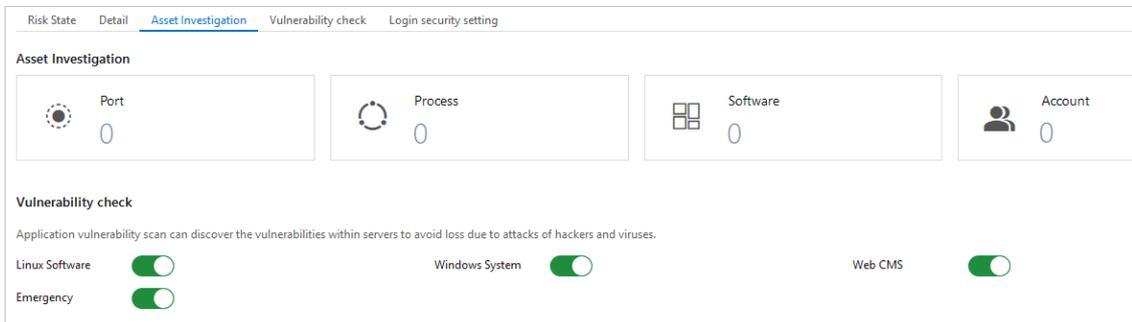
- **Modify tags**

Click the  icon. In the **Add tag** dialog box, select a tag and click **OK**.



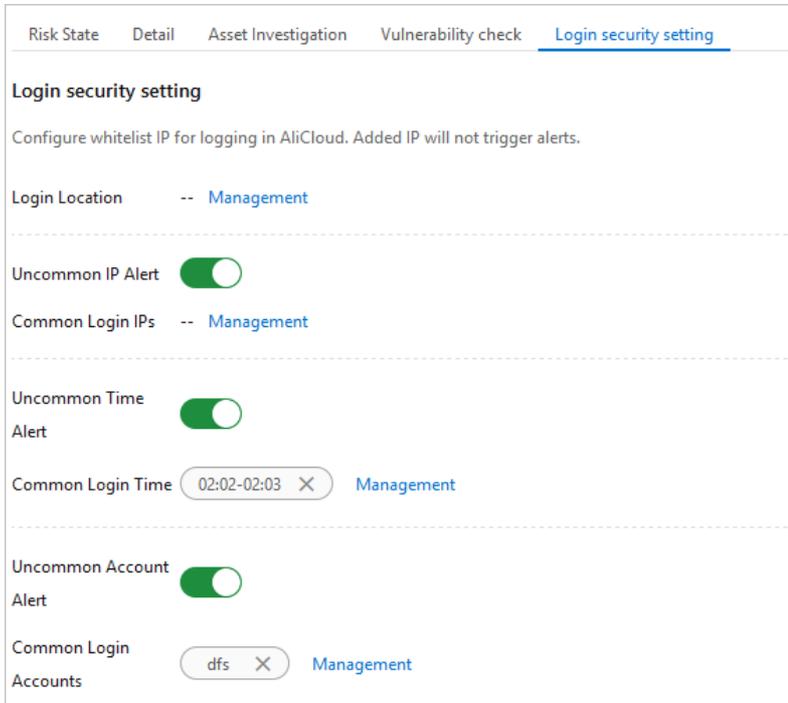
To remove the tag of an asset, click the  icon to the right of the tag.

- **Asset Investigation:** This section displays the fingerprints of an asset. You can click the number under an item to go to the **Asset Fingerprints** tab to view the details.



- **Vulnerability check:** This section displays vulnerability check items that are enabled or disabled for an asset. You can enable or disable different types of vulnerability checks for the asset. The vulnerabilities include Linux software vulnerabilities, Windows system vulnerabilities, Web CMS vulnerabilities, and emergency vulnerabilities.

- Login security setting:** This section allows you to add approved logon locations, configure advanced logon settings, turn on or turn off the unapproved IP address, time, and account alert function. The advanced logon settings include approved IP addresses or Classless Inter-Domain Routing (CIDR) blocks, time periods, and accounts. You can also add approved IP addresses, time periods, and accounts that are allowed to log on to a specific asset.



- Vulnerabilities:** This tab displays vulnerabilities of an asset.

Priority	Disclosure Time	Vulnerability	Related process	Vul (cve)	Status	Actions
High	Aug 10, 2020	RHSA-2018:1062-Important: kernel security, bug fix, and enhancement update		CVE-2016-3672 Total 30	Unfixed	Fix Verify Details
High	Aug 10, 2020	RHSA-2018:1453-Critical: dhcp security update		CVE-2018-1111	Unfixed	Fix Verify Details
High	Aug 10, 2020	RHSA-2018:3665-Important: NetworkManager security update		CVE-2018-15688	Unfixed	Fix Verify Details
High	Aug 10, 2020	RHSA-2017:3263-Moderate: curl security update		CVE-2017-1000257	Unfixed	Fix Verify Details

- Alerts:** This tab displays security alerts of an asset.

- Baseline Risks:** This tab displays baseline risks of an asset.

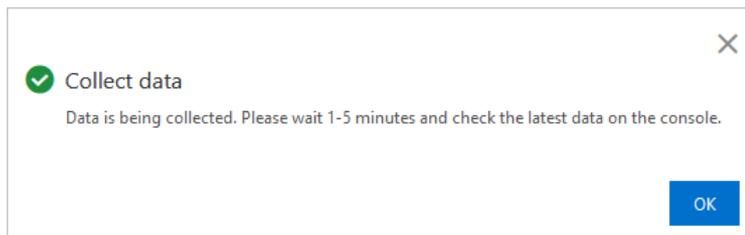
Severity	Baseline	Checked Item	Failed Items/Affected Servers	Category	Last Check
High	Alibaba Cloud Standard - CentOS Linux 7/8 Security Baseline Check	15	5 / 1	Best security practices	Aug 13, 2020, 00:35:11
High	Weak password - Linux system login weak password baseline	1	Risk free	Weak password	Aug 13, 2020, 00:35:11

- Asset Fingerprints:** This tab displays the fingerprints, including ports, processes, software, and accounts of an asset.

You can manually collect the latest fingerprints of an asset.

- You can click the **Port, Software, Process, Account, or Scheduled Tasks** tab. In the upper-right corner, click **Collect data now**.

b. In the **Collect data** dialog box, click **OK**.



After the data collection task is submitted, it takes one to five minutes to collect the fingerprints of the target asset. After the data collection task is complete, you can view the latest fingerprints of the target asset.

28.4.7.4. Enable and disable server protection

This topic describes how to enable and disable server protection.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, enable or disable server protection for specified servers.

- o **Enable server protection**

Select one or more servers where the agent is in the **Close** state, and choose **More operations > Turn on protection**.

After server protection is enabled, the status of the agent on the servers changes to **Enable**.

- o **Disable server protection**

You can disable server protection for specified servers. Select one or more servers where the agent is in the **Enable** state, and choose **More operations > Suspend Protection**.

Note After server protection is disabled, Apsara Stack Security Center stops providing protection for your servers. The protection mechanisms that are stopped include vulnerability detection and security event alerting. We recommend that you proceed with caution.

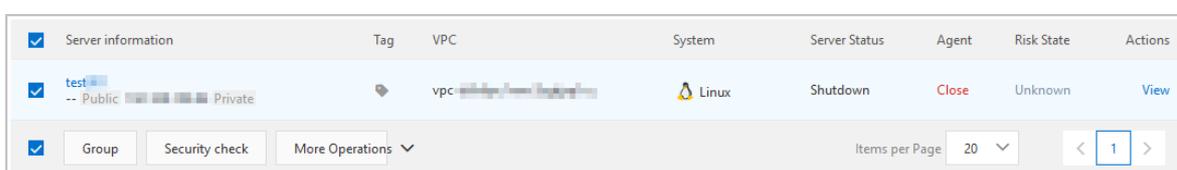
After server protection is disabled, the status of the agent on your servers changes to **Close**.

28.4.7.5. Perform a one-click security check

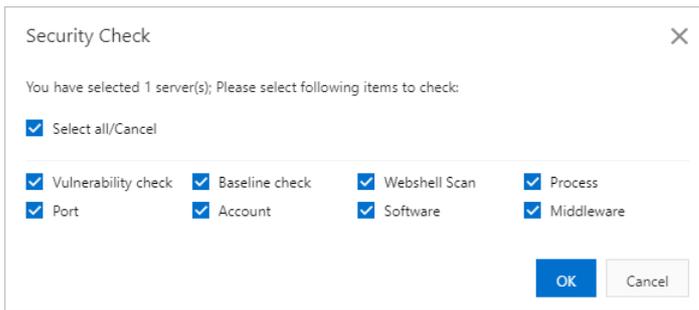
The **Server** tab of the **Assets** page allows you to run security checks. You can dispatch security check tasks to scan for vulnerabilities, baseline risks, and webshells, and collect asset fingerprints on a specific server. The asset fingerprints are ports, software, processes, and accounts. This topic describes how to perform security checks on servers.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, select one or more servers on which you want to perform security checks.
4. In the lower part of the page, click **Security check**.



5. In the Security Check dialog box, select check items.



6. Click OK to start the check.

7. In the dialog box that appears, click OK.



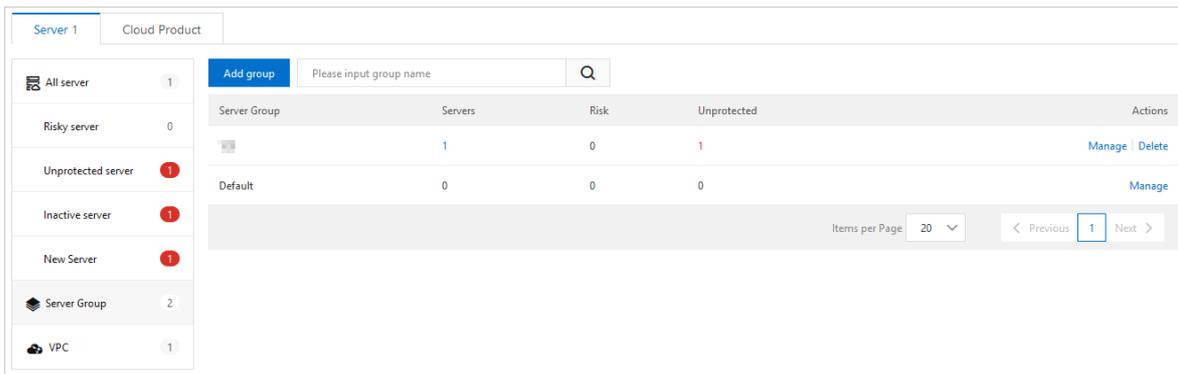
After the one-click security check is complete, the check results are automatically displayed on the Basic Information tab of the target asset.

28.4.7.6. Manage asset groups

This topic describes how to create, modify, delete, and replace server groups.

Add a group

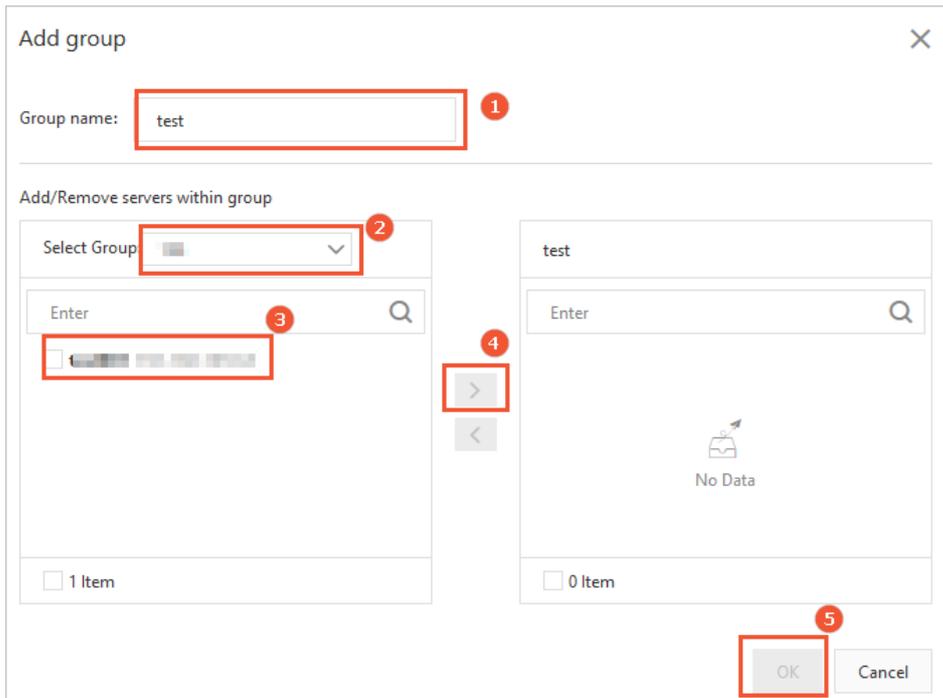
1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Threat Detection > Assets.
3. On the page that appears, click the Server tab. In the servers list on the left, click Server Group.



Note Ungrouped assets are in the Default group.

4. Click Add Group.

5. In the Add Group dialog box, configure the parameters.



You can perform the following steps to configure the parameters:

- i. Enter a group name in the **Group name** field.
- ii. Add assets to the new group.

You can add assets from the **Default** group to the new group. You can also move assets from another group to the new group. Select **Default** or other groups from the **Select Group** drop-down list, select assets in the group, and click the  icon to add the selected groups to the new group.

6. Click **OK**.

In the server group list, you can view the new group.

Modify or delete a server group

The following procedure shows how to modify a server group. You can rename a group, add assets to a group, or remove assets from a group.

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the page that appears, click the **Server** tab. In the servers list on the left, click **Server Group**.
4. Find the target group. In the **Actions** column, click **Manage** or **Delete**. Perform the following operations based on your business requirements:
 - **Modify a group**
 - a. In the **Actions** column, click **Manage** to open the **Group** dialog box.
 - b. In the **Group** dialog box, select a group from the **Select Group** drop-down list, select assets in the current group on the right, and click the  icon to add the selected groups to the new group on the left. You can also select assets in the group on the left and click the  icon to add the selected assets to the current group on the right.
 - c. Click **OK**.
 - **Delete a group**

To delete a group, click **Delete** in the **Actions** column. In the message that appears, click **OK**.

 **Note** After you delete a group, assets in this group are moved to the Default group.

Change server groups

You can add assets to an asset group to manage multiple assets at a time. We recommend that you add the same types of assets to an asset group. For example, when you configure a baseline check policy template, you can specify an asset group to apply the policy to all assets in the group. You can also filter and view assets based on asset groups in the assets list.

To add assets to a specific server group, perform the following steps:

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. On the **Server** tab, select one or more assets and click **Group**.
4. In the **Group** dialog box, select a new server group.
5. Click **OK**.

28.4.7.7. Manage asset tags

This topic describes how to add asset importance tags, and add, modify, and remove custom tags for a server.

Context

Apsara Stack Security Center provides the following asset importance tags to classify the assets. You can select appropriate importance tags for your assets.

An asset importance tag is transformed to the target asset importance score that is used to calculate the score of the urgency to fix a vulnerability. Asset importance score affects the score of the urgency to fix a vulnerability. You can determine whether to prioritize the repair of a vulnerability based on the score of the urgency to fix a vulnerability. If you add importance asset tags for core assets, Apsara Stack Security Center prompts you to fix vulnerabilities as soon as possible. The following table shows the relation between asset importance tags and asset importance scores.

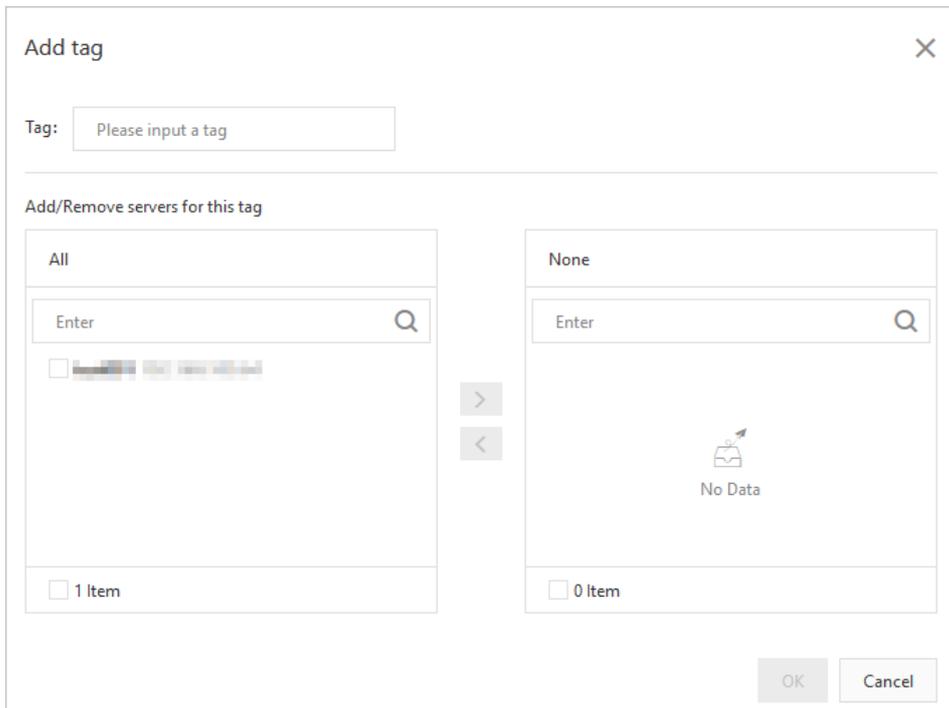
Type	Asset importance score	Recommendation
Important Assets	1.5	Assets that are related to crucial businesses and store core business data. Virus intrusion adversely affects the business system and causes major losses.
General Assets	1	Assets that are related to non-core businesses and are highly replaceable. Virus intrusion causes less impact on the system.
Test Assets	0.5	Assets for performance test, or assets that cause less impact on the system.

 **Note** If you do not add asset importance tags, the tag for each asset is set to **General Assets** by default, which specifies the asset importance score to 1.

Add a custom tag

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click **Server** or **Cloud Product**.
4. Click **Server** or **Cloud Product**. In the assets list on the left, click **Management** to the right of the **Tag** column.
5. In the **Add Tag** dialog box, enter the tag name. In the servers list on the left, select target servers, and

click the  icon to add the tag to the selected servers.



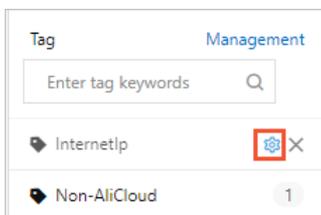
6. Click **OK**. In the left-side assets list, click the  icon in the **Tag** column to add a tag to the asset.

 **Note** You can add multiple tags to one asset. All tags of an asset are displayed in the **Tag** column.

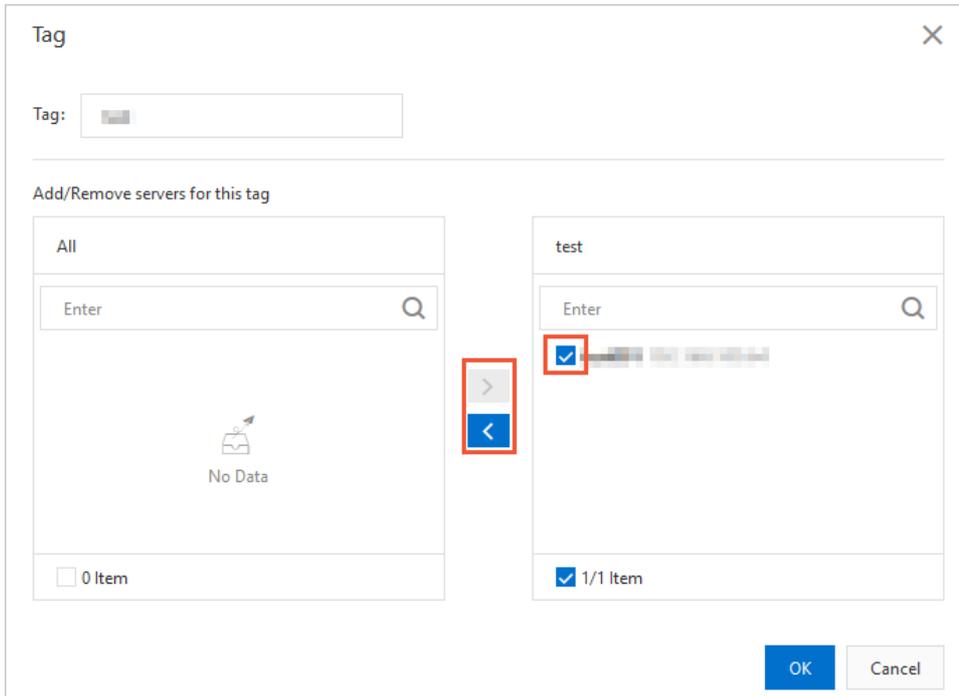
Modify and delete custom tags

Perform the following steps to modify or delete a tag:

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Threat Detection > Assets**.
3. Click **Server** or **Cloud Product**.
4. On the **Server** or **Cloud Product** tab, modify or delete a tag. Perform the following steps to modify or delete a tag:
 - o **Modify a tag**
 - a. If you want to modify a tag, move the pointer over the  icon to the right of the tag.



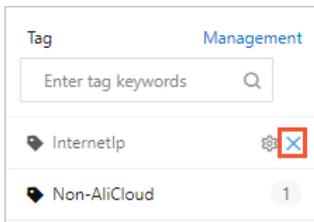
- b. In the Tag dialog box, enter a new name in the Tag field, add the tag to more servers, or remove the tag from servers.



- c. Click OK.

- o **Delete a tag**

Click the  icon in the Tag column. Click OK to delete a tag.



28.4.8. Create a security report

You can specify the report content, statistics types, and recipient email addresses. This way, you can view the security statistics of your assets. This topic describes how to create a security report.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Threat Detection > Security Reports.
3. On the Reports page, click Create Report.

 **Notice** In addition to the default security report created by Apsara Stack Security Center, you can also create a maximum of nine security reports.

4. On the Specify Basic Information tab, configure the parameters.

Threat Detection / Reports

← Reports

Specify Basic Information
Specify Reported Data

* Report Name:

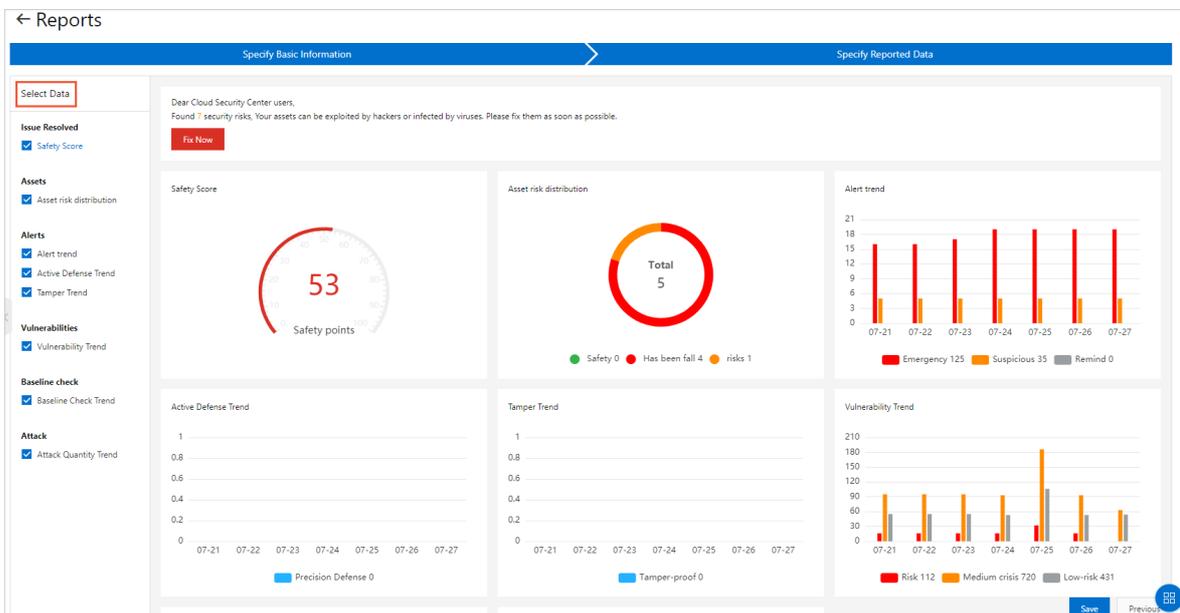
* Report Type: Daily ▼
Data Collection Period: Yesterday 00:00:00 to 24:00:00

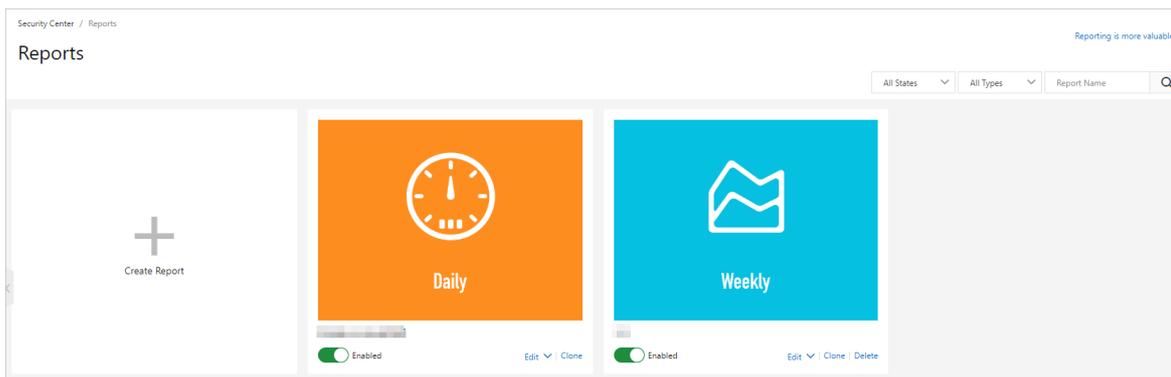
* Language: English ▼

Next

Configure the following parameters:

- **Report Name:** Enter a name for the security report.
 - **Report Type:** Select a report type from the drop-down list. Supported types are *Daily*, *Weekly*, *Monthly*, and *Custom*.
 If you select *Custom*, you must set the **Data Collection Period** parameter to specify the cycle when data is collected.
 - **Language:** Select a language for the security report.
5. Click **Next**.
6. On the **Specify Reported Data** tab, select the types of data that you want to view in the security report. You can select assets, alerts, vulnerabilities, baseline checks, attacks, and other data related to security operations.





28.5. Network Traffic Monitoring System

28.5.1. View traffic trends

This topic describes how to view the network traffic trends, inbound traffic statistics, and outbound traffic statistics.

Context

By analyzing traffic trends, the security administrator can obtain the throughput and the peaks and troughs of traffic periods. In addition, the security administrator can block traffic from malicious IP addresses by viewing the top five IP addresses with the most traffic.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Traffic Trend**.
3. In the upper-right corner of the Traffic Trends page, select the time range, which can be **Last 1 Hour**, **Last 24 Hours**, or **Last 7 Days**.
4. View network traffic information.
 - **Network traffic trends**
View the network traffic trends from the selected time range. The network traffic trends include inbound and outbound traffic measured in bit/s.
 - **Inbound Traffic**
View the information of Inbound Sessions, Inbound Applications, and Destination IPs with Most Requests.
 - **Outbound Traffic**
View the information of Outbound Sessions, Outbound Applications, and Source IPs with Most Requests.
5. (Optional) Click the  icon to export traffic trends as a PDF file.

28.5.2. View traffic at the Internet border

This topic describes how to view traffic at the Internet border. You can obtain up-to-date information about network security.

Prerequisites

The Network Traffic Monitoring System module is purchased and deployed at the egress (ISW) of Apsara Stack. This module audits, analyzes, and manages both inbound and outbound traffic at Internet borders.

Context

You can use traffic information to identify abnormal Internet traffic and block malicious traffic.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Network Security > Traffic Analysis > Internet Border.
3. Specify traffic filter conditions.

Item	Description
1	Specify the traffic direction. Valid values: <i>Inbound</i> and <i>Outbound</i> . <ul style="list-style-type: none"> ◦ <i>Inbound</i>: The traffic flows from the Internet to the internal network. ◦ <i>Outbound</i>: The traffic flows from the internal network to the Internet.
2	Specify whether you want to view the traffic from the IP address or application dimension. Valid values: <i>By IP</i> and <i>By Application</i> .
3	Specify the time range. Valid values: <i>Last 1 Hour</i> , <i>Last 24 Hours</i> , and <i>Last 7 Days</i> .

4. View details about the traffic at the Internet border.

o Traffic Statistics



- The Visits to IP section includes Source IPs, Destination IPs, Applications, and Traffic Risk.
- In the traffic chart on the right, you can view Average Traffic, Peak Traffic, and traffic trends.

o Traffic List

The Traffic List section features two tabs: 'By Source IP' (selected) and 'By Destination IP'. Below the tabs is a search bar with an 'Enter' field and a 'Search' button. The main area contains a table with the following columns: Source IP, Direction, Traffic Volume, Visited Destination IPs, Applications, Destination Ports, Sessions, and Actions. The table is currently empty, displaying a 'No Data' message.

In the Traffic List section, you can view traffic details.

5. In the Traffic List section, view abnormal traffic of the specified IP address.
 - o If *Inbound* is specified, you can view abnormal traffic on the **By Destination IP** tab of the Traffic List section.
 - o If *Outbound* is specified, you can view abnormal traffic in the Traffic List section.

28.5.3. View traffic at the internal network border

This topic describes how to view the traffic at the internal network border. You can obtain up-to-date information about network security based on the traffic.

Prerequisites

The Network Traffic Monitoring System module is purchased and deployed at the ingress (CSW) of Apsara Stack. This module is used to audit, analyze, and control both inbound and outbound traffic routed over leased lines between on-premises data centers and virtual private clouds (VPCs).

Context

You can use traffic information to identify suspicious traffic from the internal network and block malicious requests.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Traffic Analysis > Internal Network Border**.
3. Specify traffic filter conditions.

Item	Description
1	Select a VPC name from the drop-down list.
2	Specify the traffic direction. Valid values: <i>Inbound</i> and <i>Outbound</i> . <ul style="list-style-type: none"> ◦ <i>Inbound</i>: The traffic flows from the Internet to the internal network. ◦ <i>Outbound</i>: The traffic flows from the internal network to the Internet.
3	Specify whether you want to view the traffic that flows through the internal network border from the IP address or application dimensions. Valid values: <i>By IP</i> and <i>By Application</i> .
4	Specify the time range. Valid values: <i>Last 1 Hour</i> , <i>Last 24 Hours</i> , and <i>Last 7 Days</i> .

4. View details about the traffic at the internal network border.
 - **Traffic Statistics**
 - The **Visits to IP** section includes **Source IPs**, **Destination IPs**, **Applications**, and **Traffic Risk**.
 - In the traffic chart on the right, you can view **Average Traffic**, **Peak Traffic**, and traffic trends.
 - **Traffic List**

In the **Traffic List** section, you can view traffic details.

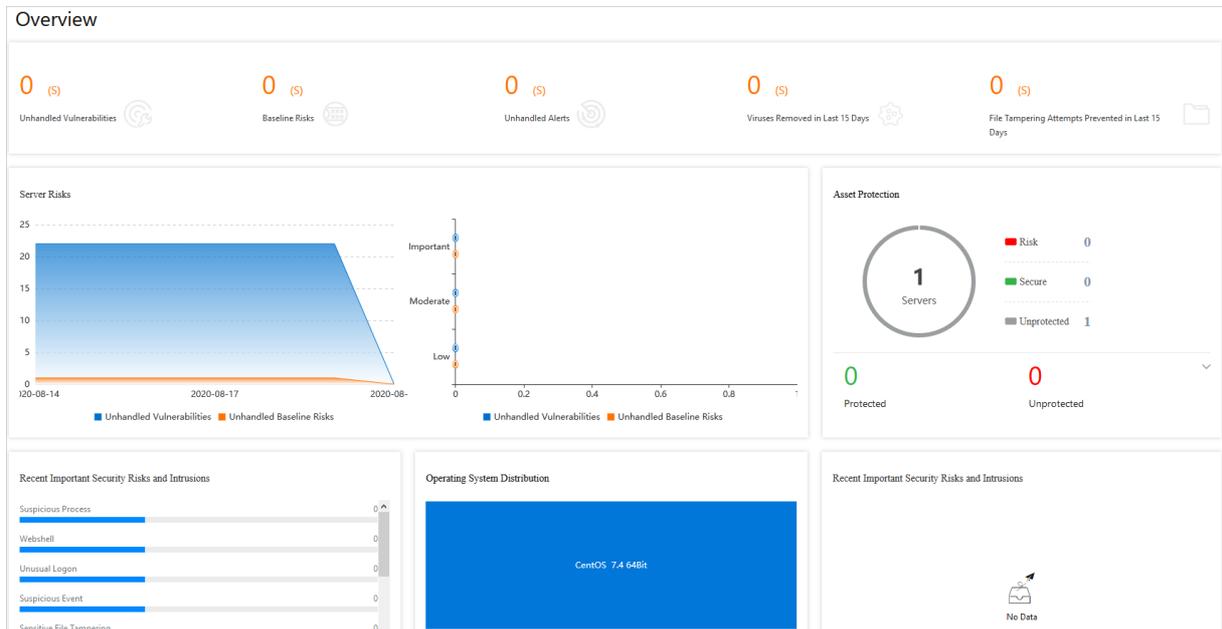
If **By IP** is specified, you can view the suspicious traffic of the specified IP address in the **Traffic List** section.

28.6. Server security

28.6.1. Server security overview

The security administrator can view the current security status of all servers on the server security overview page of Apsara Stack Security Center.

In the left-side navigation pane, choose **Server Security > Overview**. On the page that appears, you can view detailed information in the Overview, Server Risks, Asset Protection, Operating System Distribution, and Recent Important Security Risks and Intrusions sections.



- **Overview:** This section displays the number of security vulnerabilities of each type (including **Unhandled Vulnerabilities** and **Baseline Risks**) and the number of security events of each type (including **Unhandled Alerts**, **Viruses Removed in Last 15 Days**, and **File Tampering Attempts Prevented in Last 15 Days**) on servers.
- **Server Risks:** This section displays the number of unhandled vulnerabilities, the number of baseline risks, and the distribution of risk levels.
- **Asset Protection:** This section displays the number of protected servers and the number of offline servers.
- **Recent Important Security Risks and Intrusions:** This section displays the recent important risks and events on your servers. You can click a risk or event to view the details.

28.6.2. Server fingerprints

28.6.2.1. Manage listener ports

This topic describes how to regularly collect information from listener ports on a server, and record and view the port changes and historical port information. This allows you to locate suspicious listening behavior.

Context

This task is suitable to the following scenarios:

- Check for servers that listen on a specific port.
- Check for ports that are open on a specific server.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Port** tab.
3. **View Port, Protocol, and Server.** You can search for a port by using its port number, process name, server name, or IP address.
4. Click a port number to view the details, such as the assets and protocol.

28.6.2.2. Manage software versions

This topic describes how to regularly collect software version information of a server and record the changes. This helps to check your software assets.

Context

This task is suitable to the following scenarios:

- Check for software assets that are installed without authorization.
- Check for software of outdated versions.
- Locate the affected assets when vulnerabilities are detected.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Software** tab.
3. View all software in use and the number of servers that use such software. You can search for a piece of software by using its software name, version, installation directory, server name, or IP address.
4. Click a software name to view the details, such as the assets and software version.

28.6.2.3. Manage processes

This topic describes how to regularly collect the process information on a server and record changes. This helps check for processes and view historical process changes.

Context

This task is suitable to the following scenarios:

- Checks for servers that run a specified process.
- Checks for processes that run on a server.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Process** tab.
3. View all running processes and the number of servers that run these processes. You can search for a process by using its process name, running user, startup parameter, or server name or IP address.
4. Click a process name to view the details, such as the assets, path, and startup parameters.

28.6.2.4. Manage account information

This topic describes how to regularly collect the account information on a server and record changes. This helps check for accounts and view historical account changes.

Context

This task is suitable to the following scenarios:

- Check for servers where the specified account is created.
- Check for accounts that are created on a server.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Account** tab.
3. View all logged-on accounts and the number of servers that use these accounts. You can search for an account by using its account name, root permissions, server name, or IP address.

4. Click an account name to view the details, such as the assets, root permissions, and user group.

28.6.2.5. Manage scheduled tasks

This topic describes how to regularly collect information of scheduled tasks on a server. This allows you to check your tasks.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. On the page that appears, click the **Scheduled Tasks** tab.
3. View the paths of all tasks and the number of servers that run these tasks. You can search for a task by using its path, server name, or IP address.
4. Click a task path to view the details, such as the assets, executed command, and task cycle.

28.6.2.6. Set the server fingerprint collection frequency

You can set the frequency at which the data of running processes, system accounts, listening ports, and software versions is collected.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Servers > Server Fingerprints**. In the upper-right corner of the page that appears, click **Settings**.
3. Select the collection frequency from each drop-down list.
4. Click **OK** to complete the configuration.

28.6.3. Threat protection

28.6.3.1. Vulnerability management

28.6.3.1.1. Manage Linux software vulnerabilities

This topic describes how to manage Linux software vulnerabilities.

Context

Apsara Stack Security automatically scans the software that are installed on your servers based on the vulnerabilities provided in the Common Vulnerabilities and Exposures (CVE) list. It also sends you alerts about the detected vulnerabilities. In addition, Apsara Stack Security provides commands that are used to fix vulnerabilities and allows you to verify these vulnerability fixes.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Linux Software** tab.
3. View the detected Linux vulnerabilities.

 **Note** You can locate a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

 **Note** You can locate specific affected assets by using the search and filter functions.

- **Basic Information:** the basic information of the vulnerability, including the name, Common Vulnerability Scoring System (CVSS) score, description, and resolution.
 - **Affected Assets:** the servers that are affected by the vulnerability.
5. Select an action based on the impact of the vulnerability.

Actions on vulnerabilities

Option	Description
Generate Fix Command	Select this option to generate the commands that are used fix the vulnerability. You can then log on to the server to run these commands.
Fix Now	Select this option to fix the vulnerability.
Restarted and Verified	If a vulnerability fix takes effect only after a server reboot, you must reboot the server after the status of the vulnerability changes to Fixed (To Be Restarted). After the reboot, click Restarted and Verified.
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	Click Verify to verify the vulnerability fix. If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

28.6.3.1.2. Manage Windows vulnerabilities

This topic describes how to manage Window vulnerabilities.

Context

Apsara Stack Security automatically checks if your servers have the latest Microsoft updates installed, and notifies you of the detected vulnerabilities. Apsara Stack Security also automatically detects and fixes major vulnerabilities on your servers.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Window System** tab.
3. Check the detected Windows vulnerabilities.

 **Note** You can find a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

 **Note** You can find specific affected assets by using the search and filter functions.

- **Basic Information:** the basic information of the vulnerability, including the name, Common Vulnerability Scoring System (CVSS) score, description, and resolution.
 - **Affected Assets:** the servers that are affected by the vulnerability.
5. Select an action based on the impact of the vulnerability. [Actions on vulnerabilities](#) describes the actions.

Actions on vulnerabilities

Option	Description
Rectify	Select this option to fix the vulnerability. The system caches an official Windows patch in the cloud for your server to download and update.
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	Click Verify to verify the vulnerability fix.
Restarted and Verified	If a vulnerability fix takes effect only after a server reboot, you must reboot the server after the status of the vulnerability changes to Fixed (To Be Restarted) . After the reboot, click Restarted and Verified .

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

28.6.3.1.3. Manage Web CMS vulnerabilities

This topic describes how to manage Web CMS vulnerabilities.

Context

The Web CMS vulnerability detection feature obtains the information of the latest vulnerabilities and provides patches in the cloud. This helps you detect and fix vulnerabilities.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Web CMS** tab.
3. View all vulnerabilities.

 **Note** You can find a vulnerability by using the search and filter functions.

4. Click a vulnerability to go to the vulnerability details page. You can view details about the vulnerability and affected assets on this page.

 **Note** You can find specific affected assets by using the search and filter functions.

5. Select an action based on the impact of the vulnerability. [Actions on vulnerabilities](#) describes the actions.

Actions on vulnerabilities

Option	Description
--------	-------------

Option	Description
Rectify	<p>Select this option to fix the Web CMS vulnerability by replacing the web files that contain the vulnerability on your server.</p> <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note Before you fix the vulnerability, we recommend that you back up the web files affected by this vulnerability. For more information about the paths of the web files, see the paths specified in the vulnerability remarks.</p> </div>
Ignore	Select this option to ignore a vulnerability. The system does not send you an alert about an ignored vulnerability.
Verify	<p>Click Verify to verify the vulnerability fix.</p> <p>If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.</p>
Undo Fix	For vulnerabilities that have been fixed, click Undo Fix to restore the web files that have been replaced.

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

28.6.3.1.4. Manage emergency vulnerabilities

This topic describes how to manage emergency vulnerabilities.

Context

Apsara Stack Security automatically detects vulnerabilities on servers, such as the unauthorized Redis access vulnerability and Struts S2-052 vulnerability, and sends vulnerability alerts. After you fix a vulnerability, you can also check whether the fix is successful.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**. On the page that appears, click the **Emergency** tab.
3. **View all vulnerabilities.** You can quickly locate a vulnerability by using the search and filter functions.
4. Click a vulnerability to go to the vulnerability details page. You can view detailed vulnerability information and affected assets on this page. You can quickly locate specific affected assets by using the search and filter functions.
5. Select an action based on the impact of the vulnerability. [Actions on vulnerabilities](#) describes the actions.

Follow the instructions to manually fix the vulnerabilities on the **Emergency** tab.

Actions on vulnerabilities

Action	Description
Ignore	Ignore a vulnerability. The system does not alert you about an ignored vulnerability.
Verify	<p>Verify the fix after you manually fix a vulnerability.</p> <p>If you do not manually verify a fix, the system automatically verifies the fix within 48 hours after the vulnerability fix is complete.</p>

You can fix a vulnerability for one or more affected assets at a time.

- To fix a vulnerability for one affected asset, select an action from the **Actions** column of the asset.
- To fix a vulnerability for one or more affected assets, select the target servers, and select an action in the lower-left corner.

28.6.3.1.5. Configure vulnerability management policies

You can enable or disable automatic detection for different types of vulnerabilities, and enable vulnerability detection for specific servers. You can also set a time duration for which invalid vulnerabilities are retained, and configure a vulnerability whitelist.

Context

A vulnerability whitelist allows you to exclude vulnerabilities from the detection list. You can add multiple vulnerabilities in the vulnerability list to the whitelist. The system does not detect vulnerabilities in the whitelist. You can manage the vulnerability whitelist on the vulnerability settings page.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Vulnerabilities**.
3. In the upper-right corner, click **Settings** to configure vulnerability management policies.

Settings

Linux Software:	<input checked="" type="checkbox"/>	Total : 1, Scan-Disabled : 0	Manage
Windows System:	<input checked="" type="checkbox"/>	Total : 1, Scan-Disabled : 0	Manage
Web CMS:	<input checked="" type="checkbox"/>	Total : 1, Scan-Disabled : 0	Manage
Emergency:	<input checked="" type="checkbox"/>	Total : 1, Scan-Disabled : 0	Manage

Retain Invalid Vul for:

Vul scan level: High Medium Low

Vul Whitelist:

Vulnerability	Actions
No Data	

- Select a vulnerability type and enable or disable detection for vulnerabilities of this type.
- Click **Manage** next to a vulnerability type and specify the servers on which vulnerabilities of this type are detected.
- Select a time duration for which invalid vulnerabilities are retained: 7 days, 30 days, or 90 days.

Note If you do not take any action on a detected vulnerability, the system determines that the alert is invalid. The system deletes the vulnerability after the specified duration.

- Select the vulnerability severities for scanning.
 - **High:** Vulnerabilities of this severity must be fixed as soon as possible.

- **Medium:** Vulnerabilities of this severity can be fixed later.
- **Low:** Vulnerabilities of this severity do not need to be fixed for now.
- Select vulnerabilities in the whitelist and click **Remove** to enable the system to detect these vulnerabilities and send alerts again.

28.6.3.2. Baseline check

28.6.3.2.1. Baseline check overview

The baseline check feature automatically checks the security configurations on servers and provides the detailed check results and suggestions for baseline reinforcement.

Description

After you enable the baseline check feature, Apsara Stack Security automatically checks for risks related to the operating systems, accounts, databases, passwords, and security compliance configurations of your servers, and provides reinforcement suggestions. For more information, see [Baseline check items](#).

By default, a full baseline check is automatically performed from 00:00 to 06:00 every day. You can create and manage scan policies for baseline checks. When you create or modify a policy, you can customize the check items, interval, and time period of a baseline check, and select the servers to which you want to apply this policy. For more information, see [Add a custom baseline check policy](#).

Precautions

The following check items are disabled by default. To check these items, make sure that these items do not affect your business and select them when you customize a scan policy.

- Check items related to weak passwords for specific applications such as MySQL, PostgreSQL, and SQL Server

 **Note** If these check items are enabled, the system attempts to log on to servers with weak passwords. The logon attempts consume server resources and generate many logon failure records.

- Check items related to China classified protection of cybersecurity
- Check items related to the Center for Internet Security (CIS) standard

Baseline check items

Category	Check item
Database	Alibaba Cloud Standard - MongoDB Security Baseline Check
	Alibaba Cloud Standard - Redis Security Baseline Check
	Alibaba Cloud Standard - Oracle 11g Security Baseline Check
	Alibaba Cloud Standard - Memcached Security Baseline Check
	Alibaba Cloud Standard - Mysql Security Baseline Check

Category	Check item
Operating system	<p>Security baseline check against the Alibaba Cloud standard:</p> <ul style="list-style-type: none"> • Alibaba Cloud Aliyun Linux 2 Benchmark • Alibaba Cloud Standard - CentOS Linux 6 Security Baseline Check • Alibaba Cloud Standard - CentOS Linux 7/8 Security Baseline Check • Alibaba Cloud Standard - Debian Linux 8 Security Baseline • Alibaba Cloud Standard - Red Hat Enterprise Linux 6 Security Baseline Check • Alibaba Cloud Standard - Red Hat Enterprise Linux 7 Security Baseline Check • Alibaba Cloud Standard - Ubuntu Security Baseline • Alibaba Cloud Standard - Windows Server 2008 R2 Security Baseline Check • Alibaba Cloud Standard - Windows 2012 R2 Security Baseline • Alibaba Cloud Standard - Windows 2016/2019 R2 Security Baseline
	<p>Security baseline check against the CIS standard:</p> <ul style="list-style-type: none"> • Alibaba Cloud Aliyun Linux 2 CIS Benchmark • CIS CentOS Linux 6 LTS Benchmark • CIS CentOS Linux 7 LTS Benchmark • CIS Debian Linux 8 Benchmark • CIS Ubuntu Linux 14 LTS Benchmark • CIS Ubuntu Linux 16/18 LTS Benchmark • CIS Microsoft Windows Server 2008 R2 Benchmark • CIS Microsoft Windows Server 2012 R2 Benchmark • CIS Microsoft Windows Server 2016/2019 R2 Benchmark
	<p>Baseline check on compliance of China classified protection of cybersecurity level II:</p> <ul style="list-style-type: none"> • Aliyun Linux 2 Baseline for China classified protection of cybersecurity-Level II • CentOS Linux 6 Baseline for China classified protection of cybersecurity-Level II • CentOS Linux 7 Baseline for China classified protection of cybersecurity-Level II • Debian Linux 8 Baseline for China classified protection of cybersecurity-Level II • Redhat Linux 7 Baseline for China classified protection of cybersecurity-Level II • Ubuntu 14 Baseline for China classified protection of cybersecurity-Level II • Linux Ubuntu 16/18 Baseline for China classified protection of cybersecurity-Level II • Windows 2008 R2 Baseline for China classified protection of cybersecurity-Level II • Windows 2012 R2 Baseline for China classified protection of cybersecurity-Level II • Windows 2016/2019 R2 Baseline for China classified protection of cybersecurity-Level II

Category	Check item
	Baseline check on compliance of China classified protection of cybersecurity level III: <ul style="list-style-type: none"> • Aliyun Linux 2 Baseline for China classified protection of cybersecurity-Level III • CentOS Linux 6 Baseline for China classified protection of cybersecurity-Level III • CentOS Linux 7 Baseline for China classified protection of cybersecurity-Level III • Debian Linux 8 Baseline for China classified protection of cybersecurity-Level III • Redhat Linux 6 Baseline for China classified protection of cybersecurity-Level III • China's Level 3 Protection of Cybersecurity - Red Hat Enterprise Linux 7 Compliance Baseline Check • SUSE Linux 10 Baseline for China classified protection of cybersecurity-Level III • SUSE Linux 11 Baseline for China classified protection of cybersecurity-Level III • SUSE Linux 12 Baseline for China classified protection of cybersecurity-Level III • Ubuntu 14 Baseline for China classified protection of cybersecurity-Level III • Linux Ubuntu 16 Baseline for China classified protection of cybersecurity-Level III • Windows 2008 R2 Baseline for China classified protection of cybersecurity-Level III • Windows 2012 R2 Baseline for China classified protection of cybersecurity-Level III • Windows 2016/2019 R2 Baseline for China classified protection of cybersecurity-Level III
Weak password	Weak password - Linux system login weak password baseline
	Weak password - SQL Server DB login weak password baseline
	Weak password - PostgreSQL DB login weak password baseline
	Weak password - Windows system login weak password baseline
	Weak password - Ftp login weak password baseline
	Weak password - Mysql DB login weak password baseline
Middleware	Alibaba Cloud Standard - IIS 8 Security Baseline Check
	Alibaba Cloud Standard-Apache Tomcat Security Baseline
	Alibaba Cloud Standard - Apache Security Baseline Check
	Alibaba Cloud Standard - Nginx Security Baseline Check

28.6.3.2.2. Configure baseline check policies

This topic describes how to add, modify, and delete baseline check policies and how to set baseline check levels.

Context

By default, the baseline check feature uses the **default policy** to check the baseline security of assets. You can also customize baseline check policies based on your business requirements, for example, to check the compliance with China classified protection of cybersecurity-Level II.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Baseline Check**.
3. In the upper-right corner of the page that appears, click **Manage Policies**. In the **Manage Policies** pane, add, modify, or delete a baseline check policy, or modify the default policy.

- In the upper-right corner of the pane, click **Create Policy** to customize a baseline check policy. Then, click **Ok**.

Parameter	Description
Policy Name	Enter a policy name.
Schedule	Select a time interval for scheduled scan tasks from: 1 Day(s), 3 Day(s), 7 Day(s), and 30 Day(s), which represent every second day, every fourth day, every eighth day, and every thirty-first day. You can also select a time period for scheduled scan tasks from: 00:00 to 06:00, 06:00 to 12:00, 12:00 to 18:00, and 18:00 to 24:00.
Check Items	Select the baseline items that need to be checked under these categories: High risk exploit, CIS and China's Protection of Cybersecurity, Best security practices, and Weak password.
Servers	Select the asset groups to which you want to apply this policy. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> Note Newly purchased servers belong to Default under Asset Groups. To apply this policy to new servers, select Default.</p> </div>

- Click **Edit** or **Delete** next to the target policy to modify or delete it.

 **Note** You cannot restore a policy after it is deleted.

- Click **Edit** in the **Actions** column next to the **Default** policy to modify the asset groups to which the default policy is applied.

 **Note** You cannot delete the default policy or modify the check items of the default policy. You can only modify the asset groups to which the default policy is applied.

- In the lower part of the **Manage Policies** pane, set the baseline check level to High, Medium, and Low.

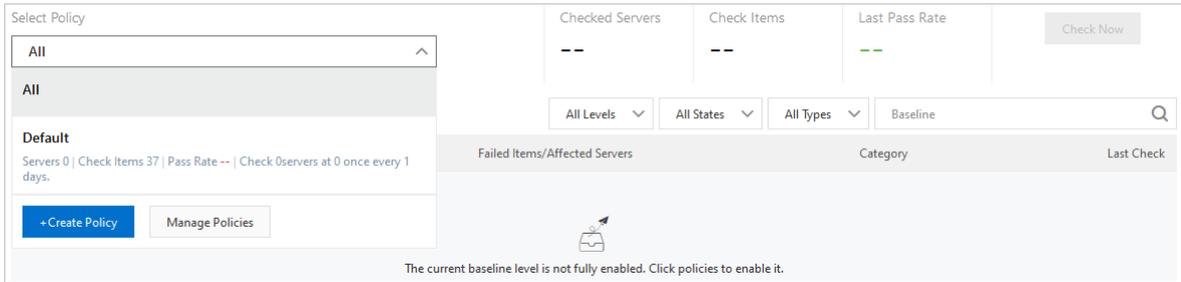
4. Click **Ok**.

28.6.3.2.3. View baseline check results and manage failed check items

The Apsara Stack Security console provides detailed baseline check results and suggestions on how to manage failed check items. This topic describes how to view baseline check results and manage failed check items in the Apsara Stack Security console. The check results include affected assets, checked items, and the suggestions.

View the summary of baseline check results

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Baseline Check**.
3. In the upper part of the **Baseline Check** page, view the summary of baseline check results. You can filter data by policy.



You can select a policy from the **Select Policy** drop-down list to view the following information:

- **Checked Servers:** The number of servers on which the baseline check runs. These servers are specified in the selected baseline check policy.
- **Check Items:** The number of Check Items specified in the selected baseline check policy.
- **Last Pass Rate:** The pass rate of the last baseline check.

If the number under **Last Pass Rate** is green, the pass rate of the checked servers is high. If this number is red, a large number of failed check items have been detected on the checked servers. We recommend that you view the check result details and manage the failed check items.

View all check items

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Baseline Check**.
3. Select **All** from the **Select Policy** drop-down list.
The **Baseline Check** page displays check result details, including **Baseline**, **Checked Item**, **Failed Items/Affected Servers**, **Category**, and **Last Check**.

Note You can also select a baseline check policy from the **Select Policy** drop-down list to view the check items specified in this policy.

View details of a check item

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Threat Prevention > Baseline Check**.
3. In the **Baseline** column, click the target check item to view its details.
Baseline details include the affected assets and numbers of **Passed Items** and **At-Risk Items**.
4. On the details page, manage the failed check items.
 - Find the target asset and click **View** in the **Actions** column to open the **At-Risk Items** pane.
 - You can click **Verify** in the **Actions** column to check whether the baseline risks of an asset have been managed. If the verification is passed, the number of **At-Risk Items** is reduced, and the status of the check items change to **Passed**.

View failed check items

1. Open the check item details page. Find the target asset and click **View** in the **Actions** column to view failed check items.
You can view the check items of the asset and the statuses of the check items (**Passed** or **Failed**).
2. You can click **Details** in the **Actions** column to view the description, result, and suggestion for this check item.

High Ensure password expiration period is set. | Identity authentication ✕

Description

Set password expiration time, force regular modification of password, reduce password leakage and guess risk. Use non-password login (e.g. key pair) please ignore this item.

Result

--

Suggestion

Use non-password login (e.g. key pair) please ignore this item. Set the PASS_MAX_DAYS parameter between 60 and 180 in the /etc/login.defs file:

```
PASS_MAX_DAYS 90
```

Meanwhile, execute the following commands to modify the root user settings:

```
chage --maxdays 90 root
```

Record the security enhancement operations, or back up the related data before the operation.

Ok

? **Note** We recommend that you follow the suggestions to manage Failed check items at the earliest opportunity, especially the high-risk check items.

Manage failed check items

In the At-Risk Items pane, manage failed check items as required.

- **Add a check item to the whitelist**

If you want to disable alerts for a check item, click **Whitelist** to add the check item to the whitelist. Check items in the whitelist do not trigger alerts.

? **Note** You can also select multiple check items and click **Whitelist** in the lower-left corner to add the check items to the whitelist at a time.

- **Remove a check item from the whitelist**

If you want to enable alerts for a check item in the whitelist, you can click **Remove** to remove the check item from the whitelist. You can remove one or more check items from the whitelist at a time. After a check item is removed from the whitelist, the check item triggers alerts again.

- **Verify a fixed check item**

After you fix a baseline risk, you can click **Verify** to check whether the risk has been fixed. After you click **Verify**, the status of the check item changes to **Verifying**.

If you do not manually perform the verification, Apsara Stack Security automatically verifies the check item based on the detection interval specified in the policies.

If the verification is passed, the **Status** of the check item changes to **Passed**.

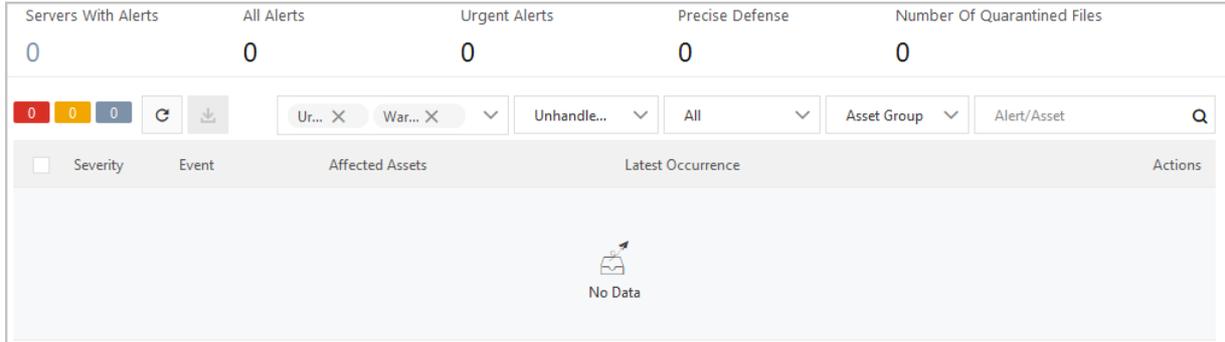
28.6.4. Intrusion detection

28.6.4.1. Intrusion events

28.6.4.1.1. Intrusion event types

If the Server Security feature detects sensitive file tampering, webshells, unusual logons, suspicious processes, and malicious processes, it generates alerts. Based on these alerts, you can monitor the security status of your assets and handle potential threats at the earliest opportunity.

Apsara Stack Security Center provides statistics based on enabled alerts and defense items. This allows you to understand enabled and disabled defense items. You can view statistics on alerts and information about defense items. To achieve this purpose, choose **Server Security > Intrusion Prevention > Intrusions**.



Alert types

The following table describes the defense items.

Alert	Description
Suspicious Process	Detects whether suspicious processes exist.
Webshell	<p>Uses engines developed by Alibaba Cloud to scan common webshell files. Apsara Stack Security Center supports scheduled scan tasks, provides real-time protection, and quarantines webshell files.</p> <ul style="list-style-type: none"> Apsara Stack Security Center scans the entire web directory early in the morning on a daily basis. A change made to files under the web directory triggers dynamic detection. You can specify the assets on which Apsara Stack Security Center scans for webshells. You can quarantine, restore, and ignore detected Trojan files.
Unusual Logon	<p>Detects unusual logons to your servers. You can specify approved logon IP addresses, time periods, and accounts. Logons from unapproved IP addresses, accounts, or time periods trigger alerts. You can manually add approved logon locations or configure the system to automatically update approved logon locations. You can also specify assets on which alerts are triggered when unusual logon locations are detected.</p> <p>Apsara Stack Security Center can detect the following logon events:</p> <ul style="list-style-type: none"> Logons to Elastic Compute Service (ECS) instances from unapproved IP addresses Logons to ECS instances from unapproved locations Execution of unusual commands after logons to ECS instances by using Secure Shell (SSH) ECS instances passwords cracked due to brute-force attacks based on the SSH protocol
Sensitive File Tampering	Checks whether sensitive files on your servers are maliciously modified, such as tampering of pre-loaded configuration files in Linux shared libraries.

Alert	Description
Malicious Process	<p>Dynamically scans your servers based on the anti-virus mechanism of Alibaba Cloud and Apsara Stack Security Center, and generates alerts if viruses are detected. Process information is collected by Apsara Stack Security Center and uploaded to Alibaba Cloud. You can manage detected viruses in the Apsara Stack Security Center console.</p> <p>Apsara Stack Security Center can detect the following malicious activities and processes:</p> <ul style="list-style-type: none"> • Visiting malicious IP addresses • Mining programs • Self-mutating Trojans • Malicious programs • Trojans
Unusual Network Connection	Detects disconnections or unusual network connections.
Suspicious Account	Detects logons to your assets from unapproved accounts.
Application intrusion event	Detects intrusion activities that use system application components.
Precision defense	The Virus Removal feature provides precise defenses against a majority of ransomware, distributed denial-of-service (DDoS) Trojans, mining programs, Trojan programs, malicious processes, webshells, and worms.
Cloud threat detection	Detects threats in other Alibaba Cloud services.
Persistence	Detects suspicious scheduled tasks on servers and generates alerts when persistent threats against the servers are detected.
Web Application Threat Detection	Detects server intrusions that use web applications.
Malicious scripts	Checks whether the system services of your assets are attacked or modified by malicious scripts. Alerts are generated if potential script attacks are detected.
Other	Detects other types of attacks, such as DDoS attacks.

28.6.4.1.2. View and handle intrusion events

This topic describes how to view and handle detected alert events on the Intrusions page.

Background information

After alert events are detected, they are displayed on the Intrusions page. You can choose **Server Security > Intrusion Prevention > Intrusions** to go to the Intrusions page.

If the alert events are not handled, they are displayed in the **Unhandled Alerts** list on the Intrusions page. After the alert events are handled, the status changes from **Unhandled Alerts** to **Handled**.

 **Note** Apsara Stack Security Center retains the records of **Unhandled Alerts** and **Handled** on the Intrusions page. By default, the records of **Unhandled Alerts** are displayed.

View alert events

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. In the intrusions list, you can view or search for detected intrusions, alert events, and relevant details.

Handle alert events

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions.**
3. On the **Intrusions** page, find the target alert event. In the **Actions** column, click **Processing**. In the dialog box that appears, configure parameters and click **Process Now**.

 **Note** If the alert event contains multiple related exceptions, click **Processing** and the details page appears. You can handle different exception events.

- **Ignore:** If you ignore this alert, the status of this alert changes to **Handled**. This alert event no longer triggers alerts if Server Guide detects it.
- **Whitelist:** If the alert is a false positive, you can add the alert event to the whitelist. After you add the alert event to the whitelist, the status of the alert changes to **Handled**. This event no longer triggers alerts if Server Guide detects it. In the **Handled** alert list, you can click **Cancel whitelist** to remove the target alert event from the whitelist.

 **Note** A false positive represents that Apsara Stack Security Center generates false alerts on a normal process. Common false positives include suspicious processes that send TCP packets. Common false positives notify you that suspicious scan activities on other devices are detected on your servers.

4. (Optional) If you confirm that one or more alert events are false positives or need to be ignored, go to the **Intrusions** page, select the target alert events, and then click **Ignore** or **Whitelist**.

Export alert events

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions.**
3. In the upper-left corner of the **Intrusions** page, click the  icon to export the alert list. After the alert list is exported, the **Done** dialog box appears in the upper-right corner.
4. In the **Done** dialog box, click **Download**.
The alert list is downloaded as an Excel file.

28.6.4.1.3. View exceptions related to an alert

Server Guide supports automatic analysis of exceptions related to an alert. You can click an alert name on the alert list to view and manage all exceptions related to this alert and view the results of automatic attack tracing.

Context

- Security Center automatically associates alerts with exceptions in real time to detect potential threats.
- The related exceptions of an alert are listed by time. This allows you to easily analyze and manage the exceptions to improve the emergency response mechanism.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions.**
3. On the **Intrusions** page, click the target alert name. The alert details page appears.
4. On the alert details page, view the details and related exceptions of the alert and manage the exceptions.
 - View alert details

You can view the assets affected by this alert, the first and latest occurrence time of the alert, and the details of the related exceptions.

- View affected assets

Click the name of an affected asset to view the details of the asset. These details include alerts, vulnerabilities, baseline risks, and asset fingerprints.

- View and manage related exceptions

In the **Related Exceptions** section, you can view the details and recommended solutions of all exceptions related to this alert.

- Click **Note** to the right of an exception name to add notes for the exception.
- Click the  icon to the right of a note to delete the note.

28.6.4.1.4. Use the file quarantine function

Server Guard can quarantine malicious files. Quarantined files are listed in the quarantine box on the Intrusions page. The system automatically deletes a quarantined file 30 days after it is quarantined. You can restore a quarantined file with a few clicks before it is deleted. This topic describes how to view quarantined files and remove files from the quarantine box.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. In the upper-right corner of the **Intrusions** page, click **Quarantine**.

You can perform the following operations in the **Quarantine** pane:

- View information about quarantined files. The information includes server IP addresses, directories that store the files, file status, and time of the last modification.
- Click **Restore** in the **Actions** column to remove a file from the quarantine box. The restored file appears in the alert list again.

28.6.4.1.5. Configure security alerts

This topic describes how to configure security alerts in Server Guard to specify common logon locations, customize scan targets, and manage advanced logon settings and security alerts.

Context

Server Guard supports advanced logon settings and security alerts. You can configure more fine-grained logon detection rules. For example, you can specify valid logon IP addresses, logon time, and logon accounts to block unauthorized requests sent to your assets.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Intrusion Prevention > Intrusions**.
3. In the upper-right corner, click **Settings**. Select the target tab and configure the following parameters:

- **Add a common logon location**
 - a. Click **Management** to the right of **Login Location**.
 - b. Select the logon location that you want to approve, and select the servers that are allowed to be logged on to from the approved location.
 - c.

Server Guard allows you to edit and delete common logon locations.

- **Configure advanced logon settings**

 **Note** After you configure advanced logon settings, alerts are triggered if your assets receive requests from unapproved locations. You can specify the IP addresses, accounts, and time periods that are allowed for logons to your assets. After the advanced logon settings are configured, Server Guard sends you alerts if your assets receive unauthorized logon requests. The procedure of configuring advanced logon settings is similar to configuring common logon locations. You can follow the preceding procedure to add, edit, and delete advanced logon settings.

- Turn on or turn off **Uncommon IP Alert** to the right of **Common Login IPs**. Alerts are triggered if your assets receive requests from unauthorized IP addresses.
- On the right of **Common Login Time**, turn on or turn off **Uncommon Time Alert**. Alerts are triggered if your assets receive requests at unauthorized times.
- On the right of **Common Login Accounts**, turn on or turn off **Uncommon Account Alert**. Alerts are triggered if your assets receive requests from unauthorized accounts.

- **Add a scan target**

Server Guard automatically scans directories of your servers and runs dynamic and static scan tasks. You can also manually add directories of servers.

- a. On the right of **Add Scan Targets**, click **Management**.
- b. Specify a valid directory, and select the servers on which the specified directory is scanned.

 **Note** To ensure the performance and efficiency, do not specify a root directory.

- c.

28.6.4.1.6. Virus removal

The virus removal feature provided by Server Guard is integrated with major antivirus engines worldwide. It detects viruses against large amounts of threat intelligence data provided by Alibaba Cloud. Virus removal also provides an exception detection module designed by Alibaba Cloud to detect viruses based on machine learning and deep learning. This way, virus removal can provide full-scale and dynamic antivirus protection to safeguard your servers.

The cloud threat detection feature can scan millions of files on a daily basis and is currently protecting millions of assets on the cloud.

Detection capabilities

The virus removal feature uses the Server Guard client to collect process information, and then scans the retrieved data for viruses in the cloud. If a malicious process is detected, you can stop the process and quarantine the related files.

- **Deep learning engine (developed by Alibaba Cloud):** The deep learning engine is built on deep learning technology and a large amount of attack samples. The engine specializes in detecting malicious files in the cloud and automatically identifies potential threats. It provides additional detection capabilities compared to traditional antivirus engines.
- **Cloud sandbox (developed by Alibaba Cloud):** It allows you to simulate cloud environments and monitor attacks launched by malicious samples. Based on big data analytics and machine learning modeling techniques, cloud sandbox automatically detects threats and offers dynamic analysis and detection capabilities.
- **Integration with major antivirus engines:** The cloud threat detection feature is integrated with major antivirus engines worldwide. Its virus library is updated in real time.
- **Threat intelligence detection:** Based on the threat intelligence data provided by Alibaba Cloud Security, cloud threat detection works with the exception detection module to detect malicious processes and operations.

Detectable virus types

Cloud threat detection is one of the best practices tested by Alibaba Cloud Security technologies and specialists. It provides end-to-end security services, including threat intelligence collection, data masking, threat identification, threat analysis, and malicious file quarantine and restoration. You can quarantine and restore data that has viruses in the Security Center console.

Cloud threat detection can detect the following types of viruses:

Virus	Description
Mining programs	A mining program consumes server resources without authorization to mine virtual currencies.
Computer worms	A computer worm uses computer networks to replicate itself and spread to a large number of computers within a short period of time.
Ransomware	Ransomware, such as WannaCry, uses encryption algorithms to encrypt files and prevent users from accessing the files.
Trojans	A Trojan is a program that allows the attacker to access information about the server and users, to gain control of the server, and to consume system resources.
DDoS Trojans	A DDoS Trojan hijacks servers and uses zombie servers to launch DDoS attacks, which can interrupt your workloads.
Backdoors	A backdoor is a malicious program injected by an attacker, who uses the backdoor to control the server or launch attacks.
Computer viruses	A computer virus inserts malicious code into other programs, and may replicate and infect the whole system.
Malicious programs	Programs that may pose a threat to the system and data security.

Benefits

- **Independent development and controllability:** Cloud threat detection is based on deep learning, machine learning, and big data analytics with a large amount of attack and defense practices. It uses multiple detection engines to protect your assets against viruses without delay.
- **Lightweight:** Cloud threat detection only takes 1% CPU usage and 50 MB of memory.
- **Dynamic:** Cloud threat detection dynamically retrieves log data to monitor the launches of malicious programs.
- **Easy to manage:** You can manage all servers and view their status at any time in the Security Center console.

Threat detection limits

Apsara Stack Security Center allows you to process security alerts, scan for vulnerabilities, analyze attacks, and check security settings in the Security Center console. Apsara Stack Security Center can analyze alerts and automatically trace attacks. This allows you to protect your assets. Apsara Stack Security Center supports a wide range of protection features. We recommend that you also install the latest system patches on your server. We also recommend you to use security services, such as Cloud Firewall and Web Application Firewall (WAF), to better protect your assets against attacks.

 **Note** Attacks and viruses are evolving, and your business environments vary. Security breaches may occur. We recommend that you use the alerting, vulnerability detection, baseline check, and configuration assessment features provided by Apsara Stack Security Center to better protect your assets against attacks.

28.6.4.2. Website tamper-proofing

28.6.4.2.1. Overview

Tamper protection monitors website directories in real time, restores modified files or directories, and protects websites from trojans, hidden links, and uploads of violent and illicit content.

Background information

To make illegal profits or conduct business attacks, attackers exploit vulnerabilities in websites to insert illegal hidden links and tamper with the websites. Defaced web pages affect normal user access and may lead to serious economic losses, damaged brand reputation, or political risks.

Tamper protection allows you to add Linux and Windows processes to the whitelist and update protected files in real time.

How tamper protection works

The Security Center agent automatically collects the list of processes that attempt to modify files in the protected directories of the protected servers. It identifies unusual processes and file changes in real time and blocks unusual processes.

The alert list is displayed on the Tamper Protection page. You can view unusual file changes, the corresponding processes, and the number of attempts made by each process in the alert list. If a file is modified by a trusted process, you can add the process to the whitelist. After the process is added to the whitelist, tamper protection no longer blocks the process. In scenarios where the content of websites, such as news and education websites, is frequently modified, the whitelist saves you the effort of frequently enabling and disabling tamper protection.

Versions of operating systems and kernels supported by tamper protection

OS	Supported operating system version	Supported kernel version
Windows	Windows Server 2008 and later	All versions
CentOS	6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, and 7.6	<ul style="list-style-type: none"> • 2.6.32-x • 3.10.0-x
Ubuntu	14, 16, and 18	<ul style="list-style-type: none"> • 3.13.0-32-generic • 3.13.0-86-generic • 4.4.0-62-generic • 4.4.0-63-generic • 4.4.0-93-generic • 4.4.0-151-generic • 4.4.0-117-generic • 4.15.0-23-generic • 4.15.0-42-generic • 4.15.0-45-generic • 4.15.0-52-generic

 **Note**

- The preceding table lists kernel versions supported by tamper protection. Servers that use an unsupported kernel version cannot use tamper protection. Make sure that your server uses a supported kernel version. If a kernel version is not supported, you must upgrade it to a supported version. Otherwise, you cannot add processes to the whitelist.
- Before you upgrade the server kernel, back up your asset data.

28.6.4.2.2. Configure tamper protection

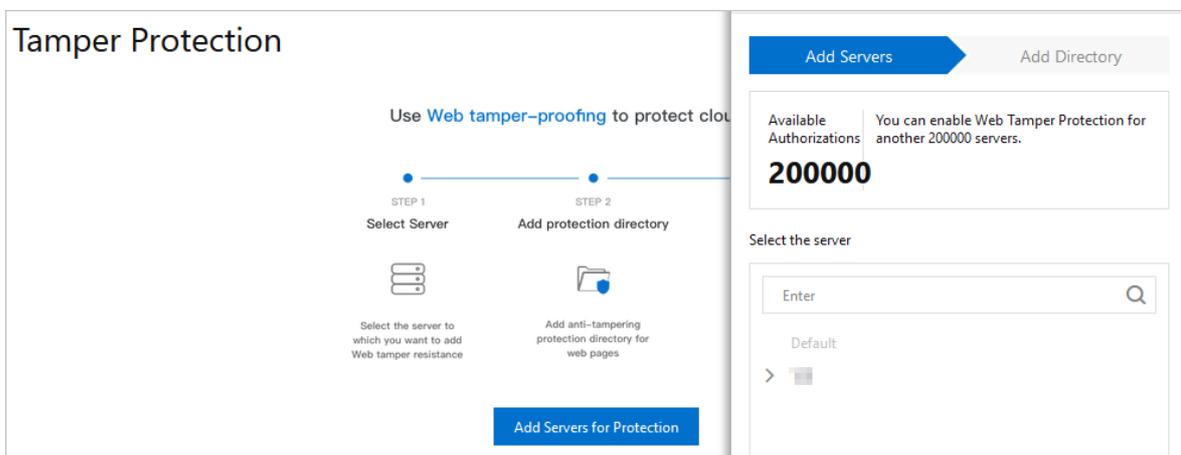
The Server Security feature allows you to configure tamper protection for web pages.

Limits

- For each server, you can add a maximum of 10 directories for protection.
- The protected directories of a Windows server must meet the following requirements: The maximum size of each directory is no more than 20 GB. Each directory contains a maximum of 2,000 folders. The maximum directory level is 20. The maximum size of each file is 3 MB.
- The protected directories of a Linux server must meet the following requirements: The maximum size of each directory is no more than 20 GB. Each directory contains a maximum of 3,000 folders. The maximum directory level is 20. The maximum size of each file is 3 MB.
- Before you add a directory for protection, make sure that the directory level, the number of folders, and the directory size meet the preceding requirements.
- We recommend that you exclude file formats that do not require protection, such as *.log*, *.png*, *.jpg*, *.mp4*, *.avi*, and *.mp3*. Separate multiple file formats with semicolons (;).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > File Tamper Protection**.
3. On the **Tamper Protection** page, click the **Management** tab.
4. On the **Tamper Protection** page, click **Add Servers for Protection**.
5. In the **Add Servers for Protection** dialog box, select a server that you want to protect.



6. Click **Next** to go to the **Add Directory** step.
7. In the **Add Directory** step, configure the following parameters:

Add Servers for Protection
✕

Add Servers
Add Directory

We recommend that you use the whitelist mode. In this mode, the file formats that usually require protection have been added to the protection list by default. You can add more directories and file formats for protection. [Blacklist Mode >](#)

* Protected Directory ⓘ

Enter or select the directory to be protected. the directory currer

* Protected File Formats ⓘ

php ✕
jsp ✕
asp ✕
aspx ✕

js ✕
cgi ✕
html ✕
htm ✕

xml ✕
shtml ✕
shtm ✕
jpg ✕

gif ✕
png ✕

* Local Backup Directory ⓘ

/usr/local/aegis/bak

Enable Protection
Cancel

Select the protection mode. You can select the **Whitelist Mode** or **Blacklist Mode**. In whitelist mode, tamper protection is enabled for the specified directories and file formats. In blacklist mode, tamper protection is enabled for the subdirectories, file formats, and files that are not specified. The whitelist mode is selected by default.

- o In whitelist mode, configure the following parameters.

Parameter	Description
Protected Directory	Enter the path of the directory that you want to protect. <div style="background-color: #e0f2f1; padding: 5px; margin-top: 5px;"> ⓘ Note Servers that run Linux and Windows operating systems use different path formats. Enter a valid directory path based on your operating system type. </div>
Protected File Formats	Select file formats from the drop-down list, such as <i>.js</i> , <i>.html</i> , <i>.xml</i> , and <i>.jpg</i> .
Local Backup Directory	The default path where backup files of the protected directories are stored. By default, Security Center assigns <i>/usr/local/aegis/bak</i> to servers that run the Linux operating system and <i>C:\Program Files (x86)\Alibaba\Aegis\bak</i> to servers that run the Windows operating system. You can change the default path as needed.

- o In blacklist mode, configure the following parameters.

Parameter	Description
Protected Directory	Enter the path of the directory that you want to protect.

Parameter	Description
Excluded Sub-Directories	<p>Enter the subdirectories that do not require tamper protection.</p> <p>Click Add Sub-Directory to enter more subdirectories.</p> <p>Security Center does not provide tamper protection for files under the excluded sub-directories.</p>
Excluded File Formats	<p>Select the formats of files that do not require tamper protection.</p> <p>Supported formats include log, txt, and ldb.</p> <p>Security Center does not provide tamper protection for the files in the excluded formats.</p>
Excluded Files	<p>Enter the path of the file that does not require tamper protection.</p> <p>You can click Add File to add more files.</p> <p>Security Center does not provide tamper protection for the excluded files.</p>
Local Backup Directory	<p>The default path where backup files of the protected directories are stored.</p> <p>By default, Security Center assigns <code>/usr/local/aegis/bak</code> to servers that run the Linux operating system and <code>C:\Program Files (x86)\Alibaba\Aegis\bak</code> to servers that run the Windows operating system. You can change the default path as needed.</p>

8. Click **Enable Protection**.

After you enable tamper protection for a server, it is displayed in the server list on the **Tamper Protection** page.

 **Note** By default, tamper protection is in the **Not Initiated** state for newly added servers. To enable tamper protection, you must turn on the **On** switch on the **Tamper Protection** page for the target server.

9. In the server list on the **Tamper Protection** page, find the target server and turn on the switch in the **Protection** column to enable tamper protection for the server.

 **Note** By default, tamper protection is in the **Not Initiated** state for newly added servers. You must enable tamper protection on the **Tamper Protection** page for the target server.

After tamper protection is enabled, the protection state changes to **Running**.

 **Note** If the protection state of a server is **Exception**, click **Exception** in the **Status** column. A message that indicates the causes appears. Click **Retry** in the message.

What to do next

After you enable tamper protection for a server, you can go to the **Overview** page, and select tamper protection in the event type drop-down list to view the alerts on tampering events.

Note

Tamper protection does not take effect immediately after you configure the protected directory, and you can still write files to the directory. In this case, you must go to the **Management** page, disable **Protection** for the server where the directory is located, and then enable **Protection** again.

Handle abnormal protection states

Protection state	Description	Suggestion
Initializing	Web tamper protection is being initialized.	If this is your first time enabling tamper protection for a server, the protection status becomes Initializing . It takes a few seconds to enable tamper protection.
Running	Tamper protection is enabled and running as expected.	-
Exception	An error occurred when tamper protection was enabled.	Click Exception in the Status column to view the exception cause and click Retry .
Not Initiated	Tamper protection is disabled.	You must turn on the On switch to enable tamper protection.

28.6.4.2.3. View the protection status

This topic describes how to view the status of tamper protection for your assets.

Context

The tamper protection feature monitors changes of directories and files in real time and blocks suspicious file changes. To view the status and details of tamper protection in the Apsara Stack Security console, click **Server Security** and choose **Intrusion Prevention > File Tamper Protection**. The details include:

- Overview

On the Overview page, you can view the total number of changed files, protected servers, and protected directories on the current day and during the last 15 days.

- Distribution of protected file types

Protected file types include .txt, .png, .msi, and .zip. You can also add more types of files for protection as needed.

 **Note** All types of files can be added for tamper protection.

- Top five files with the largest number of changes

This module shows the names and paths of the five files with the largest number of changes in the last 15 days.

- Details of tamper protection alerts

The tamper protection feature helps you block all suspicious changes to directories and files on your assets. On the alert details page, you can view the alerts of these changes, including the severity, alert name, affected assets, paths of files with suspicious changes, and protection status.

 **Note**

- If the number of alerts exceeds 100, we recommend that you process these alerts at your earliest opportunity.
- Only the alerts at the **Warning** level are displayed in the console.
- Only alerts in the **Isolation successful** state are displayed. This indicates that the tamper protection feature has blocked the suspicious processes that attempted to make unauthorized file changes.

28.6.4.3. Configure the Virus Removal feature

The Virus Removal feature of Server Guard allows you to customize virus and webshell detection settings.

Virus detection

The Virus Removal feature can automatically quarantine common Internet viruses, such as mainstream trojans, ransomware, mining programs, and DDoS trojans. Apsara Stack Security experts test and verify all automatically quarantined viruses to ensure a minimum false positive rate.

If automatic quarantine is disabled, Server Guard generates alerts when viruses are detected. You must manually manage detected viruses in the console. We recommend that you enable automatic quarantine to better safeguard your servers.

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > Virus Removal**.
3. In the **Anti-virus** section, click **Manage**.
4. In the **Configure Servers for Virus Detection** dialog box, select the servers for which you want to enable the Virus Removal feature.

Select servers from the **Detection Disabled** list on the left side of the tab and click the right arrow to move them to the **Detection Enabled** list on the right side. The anti-virus feature is enabled for the servers in the **Detection Enabled** list. To disable the anti-virus feature for a server, move the server from the **Detection Enabled** list to the **Detection Disabled** list.

5. Click **OK**.
6. In the **Anti-virus** section, turn on **Virus Blocking** to enable virus blocking. After virus blocking is enabled, Server Guard automatically quarantines detected viruses. Quarantined viruses are listed on the **Overview** page. You can select the **Precision Defense** type to filter quarantined viruses.

Webshell detection

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, click **Server Security** and choose **Intrusion Prevention > Virus Removal**.
3. Configure servers for webshell detection.
 - i. In the **Webshell Detection** section, click **Manage**.
 - ii. Select the servers for which you want to enable webshell detection.
 - iii. Click **OK** to complete the configuration.

28.6.5. Log retrieval

28.6.5.1. Log retrieval overview

The log retrieval function provided by Server Security allows you to manage logs scattered in various systems of Apsara Stack in a centralized manner, so that you can easily identify the causes of issues that occur on your servers.

The log retrieval function supports storage of logs for 180 days and query of logs generated within 30 days.

Benefits

The log retrieval function provides the following benefits:

- **End-to-end log retrieval platform:** Allows you to retrieve logs of various Apsara Stack services in a centralized manner and trace issues easily.
- **Cloud-based SaaS service:** Allows you to query logs on all servers in Apsara Stack without additional installment and deployment.
- Supports TB-level data retrieval. It also allows you to add a maximum of 50 inference rules (Boolean expressions) in a search condition and obtain full-text search results within several seconds.
- Supports a wide range of log sources.
- Supports log shipping, which allows you to import security logs to Log Service for further analysis.

Scenarios

You can use log retrieval to meet the following requirements:

- **Security event analysis:** When a security event is detected on a server, you can retrieve the logs to identify the cause and assess the damage and affected assets.
- **Operation audit:** You can audit the operation logs on a server to identify high-risk operations and serious issues in a meticulous way.

Supported log types

Log types

Log type	Description
Logon history	Log entries about successful system logons
Brute-force attack	Log entries about system logon failures that are generated during brute-force attacks
Process snapshot	Log entries about processes on a server at a specific time
Listening port snapshot	Log entries about listening ports on a server at a specific time
Account snapshot	Log entries about account logon information on a server at a specific time
Process initiation	Log entries about process initiation on a server
Network connection	Log entries about active connections from a server to external networks

28.6.5.2. Log retrieval

This topic describes how to search for and view server logs.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Log Retrieval.**
3. Configure search conditions.

Search condition	Description
Log source	Select a supported log source. For more information, see Log sources.
Field	Select a field that is supported by the specified log source. For more information, see Log sources.
Keyword	Enter the keyword of the field that you want to search for.
Logical operator	Select a logical operator from AND, OR, and NOT. For more information, see Logical operators.
+	Add inference rules in a search condition for a log source.
Add conditions	Add search conditions for different log sources.

4. Click **Search** and view the search result.
 - **Reset:** Click **Reset** to clear the search condition configuration.

- **Save Search:** Click **Save Search** to save the search condition configuration for future use.
- **Saved Searches:** Click **Saved Searches** to select and apply a search condition configuration that has been saved.

28.6.5.3. Supported log sources and fields

This topic describes log source types and fields that are supported by the log retrieval feature.

Log retrieval allows you to query the following types of logs. You can click a log source to view the fields that can be retrieved.

Log sources

Log source	Description
Account log	Log entries of successful system logons.
Brute Force log	Log entries of system logon failures generated during brute-force attacks.
Process Snapshot log	Log entries about processes on a server at a specific time.
Network Snapshot log	Log entries of listening ports on a server at a specific time.
Account Snapshot log	Log entries of account logon information on a server at a specific time.
Process log	Log entries of process startup on a server.
Network log	Log entries of active connections from a server to external networks.

Account log

The following table lists fields that are supported in queries.

Supported fields of Account logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
warn_ip	String	The source IP address.
warn_port	String	The logon port.
warn_user	String	The username for the logon.
warn_type	String	The logon type.
warn_count	String	The number of logon attempts.

Brute Force log

The following table lists fields that are supported in queries.

Supported fields of the Brute Force logs

Field	Date type	Description
uuid	String	The agent ID.

Field	Date type	Description
ip	String	The server IP address.
warn_ip	String	The attacker IP address.
warn_port	String	The target port number.
warn_user	String	The target username.
warn_type	String	The type.
warn_count	String	The number of brute-force attack attempts.

Process log

The following table lists fields that are supported in queries.

Supported fields of Process logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
pid	String	The process ID.
groupname	String	The name of the user group.
ppid	String	The ID of the parent process.
uid	String	The ID of the user.
username	String	The user name.
filename	String	The file name.
pfilename	String	The file name of the parent process.
cmdline	String	The command line.
filepath	String	The process path.
pfilepath	String	The parent process path.

Network Snapshot log

The following table lists fields that are supported in queries.

Supported fields of Network Snapshot logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
src_port	String	The listening port.
src_ip	String	The listening IP address.

Field	Date type	Description
proc_path	String	The process path.
pid	String	The process ID.
proc_name	String	The process name.
proto	String	The protocol.

Account Snapshot log

The following table lists fields that are supported in queries.

Supported fields of Account Snapshot logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
perm	String	Indicates whether the agent has root permissions.
home_dir	String	The home directory.
warn_time	String	The time when the password expiration notification was sent.
groups	String	The group to which the user belongs.
login_ip	String	The IP address of the last logon.
last_chg	String	The last time when the password was changed.
shell	String	The Linux shell command.
domain	String	The Windows domain.
tty	String	The logon terminal.
account_expire	String	The time when the account expired.
passwd_expire	String	The time when the password expired.
last_logon	String	The last logon time.
user	String	The user.
status	String	The user status. Valid values: <ul style="list-style-type: none"> 0: disabled 1: normal

Process Snapshot log

The following table lists fields that are supported in queries.

Supported fields of Process Snapshot logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
path	String	The process path.
start_time	String	The time when the process was started.
uid	String	The ID of the user.
cmdline	String	The command line.
pname	String	The name of the parent process.
name	String	The process name.
pid	String	The process ID.
user	String	The user name.
md5	String	The MD5 value of the process file. This value is not calculated if the file size exceeds 1 MB.

Network log

The following table lists fields that are supported in queries.

Supported fields in Network logs

Field	Date type	Description
uuid	String	The agent ID.
ip	String	The server IP address.
src_ip	String	The source IP address.
src_port	String	The source port.
proc_path	String	The process path.
dst_port	String	The destination port.
proc_name	String	The process name.
dst_ip	String	The destination IP address.
status	String	The status.

28.6.5.4. Logical operators

The log retrieval feature supports multiple search conditions. You can add multiple logical operators in one search condition for one log source, or combine multiple search conditions for several log sources by using different logical operators. This topic describes the logical operators that are supported in log retrieval. Examples are provided to help you understand these operators.

The following table describes the logical operators that are supported in log queries.

Logical operator	Description
and	<p>Binary operator.</p> <p>This operator is in the format of <code>query1 and query2</code> , which indicates the intersection of the query results of <code>query1</code> and <code>query2</code> .</p> <p>Note If no logical operators are used for multiple keywords, the default operator is "AND".</p>
or	<p>Binary operator.</p> <p>This operator is in the format of <code>query1 or query2</code> , which indicates the union of the query results of <code>query1</code> and <code>query2</code> .</p>
not	<p>Binary operator.</p> <p>This operator is in the format of <code>query1 not query2</code> , which indicates results that match <code>query1</code> but does not match <code>query2</code> . This format is equivalent to <code>query1 - query2</code> .</p> <p>Note <code>not query1</code> indicates that the log data that does not contain the query results of <code>query1</code> is returned.</p>

28.6.6. Settings

28.6.6.1. Install the Server Guard agent

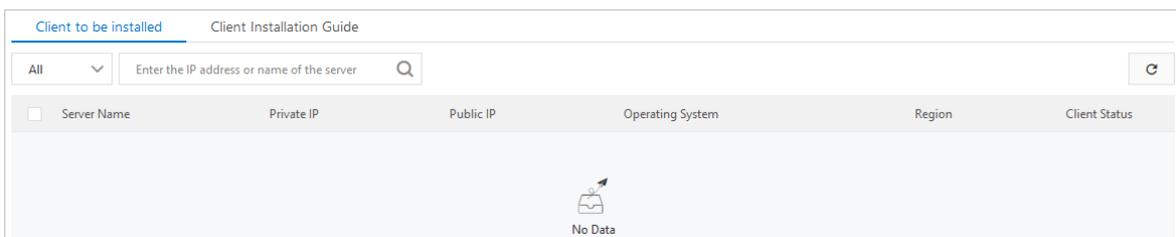
This topic describes how to install the Server Guard agent by specifying parameters.

Context

To use the protection services provided by Server Guard, you must install the Server Guard agent on the operating system of your server.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Server Security > Server Settings > Client Installation**.
3. (Optional) On the Client Installation page, click the **Client to be installed** tab to view the number of servers on which the Server Guard agent is not installed. On this tab, you can also view the relevant information in the server list. You can specify the operating system type, IP address, or server name to search for a target server.



4. Click the **Client Installation Guide** tab.
5. Obtain and install the Server Guard agent based on the operating system type of your server.

- **Windows**
 - a. In the left-side pane of the page, click **Click to download** to download the client software package to your computer.
 - b. Upload the installation package to your server. For example, you can use an FTP client to upload the package to your server.
 - c. Run the installation package on your server as an administrator.

 **Note** When you install the agent on a server that is not in Alibaba Cloud, you will be prompted to enter the installation verification key. You can find the installation verification key on the Server Guard agent installation page.

- **Linux**
 - a. In the right-side pane of the page, select **Alibaba Cloud Server** or **Non-Alibaba Server**.
 - b. Select the installation command for your 32-bit or 64-bit operating system and click **Copy** to copy the command.
 - c. Log on to your Linux server as an administrator.
 - d. Run the installation command on your Linux server to download and install the Server Guard agent.

28.6.6.2. Manage protection modes

This topic describes how to manage protection modes for each server to make them more efficient and secure.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Server Security** > **Server Settings** > **Protection Mode**.
3. On the Protection Mode page, click **Manage**. Configure protection modes for each server.
 - **Business First Mode**: The peak CPU utilization is less than 10%, and the peak memory usage is less than 50 MB.
 - **Protection First Mode**: The peak CPU utilization is less than 20%, and the peak memory usage is less than 80 MB.
4. Click **OK**.

28.7. Application security

28.7.1. Quick start

This topic helps you get started with Web Application Firewall (WAF).

WAF uses intelligent semantic analysis algorithms to identify web attacks. WAF also integrates a learning model to enhance its analysis capabilities and meet your daily security protection requirements without relying on traditional rule libraries.

The following content describes the procedure for using WAF:

1. **Customize WAF protection rules.**

WAF provides a default protection policy. You can also customize policies that suit your business requirements.

 - For more information about how to configure protection policies, see [Configure protection policies](#).
 - For more information about how to configure custom rules, see [Create a custom rule](#).
 - For more information about how to configure HTTP flood protection rules, see [Configure an HTTP flood protection rule](#).
2. **Add protected websites.**

WAF can protect Internet websites and virtual private cloud (VPC) websites.

- For more information about how to add an Internet website to WAF for protection, see [Add an Internet website for protection](#).
 - For more information about how to add a VPC website to WAF for protection, see [Add a VPC website for protection](#).
3. Configure Domain Name System (DNS) resolution.
- For more information about how to change the DNS-resolved source IP address of a website to a virtual IP address of WAF, see [Modify DNS resolution settings](#).
4. View WAF protection results.
- For more information about how to view the protection overview, see [View protection overview](#).
 - For more information about how to view the service access information, see [View Web service access information](#).
 - For more information about how to view the detection logs for web attacks, see [View attack detection logs](#).
 - For more information about how to view the detection logs for HTTP flood attacks, see [View HTTP flood protection logs](#).

28.7.2. Detection overview

28.7.2.1. View protection overview

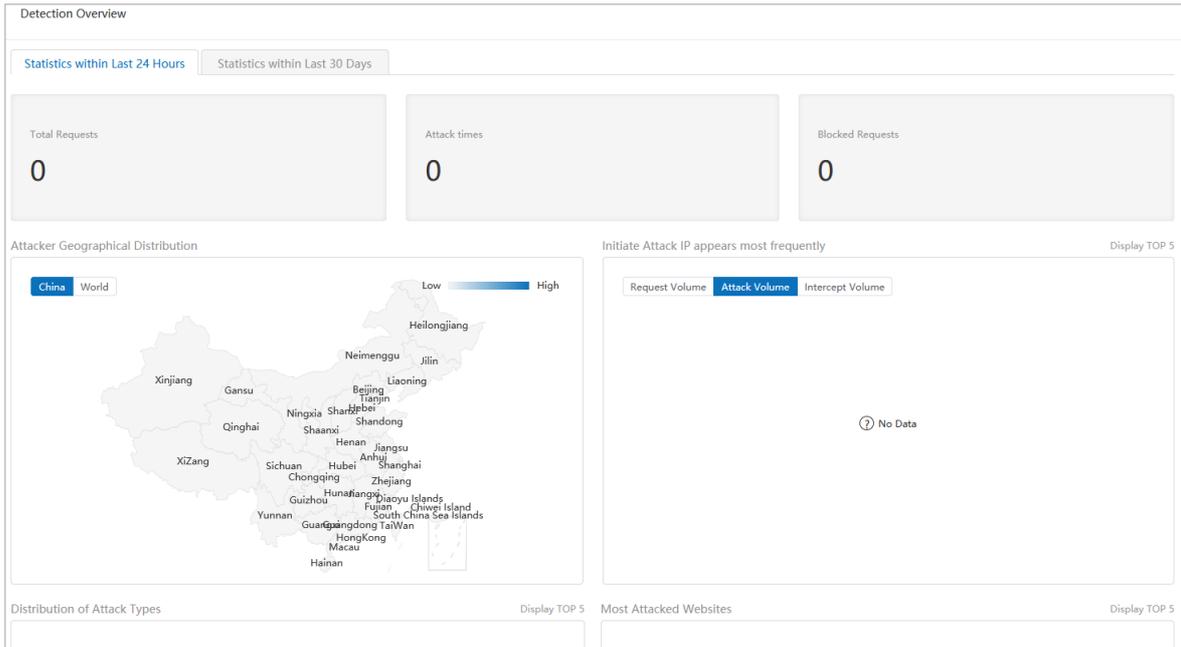
This topic describes how to view the Web Application Firewall (WAF) detection overview.

Context

The Protection Overview page displays information such as the statistics of previous attacks, the geographical distribution of attackers, and the numbers of total requests and blocked requests. You can learn the web attack protection information and custom protection rules.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation tree, choose **Detection Overview > Detection Overview**.
4. On the **Detection Overview** page, you can view **Statistics within Last 24 Hours** and **Statistics within Last 30 Days**.



- **Requests**
Displays the total number of requests.
- **Attacks**
Displays the total number of attacks.
- **Blocked requests**
Displays the total number of blocked requests.
- **Attacker geographical distribution on a map**
Displays the distribution of attackers on a map. You can select a map of China or a map of the world.
- **Distribution of total requests and blocked requests.**
- **Distribution of top five attack IP addresses**
Displays the top five IP addresses that have launched the most attacks in a bar chart. The X-axis indicates the number of requests. The Y-axis indicates the IP address.
- **Distribution of top five attack types**
Displays the distribution of the top five attack types and the number of attacks of each type in a pie chart.
- **Top five attacked websites**
Displays the top five attacked websites and the number of attacks on each website in a bar chart .

28.7.2.2. View web service access information

This topic describes how to view the service access information.

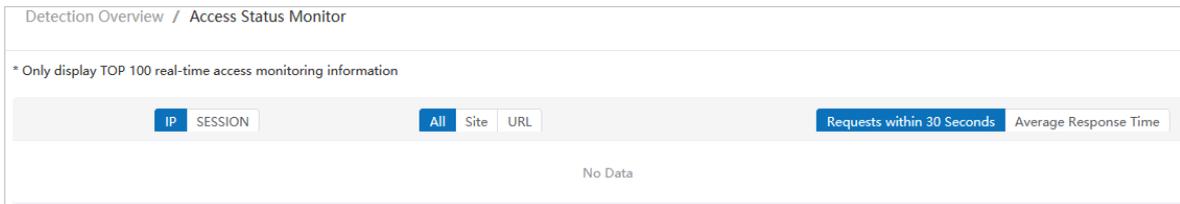
Context

Web Application Firewall (WAF) monitors the web service access information. This allows security administrators to analyze the business access information and detect vulnerabilities.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Application Security > WAF**.

- In the left-side navigation tree, choose **Detection Overview > Access Status Monitor**.
- Filter the access records to view the details.



28.7.3. Protection logs

28.7.3.1. View attack detection logs

This topic describes how to view attack detection logs.

Context

These logs allow you to analyze the attacks on your web services. Based on the analysis, you can update the attack protection policies and custom rules and fix the web service vulnerabilities.

Procedure

- Log on to [Apsara Stack Security Center](#).
- In the left-side navigation pane, choose **Application Security > WAF**.
- In the left-side navigation tree, choose **Detection Logs > Attack Detection Logs**.
- Click **Filter**, specify filter conditions, and click **OK**.

 **Note** If you specify multiple conditions, all of the conditions must be met.

- View the detected attacks.

28.7.3.2. View HTTP flood protection logs

This topic describes how to view HTTP flood protection logs.

Context

These logs allow you to analyze HTTP flood attacks on your web services. Based on the analysis, you can update the HTTP flood protection rules and HTTP flood whitelist and fix the web service vulnerabilities.

Procedure

- Log on to [Apsara Stack Security Center](#).
- In the left-side navigation pane, choose **Application Security > WAF**.
- In the left-side navigation tree, choose **Detection Logs > HTTP Flood Detection Logs**.
- Click **Filter**, specify filter conditions, and then click **OK**.

 **Note** If you specify multiple conditions, all of the conditions must be met.

- View the HTTP flood detection result. The blocked HTTP flood attacks, related rules, and attack time are displayed.

28.7.3.3. View system operation logs

This topic describes how to view system operations logs.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF.**
3. In the left-side navigation tree, choose **Detection Logs > System operation log.**
4. View the system operations logs. The usernames, content, IP addresses, and creation time are displayed.

28.7.3.4. View access logs

This topic describes how to view access logs.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF.**
3. In the left-side navigation tree, choose **Detection Logs > Access Log.**
4. Click **Filter**, specify filter conditions, and then click **OK.**

 **Note** If you specify multiple conditions, all of the conditions must be met.

5. View the access logs. The requested IP addresses, destination IP addresses, source IP addresses, methods, and response codes are displayed.

28.7.4. Protection configuration

28.7.4.1. Configure protection policies

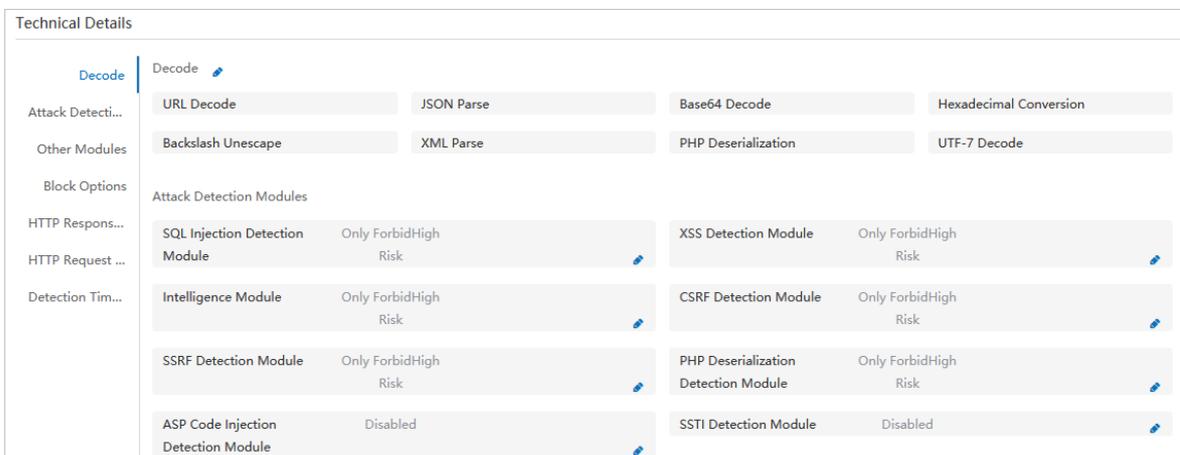
This topic describes how to configure Web Application Firewall (WAF) protection policies.

Context

WAF provides a default protection policy. You can also customize policies that suit your business requirements.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF.** On the **Detection Overview** page, choose **Protection Configuration > Website Protection Policies.**
3. Click **Add protection policy.** In the dialog box that appears, specify **Policy name** and click **Confirm.**
4. In the **Operation** column of the new protection policy, click the  icon to view details.



Parameter	Description
Decode	Select algorithms for decoding the requests.
Attack Detection Modules	Specify the types of attacks to be detected and the risk levels of attacks to be blocked.
Block Options	Specify the status code and image to be returned when an attack is blocked.
HTTP Response Processing	Set the status of the Enable HTTP response processing switch and specify Response Detection Max Body Size.
HTTP Request Body Detection	Specify Response Detection Max Body Size.
Detection Timeout	Set the status of the Enable Detection Timeout switch, and specify Timeout Threshold.

For example, perform the following steps to configure **Attack Detection Modules**:

- i. Move the pointer over a specific module in the **Attack Detection Modules** section, for example, **SQL Injection Detection Module**, and click the modify icon.
- ii. In the **SQL Injection Detection Module** dialog box, specify the detection parameters.

Parameter	Description
Enabled	Indicates whether to enable the detection module.
Blocking Threshold	You can select NotForbid , Only ForbidHigh Risk , ForbidMedium or High Risk , or Forbid All .
Record Threshold	You can select Notrecord , Onlyrecord High Risk , recordMedium or High Risk , or record All .
Detect Non-Injected SQL	Indicates whether to enable detection for non-injected SQL attacks.

iii. Click **OK**.

5. Manage the protection policies.

To delete a protection policy, click the protection policy first. Then, in the upper-right corner, choose **More > Delete Selected Protection Policies**. In the dialog box that appears, click **OK**.

 **Note** You cannot delete the default policy.

28.7.4.2. Create a custom rule

This topic describes how to create a custom rule for Web Application Firewall (WAF).

Context

Security administrators can customize rules to meet various requirements for intrusion detection. You can add, edit, or delete custom rules in the WAF console as an administrator or by using an administrator account. You can use custom rules to filter out requests that meet specific conditions.

Multiple custom rules have the **OR** logical relation. If two custom rules specify the same conditions but differentiate in the operating mode such as blocking traffic and allowing traffic, the system runs the first rule.

Procedure

1. [Log on to Apsara Stack Security Center](#).

- In the left-side navigation pane, choose **Application Security > WAF**. On the **Detection Overview** page, choose **Protection Configuration > Customized Rules**.
- In the upper-right corner, click **Add Rule**. In the **Add Customized Rule** dialog box, configure parameters.

Parameters for creating a custom rule

Parameter	Description
Type	The operating mode of the rule. Valid values: Block , Allow , Monitor , and Detection module control . <ul style="list-style-type: none"> ◦ Block: An HTTP request is blocked if it meets the conditions of the rule. ◦ Allow: An HTTP request is allowed if it meets the conditions of the rule. ◦ Monitor: An HTTP request is allowed and recorded if it meets the conditions of the rule. ◦ Detection module control.
Comment	The remarks of the rule, such as the purpose of the rule.
Risk Level	The risk level. Valid values: No threat , Low Risk , Medium Risk , and High Risk .
Matching Patter	The conditions that trigger the rule. Click Add Pattern to specify multiple conditions. Multiple conditions have the AND logical relation. The custom rule takes effect only when all conditions are met.
Apply to Websites	The websites to be protected by the rule.
Log Recording Option	Specifies whether to record a protection event in the intrusion detection logs if the rule is triggered. The default value is Enable Log Recording . After Log Recording Option is set to Enable Log Recording , all interception records are recorded in the intrusion detection logs.
Attack type	The type of attack to be blocked by the rule.
Expiration Time	The time when the rule expires.

- Click **Confirm**.

5. Manage a custom rule.

- Edit a rule.

To edit a rule, click the  icon in the Actions column.

- Enable a rule.

To enable a rule, select this rule and click **Enable Selected Rules**.

- Disable a rule.

To disable a rule, select this rule and click **Disable Selected Rules**.

- Delete a rule.

To delete a rule, select this rule and click **Delete Selected Rules**.

28.7.4.3. Configure an HTTP flood detection rule

This topic describes how to configure an HTTP flood protection rule.

Context

An HTTP flood is a type of distributed denial of service (DDoS) attack that targets at web applications. Attackers use proxy servers or zombies to overwhelm targeted web servers by sending large quantities of HTTP requests.

Create an HTTP flood protection rule

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Application Security > WAF**. On the Detection Overview page, choose **Protection Configuration > HTTP Flood Detection**.
3. Click **Add Rule**. The **Add HTTP Flood Detection Rules** dialog box appears.
4. Configure parameters and click **Confirm**.

Add HTTP Flood Detection Rules
×

Rule Mode Observe Blocking Mode

Rule Types Restrict Users by Policy Restrict Known Users

Rule Name *

Target Type IP SESSION

Restricted IP List * Fill IP

One IP address or IP address segment per line
If it is an IP address segment, please use "IP address/subnet mask" format such as
192.168.100.200

Cancel
Confirm

Parameter	Description
Rule Mode	<p>The operating mode of the HTTP flood protection rule. Valid values: Blocking Mode and Observe.</p> <ul style="list-style-type: none"> ◦ Blocking Mode: blocks the requests that meet the access control rules. ◦ Observe: records the requests that meet the access control rules, but does not block the requests.
Rule Types	<p>The type of the HTTP flood detection rule. Valid values: Restrict Users by Policy and Restrict Known Users. The difference between the two types is that whether the user is a known IP address or session.</p> <ul style="list-style-type: none"> ◦ Restrict Users by Policy: Restrict users who meet all configuration items of the restriction rule. Configuration items include Restriction Trigger Threshold, Restricted URL Address, Restriction Mode, Restriction Time, and Statistical Range of Visits in the Advanced section. ◦ Restrict Known Users: Restrict known users based on the restriction rule. To achieve this purpose, you must configure the IP address or session list and the restriction mode. After the configuration is complete, the restriction rule takes effect.
Rule Name	The name of the HTTP flood detection rule.
Target Type	<p>The type of the restricted target. Valid values: IP and Session.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note If Target Type is set to SESSION, you must set User Identification to WAF User System when you add a website bound to this HTTP flood protection rule to WAF. For more information, see Add an Internet website for protection.</p> </div>
Restriction Trigger Threshold	If Rule Types is set to Restrict Users by Policy , you must complete the configuration items that trigger the restriction rule.
Restricted URL Address	<p>If Rule Types is set to Restrict Users by Policy, you must specify the target URL address to be protected by the restriction rule.</p> <ul style="list-style-type: none"> ◦ URL Prefix ◦ URL ◦ Record all IP addresses
Restricted IP List or Restricted Session List	If Rule Types is set to Restrict Known Users , enter the IP addresses or sessions to be restricted based on the setting of Rule Types. Specify only one IP address or session in each line.
Restricted URL Address	<p>If Rule Types is set to Restrict Known Users, you must specify the target URL address to be protected by the restriction rule.</p> <ul style="list-style-type: none"> ◦ URL Prefix ◦ URL ◦ Restrict user access to all addresses

Parameter	Description
Restriction Mode	<p>The mode in which the requests from a user are restricted. Valid values:</p> <ul style="list-style-type: none"> ○ Forbidden: All requests from the restricted user to the specified URL are blocked. ○ Frequency control: The frequency at which requests from the restricted user are allowed to the specified URL is limited.
Restriction Time	The time when the restriction rule takes effect.
Statistical Range of Visits	<p>If Target Type is set to Restrict Users by Policy, you can specify the range of targets that you want to analyze in the Advanced section.</p> <ul style="list-style-type: none"> ○ Statistics Full Access Data: After you select Statistics Full Access Data, the request frequency is limited for all users who have passed WAF and meet the frequency restrictions. However, this decreases system performance. ○ Statistics TOP Access Data: After you select Statistics TOP Access Data, the request frequency is limited only for the top 100 data records in real-time access monitoring. When the size of the full access data is greater than that of the top 100 data records, the system performance decrease is insignificant.

Manage an HTTP flood protection rule

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF**. On the **Detection Overview** page, choose **Protection Configuration > HTTP Flood Detection**.
3. In the rule list, manage the added HTTP flood detection rules.
 - Search for a rule.
Click **Filter** and add filter conditions to find a rule.
 - Enable a rule.
Select a rule that is in the **Disabled** state and choose **More > Enable Selected Rules**.
 - Disable a rule.
Select a rule that is in the **Enabled** state and choose **More > Disable Selected Rules**.
 - Delete a rule.
Select a rule and choose **More > Delete Selected Rules**.

28.7.4.4. Configure an HTTP flood protection whitelist

This topic describes how to configure an HTTP flood protection whitelist.

Context

If a request source is trusted, you can add this request source to an HTTP flood protection whitelist to allow the requests from this source.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF**. On the **Detection Overview** page, choose **Protection Configuration > HTTP Flood Detection**. On the page that appears, click the **HTTP Flood Detection Whitelist** tab.

3. Click the HTTP Flood Detection Whitelist tab. On the HTTP Flood Detection Whitelist tab, click **Add Whitelist Item** to add a request source to the whitelist, and click **Confirm**.

Parameter	Description
Type	Set the type of the whitelisted request source to IP or Session.
IP or SESSION	Specify the IP addresses or sessions based on the selected Type. Specify one IP address or session in each line.
Comment	Enter comments for the whitelist.

4. Manage the users in the whitelist.
 - Search for a user in the whitelist.

Click **Filter**. In the dialog box that appears, click **Add Filter Item** to find a whitelisted user.
 - Remove a whitelisted user.

Select a whitelisted user and choose **More > Delete Selected Items**.

28.7.4.5. Manage SSL certificates

This topic describes how to upload or delete SSL certificate files.

Context

After you upload an SSL certificate file on the **SSL Certificate Management** page, you can select this certificate file when you add an HTTPS website for protection.

Note When you add an HTTPS website for protection on the **Protected Websites** page, you must select the SSL certificate file that corresponds to the domain.

Procedure

1. Log on to **Apsara Stack Security Center**.
2. In the left-side navigation pane, choose **Application Security > WAF**.

3. On the page that appears, choose **Protection Configuration > SSL Certificate Management**.
4. Upload a new certificate file.
 - i. Click **Upload SSL Certificate**.
 - ii. In the **Add File** dialog box, specify **Name**. We recommend that you enter the domain name for easier management.

 **Note** If your certificate and private key are in the same file, select **Include private key in certificate file**.

- iii. In the **File** section, upload the HTTPS certificate file and private key.
 - iv. Specify **Certificate Password**.
 - v. Click **Confirm**.
5. (Optional) Delete an uploaded SSL certificate file. You can delete expired SSL certificate files.
 - i. In the SSL certificate list, select the certificate file that you want to delete.
 - ii. Choose **More > Delete selected SSL Certificates**.
 - iii. In the dialog box that appears, click **Confirm**.

28.7.4.6. Add Internet websites for protection

This topic describes how to add Internet websites to Web Application Firewall (WAF) for protection.

Context

WAF can protect the following types of websites:

- Internet websites.
- Virtual Private Cloud (VPC) websites. For more information about how to add a VPC website for protection, see [Add a VPC website for protection](#).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Application Security > WAF**.
3. In the left-side navigation pane of the **Detection Overview** page, choose **Protection Configuration > Protection site management**. Then click the **Internet Websites** tab.
4. Click **Add a site** in the upper-right corner.
5. In the **Add Protected Site** pane, configure parameters in the **Monitoring Information** step and click **Next**.
Specify the Internet website to be protected. WAF can protect both HTTP and HTTPS websites.

Add Protected Site
✕

1

Monitoring Information

Configure Protected Site Information on WAF

Protected Website Name *

Domain Name *

Remarks

Remarks

Port Settings * Enable SSL

Create Virtual IP Method

Parameter	Description
Protected Website Name	The name of the website to be protected.
Domain Name	The domain name of the website. <ul style="list-style-type: none"> ◦ You can use an asterisk (*) as a wildcard domain name. ◦ Separate multiple domain names with commas (,).
Port Settings	The port that WAF listens on. <ul style="list-style-type: none"> ◦ If the website supports HTTPS requests, select Enable SSL and upload an HTTPS certificate. ◦ If the website can be accessed over multiple ports, click Add a group of ports to add required ports.
Cert Setting	The HTTPS certificate of the website. Valid values: <i>Upload a New Certificate</i> and <i>Choose an Existing Certificate</i> . <ul style="list-style-type: none"> ◦ <i>Upload a New Certificate</i>: If the HTTPS certificate used by the website has not been uploaded to WAF, select this option. By default, the HTTPS certificate and private key are separately uploaded. If you select Include private key in certificate file, you need to upload only a file that contains both the HTTPS certificate and private key. ◦ <i>Choose an Existing Certificate</i>: If the HTTPS certificate used by the website has been uploaded to WAF, select this option, and then select the required HTTPS certificate from the drop-down list. <div style="background-color: #e0f2f7; padding: 5px; margin-top: 10px;"> ? Note Specify this parameter only if you select Enable SSL next to Port Settings. </div>

Parameter	Description
Name	<p>The name of the HTTPS certificate.</p> <p>Note Specify this parameter only if you select Enable SSL next to Port Settings and set Cert Setting to Upload a New Certificate.</p>
File	<p>Upload the HTTPS certificate and private key.</p> <p>By default, the HTTPS certificate and private key are separately uploaded. If you select Include private key in certificate file next to Name, you need to upload only a file that contains both the HTTPS certificate and private key.</p> <p>Note Specify this parameter only if you select Enable SSL next to Port Settings.</p>
Virtual IP	<p>Select an IP address type and a virtual IP address.</p> <p>Note You can specify an IPv6 address as the virtual IP address of WAF.</p> <p>By default, WAF provides 10 virtual IP addresses. You can add virtual IP addresses as required.</p> <p>Note A virtual IP address is available only for the department to which the user that has created this virtual IP address belongs.</p>

6. In the Request Processing Method step, configure parameters and click Next.

Add Protected Site
✕

Monitoring Information
Configure Protected Site Information on WAF

2 Request Processing Method
Configure WAF server response method

Request Processing Method Forward to Backend Server Redirect
 Respond with Specified Content

Load Balancing Algorithm Weighted Round Robin Least Connections Method
 Source Address Hash

Backend Server Address *

Fill in the back-to-source address Return to the back-to-source instance

http:// : 80 Weight

Response mode	Configuration item	Description
Forward to Backend Server	Load Balancing Algorithm	The algorithm for load balancing. Valid values: Weighted Round Robin , Source Address Hash , and Least Connections Method .
	Backend Server Address	The IP address of the origin server to which WAF forwards inbound traffic based on traffic filtering. Valid values: Fill in the back-to-source address and Return to the back-to-source instance . <ul style="list-style-type: none"> ◦ Fill in the back-to-source address: Enter the address of the origin server. If you enter multiple addresses, load balancing is performed based on the specified load balancing algorithm. ◦ Return to the back-to-source instance: Enter the address of a specific ECS or SLB instance. If you enter multiple addresses, load balancing is performed based on the specified load balancing algorithm.
	X-Forwarded-For	The passthrough mode of the source IP address. The X-Forwarded-For (XFF) HTTP header is used to identify the original IP address of an HTTP client. It is used for request forwarding services such as HTTP proxy and load balancing.
Redirect	Response Status Code	WAF forwards inbound traffic to a specified address. You can set this parameter to 301 , 302 , or 307 . <ul style="list-style-type: none"> ◦ 301: The requested page has been permanently moved to another URL. ◦ 302: The requested page has been temporarily moved to another URL. The requester must continue to use the original URL for future requests. ◦ 307: The resource requested has been temporarily moved to the URL given by the Location headers. The method and the body of the original request are reused to perform the redirected request.
	Redirect address	The target URL of redirection.
Respond with Specified Content	Response Status Code	Specify the content of the response. You can select a value from multiple status codes, such as 200 , 404 , and 503 .
	Response	Upload the response content. For example, upload an image. This image is returned if a user visits the website.

7. In the Protection Policy step, configure parameters and click Next. Then go to the Finish step to complete website addition.

 **Note** You can configure a protection policy only if Request Processing Method is set to Forward to Backend Server.

Parameter	Description
Protection Policy	Select a WAF protection policy. For more information, see Configure protection policies .
User Identification	Specify whether to enable the user identification feature. <div style="border: 1px solid #add8e6; padding: 5px;"> <p>? Note If you have enabled HTTP flood protection for the protected website and have set Target Type to Session when you configure request limits, you must set User Identification to WAF User System.</p> </div>

28.7.4.7. Add VPC websites for protection

This topic describes how to add VPC websites to WAF for protection.

Context

WAF can protect the following types of websites:

- Internet websites. For more information about how to add an Internet website for protection, see [Add an Internet website for protection](#).
- VPC websites.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Application Security > WAF**. In the left-side navigation pane of the Detection Overview page, choose **Protection Configuration > Protection site management**. Then click the **VPC Websites** tab.
3. Click **Add a site** in the upper-right corner. The **Add Protected Site** pane appears.
4. In the **Monitoring Information** step, configure parameters and click **Next**.

Specify the VPC website to be protected. WAF can protect both HTTP and HTTPS websites.

Add Protected Site
✕

1 Monitoring Information

Configure Protected Site Information on WAF

Protected Website Name *

Domain Name *

Remarks

Port Settings * Enable SSL

2 set up VPC

Parameter	Description
Protected Website Name	The name of the website to be protected.
Domain Name	<p>The domain name of the website.</p> <ul style="list-style-type: none"> You can use an asterisk (*) as a wildcard domain name. Separate multiple domain names with commas (,).
Port Settings	<p>The port that WAF listens on.</p> <ul style="list-style-type: none"> If the website supports HTTPS requests, select Enable SSL and upload an HTTPS certificate. If the website can be accessed over multiple ports, click Add a group of ports to add required ports.
Cert Setting	<p>The HTTPS certificate of the website. Valid values: Upload a New Certificate and Choose an Existing Certificate.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-bottom: 10px;"> <p> Note Specify this parameter only if you select Enable SSL next to Port Settings.</p> </div> <ul style="list-style-type: none"> Upload a New Certificate: If the HTTPS certificate used by the website has not been uploaded to WAF, select this option. By default, the HTTPS certificate and private key are separately uploaded. If you select Include private key in certificate file, you need to upload only a file that contains both the HTTPS certificate and private key. Choose an Existing Certificate: If the HTTPS certificate used by the website has been uploaded to WAF, select this option, and then select the required HTTPS certificate from the drop-down list.
Name	<p>The name of the HTTPS certificate.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Specify this parameter only if you select Enable SSL next to Port Settings and set Cert Setting to Upload a New Certificate.</p> </div>
File	<p>Upload the HTTPS certificate and private key. By default, the HTTPS certificate and private key are separately uploaded. If you select Include private key in certificate file next to Name, you need to upload only a file that contains both the HTTPS certificate and private key.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Specify this parameter only if you select Enable SSL next to Port Settings.</p> </div>

5. In the set up VPC step, configure parameters and click **Next**.

Add Protected Site
✕

1 Monitoring Information
Configure Protected Site Information on WAF

2 set up VPC
Configure the VPC and related parameters of the protection site

Protected VPC *

Virtual Switch *

Create Virtual IP Method

VPC Virtual IP *

3 Request Processing Method
Configure WAF server response method

Parameter	Description
Protected VPC	The VPC to which the website belongs.
Virtual Switch	The VSwitch to which the website belongs.
Create Virtual IP Method	The method to create a virtual IP address. Valid values: Select an existing virtual IP and Create virtual IP.
VPC Virtual IP	<ul style="list-style-type: none"> ◦ If you set Create Virtual IP Method to Select an existing virtual IP, select an existing virtual IP address from the VPC Virtual IP drop-down list. ◦ If you set Create Virtual IP Method to Create virtual IP, click Click to Create Vip next to VPC Virtual IP to generate a virtual IP address.

6. In the Request Processing Method step, configure parameters and click Next.

Response mode	Configuration item	Description
Forward to Backend Server	Load Balancing Algorithm	Select an algorithm for load balancing. Valid values: Weighted Round Robin, Source Address Hash, and Least Connections Method.
	Backend Server Address	The IP address of the origin server to which WAF forwards inbound traffic based on traffic filtering.
	X-Forwarded-For	The passthrough mode of the source IP address. The XFF HTTP header is used to identify the original IP address of an HTTP client. It is used for request forwarding services such as HTTP proxy and load balancing.

Response mode	Configuration item	Description
Redirect	Response Status Code	<p>WAF forwards inbound traffic to a specified address.</p> <p>You can set this parameter to 301, 302, or 307.</p> <ul style="list-style-type: none"> ○ 301: The requested page has been permanently moved to another URL. ○ 302: The requested page has been temporarily moved to another URL. The requester must continue to use the original URL for future requests. ○ 307: The resource requested has been temporarily moved to the URL given by the Location headers. The method and the body of the original request are reused to perform the redirected request.
	Redirect address	The target URL of redirection.
Respond with Specified Content	Response Status Code	<p>Specify the content of the response.</p> <p>You can select a value from multiple status codes, such as 200, 404, and 503.</p>
	Response	<p>Upload the response content.</p> <p>For example, upload an image. This image is returned if a user visits the website.</p>

7. In the Protection Policy step, configure parameters and click Next. Then go to the Finish step to complete website addition.

Parameter	Description
Protection Policy	Select a WAF protection policy. For more information, see Configure protection policies .
User Identification	Specify whether to enable the user identification feature.

28.7.4.8. Verify the access configuration of a domain on your computer

This topic describes how to verify the access configuration of a domain on your computer.

Context

Before you redirect traffic to Web Application Firewall (WAF), we recommend that you perform a verification on your computer. This allows you to ensure that the domain is connected to WAF. This also allows you to ensure that WAF can correctly forward traffic. After you add the virtual IP address of WAF and the domain name of a website to the hosts file on your computer, the request to access the domain from a local browser passes through WAF first.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. Add the virtual IP and domain name to the `hosts` file on your computer. If your computer runs Windows 7, in Windows 7, the hosts file is stored in the following path: `C:\Windows\System32\drivers\etc\hosts`.
 - i. Open the hosts file by using a text editor such as Notepad.

- ii. Add the following content to the end of the file: `<Protected website virtual IP address><Protected website domain>` .

```
# localhost name resolution is handled within DNS itself.
# ->127.0.0.1 ..... localhost
# ->::1 ..... localhost
# ->4.115 ..... example.com
```

Note The IP address preceding the domain name is the virtual IP address assigned by WAF.

- Ping the protected domain name from your computer. The resolved IP address must be the virtual IP address of WAF in the hosts file. If the resolved IP address is still origin IP address, refresh the local Domain Name System (DNS) cache.
- Enter the domain name in the address bar of a browser and press Enter. If the access configuration on WAF is correct, you can visit the requested website.
- Verify the WAF protection feature. Simulate a web attack request to check whether WAF blocks the request.

For example, add `/? alert(xss)` after the URL. If you try to visit `www.example.com /? alert(xss)`, WAF must block the request.

28.7.4.9. Modify DNS resolution settings

This topic describes how to modify the DNS resolution settings to connect your business to WAF.

Context

Before you modify the DNS resolution settings and redirect business traffic to WAF, make sure that you have passed local verification.

The domain name of a protected website may not be resolved by a DNS provider. For example, a website may use an SLB instance to connect to the Internet. To connect such a domain name to WAF, perform the following steps to specify the virtual IP address of the protected website as the origin IP address of the SLB instance:

Procedure

- Log on to [Apsara Stack Security Center](#).
- In the left-side navigation pane, choose **Application Security > WAF**.
- In the left-side navigation pane of the Detection Overview page, choose **Protection Configuration > Protection site management**.
- Find the target website and click the  icon in the Operation column.
- On the **Basic Information** tab, obtain the virtual IP address of the protected website.
- Log on to the console provided by the DNS provider and find the domain name resolution settings for the relevant domain name. Then, change the value of record A to the virtual IP address of the protected site.

Note We recommend that you set the TTL to 600s in DNS resolution settings. The greater the TTL is, the longer it takes to synchronize and update the DNS records.

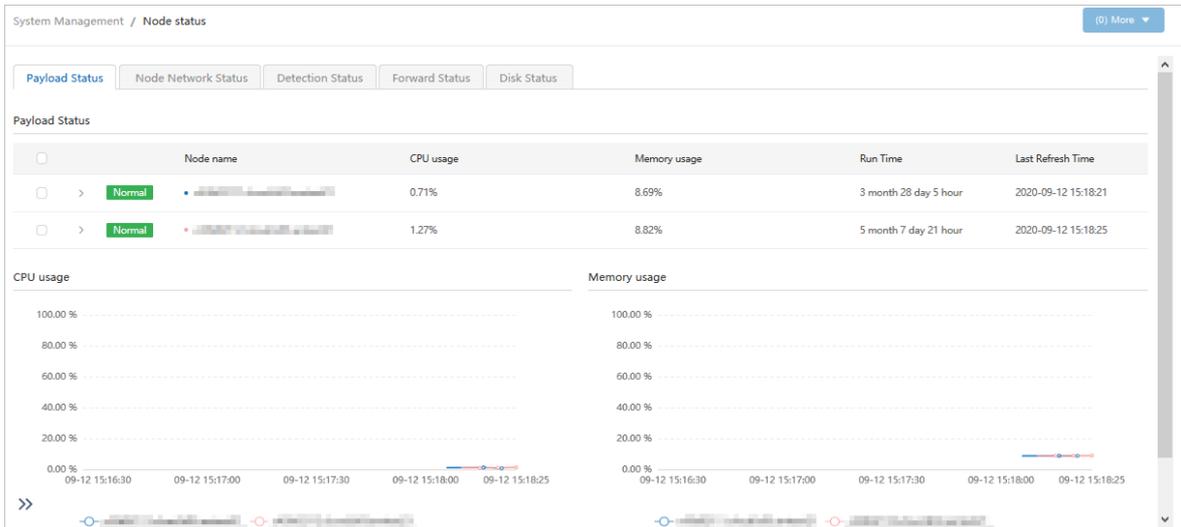
28.7.5. System management

28.7.5.1. View the payload status of nodes

This topic describes how to view the CPU utilization and memory usage of WAF nodes. You can use the query results to identify faults or check whether scaling is required.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Application Security > WAF. In the left-side navigation pane of the Detection Overview page, choose System Management > Node status.
3. On the Payload Status tab, view the payload status of WAF nodes.



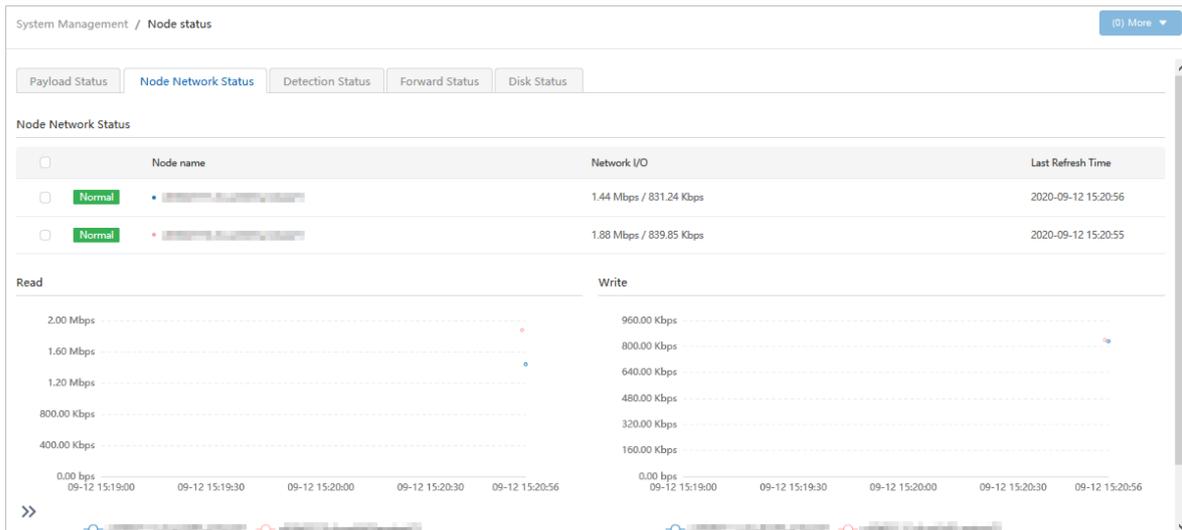
In the Payload Status section, you can view the CPU utilization and memory usage of each node. In the CPU usage and Memory usage sections, you can view changes in the CPU utilization and memory usage over a period of time.

28.7.5.2. View the network status of nodes

This topic describes how to view the network status of WAF nodes, such as the network I/O, traffic detection status, and traffic forwarding status.

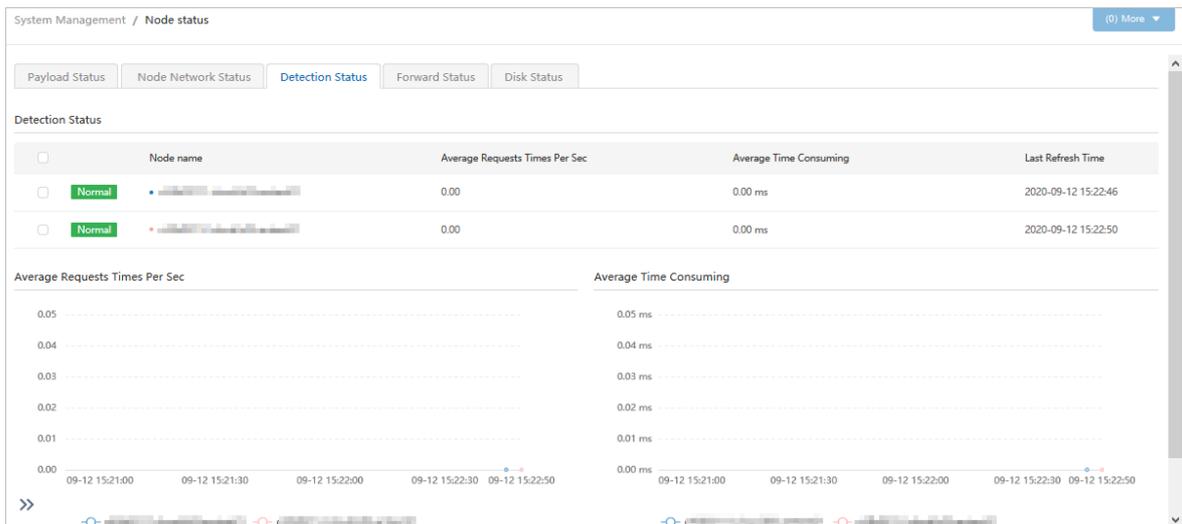
Node network status

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Application Security > WAF. In the left-side navigation pane of the Detection Overview page, choose System Management > Node status.
3. Click the Node Network Status tab.
4. View the network I/O of WAF nodes.



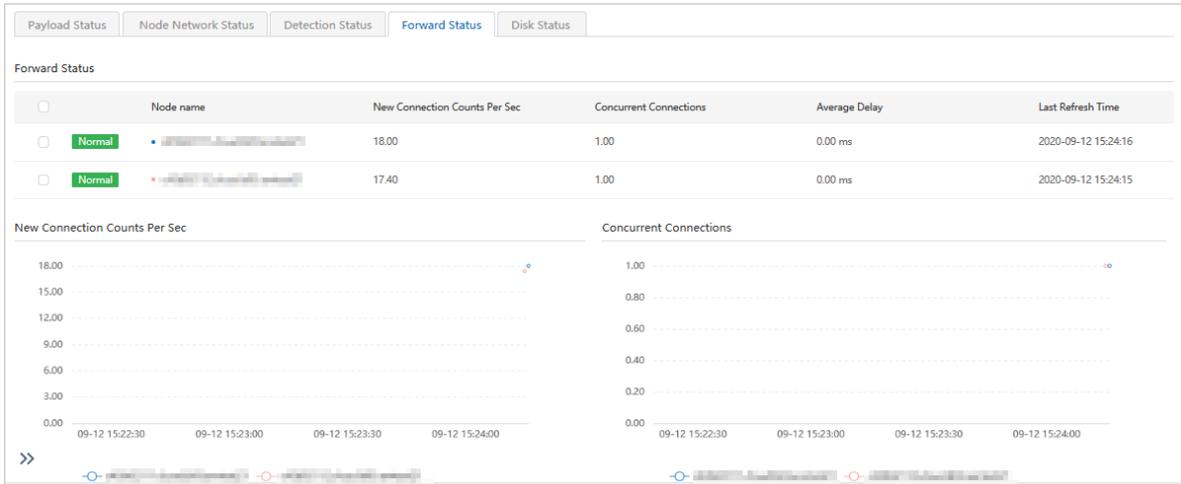
Traffic detection status

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF**. In the left-side navigation pane of the Detection Overview page, choose **System Management > Node status**.
3. Click the **Detection Status** tab.
4. View the network traffic detection status of WAF nodes.



Traffic forwarding status

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Application Security > WAF**. In the left-side navigation pane of the Detection Overview page, choose **System Management > Node status**.
3. Click the **Forward Status** tab.
4. View the network traffic forwarding status of WAF nodes.

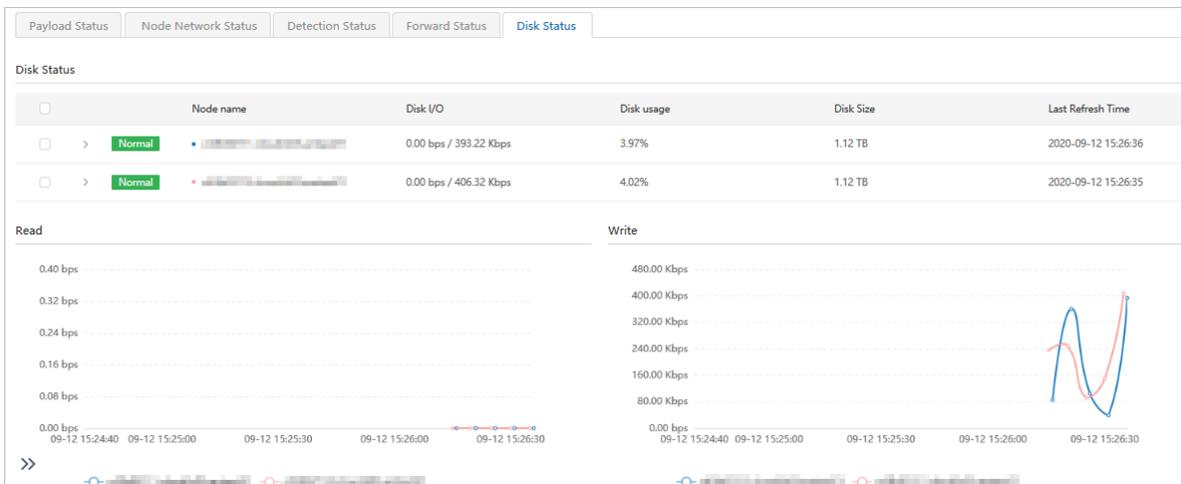


28.7.5.3. View the disk status of nodes

This topic describes how to view the disk status of WAF nodes. You can use the query results to identify faults or check whether scaling is required.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Application Security > WAF. In the left-side navigation pane of the Detection Overview page, choose System Management > Node status.
3. Click the Disk Status tab to view the disk status of WAF nodes.



In the Disk Status section, you can view the disk I/O and disk usage of nodes. In the Read and Write sections, you can view the disk read and write changes over a period of time.

28.7.5.4. Configure alert service

This topic describes how to add a syslog server to WAF. After the syslog server is added, WAF alert logs can be pushed to the syslog server over the syslog protocol.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Application Security > WAF. On the Detection Overview page, choose System Management > Syslog Configuration.
3. Click the Alarm Service Configuration tab and click Add alarm service.

4. In the **Add Alarm Service** pane that appears, specify the syslog server parameters.

Parameter	Description
Syslog Server	The IP address and port number of the syslog server.
RFC	The syslog protocol. Valid values: RFC3164 and RFC5424.
Protocol	The transport protocol. Valid values: TCP and UDP.
Comment	The description of the syslog server. This information facilitates subsequent identification and management.
Security	Allows you to select the module to which syslog logs are sent.

5. Click **Confirm**. The newly added syslog server will be displayed in the syslog list.
6. Find the newly added syslog server and click  in the **Operation** column to check whether alerts are sent properly.
- If a message appears, indicating that the alert test is successful, the syslog server is added.
 - If an error message appears, WAF cannot connect to the syslog server.

28.7.5.5. Configure alert thresholds

This topic describes how to configure alert thresholds.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Application Security > WAF**. On the **Detection Overview** page, choose **System Management > Syslog Configuration**.
3. Click the **Alarm ThresholdConfiguration** tab and click the edit icon  next to the thresholds that you want to modify.
4. In the edit pane that appears, specify the thresholds.

Threshold	Description
Concurrent Connections	No alerts are sent if a large number of concurrent connections exist.
Number of new connections	No alerts are sent if a large number of new connections are set up.
CPU usage is too high	Alerts are sent if the CPU utilization exceeds a specified percentage in a period of time.
Memory usage is too high	Alerts are sent if the memory usage exceeds a specified percentage in a period of time.

Threshold	Description
Disk usage is too high	Alerts are sent if the disk usage exceeds a specified percentage.

5. Click Ok.

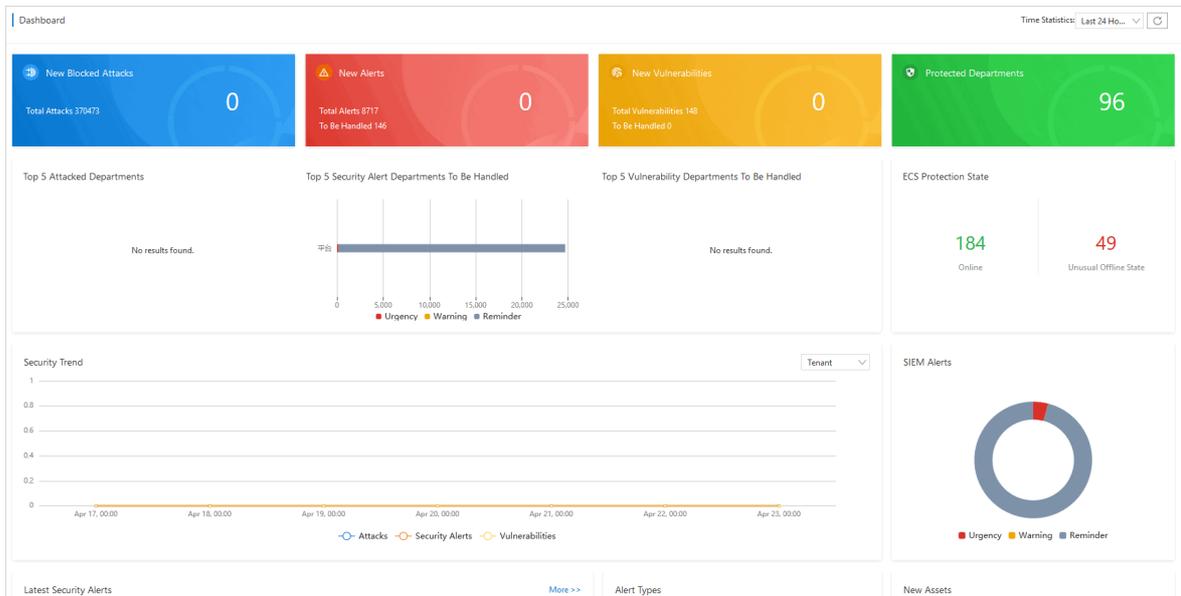
28.8. Security Operations Center (SOC)

28.8.1. View the dashboard

This topic describes how to view the overall security information of the Apsara Stack network environment.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Overview.**
3. In the upper-right corner of the **Dashboard** page that appears, select a time range from the **Time Statistics** drop-down list. You can select *Last 24 Hours*, *Last 7 Days*, or *Last 30 Days* from the drop-down list.
4. View the overall security information.



The dashboard displays the following information:

- **New Blocked Attacks, New Alerts, New Vulnerabilities, and Protected Departments**
- **Top 5 Attacked Departments, Top 5 Security Alert Departments To Be Handled, and Top 5 Vulnerability Departments To Be Handled**
- **Security trend based on the tenant or platform**
- **Latest Security Alerts and Alert Types**
- **Latest Attacks and Attack Types**
- **ECS Protection State, SIEM Alerts, New Assets, and Protected Assets**

28.8.2. Security Monitoring

28.8.2.1. View security monitoring data of tenants

This topic describes how to view the security monitoring data of tenants on the **Attack Protections**, **Security Alerts**, and **Vulnerabilities** tabs.

Attack Protections

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring**. On the Tenant Security Monitoring page that appears, click the **Attack Protections** tab.
3. Set search conditions.

 **Note** Skip this step if you want to view all attack protection events.

All Departments	All Data Sources	All States	All Attack Types	Start time	-	End time	Attack Name/Attached Asset
Search condition	Description						
Department	The department to which the assets affected by the attack belong.						
Data source	The data source.						
State	The attack status.						
Attack type	The attack type.						
Start time and end time	The time range for which you want to view attack events.						
Attack name or asset keyword	The attack name or the keywords of the affected asset.						

4. View the details in the attack list.
5. Click the buttons in the upper-left corner to refresh or export the list.



Security Alerts

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring**. On the Tenant Security Monitoring page that appears, click the **Security Alerts** tab.
3. (Optional)Set search conditions.

 **Note** Skip this step if you want to view all security alerts.

All Departments	All Data Sources	Reminder ×	Warning ×	Urgency ×	All States	All Alert Types	Alert Name/Assets
Search condition	Description						
Department	The department to which the assets associated with the security alerts belong.						
Data source	The data source.						
Level	The alert level. You can select one or more levels. Valid values: <ul style="list-style-type: none"> ○ Urgent ○ Warning ○ Reminder 						

Search condition	Description
Alert state	The alert status.
Alert type	The alert type, which can be set to All Alert Types or a specific type.
Alert name or asset keyword	The alert name or the keywords of affected assets.

4. View details in the security alert list.
5. Click the buttons in the upper-left corner to refresh or export the list.



Vulnerabilities

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Security Operations Center (SOC) > Security Monitoring > Tenant Security Monitoring. On the Tenant Security Monitoring page that appears, click the Vulnerabilities tab.
3. Click the Vulnerabilities or Server Configurations tab.
 - o Vulnerabilities: provides vulnerability information.
 - o Server Configurations: lists risks in server configurations.
4. Set search conditions.

Note Skip this step if you list to view all vulnerabilities or server configuration risks.

Search condition	Description
Department	The department to which the asset affected by the vulnerability or server configuration risk belongs.
Level	The vulnerability level or server configuration risk level.
Type	The vulnerability type or server configuration risk type.
State	The status of the vulnerability or the server configuration risk.
Vulnerability name or asset keyword	The vulnerability or risk name, or the keywords of affected assets.

5. Click the buttons in the upper-left corner to refresh or export a list of vulnerabilities or server configuration risks.



28.8.2.2. View security monitoring data of platforms

This topic describes how to view the security monitoring data of platforms on the Attack Protections, Security Alerts, and Vulnerabilities tabs.

Attack Protections

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Security Operations Center (SOC) > Security Monitoring > Platform

Security Monitoring. On the Platform Security Monitoring page that appears, click the **Attack Protections** tab.

3. Set search conditions to search for attack protection events.

 **Note** Skip this step if you want to view all events.

Search condition	Description
Data source	The data source.
State	The attack status.
Attack type	The attack type.
Attack name or asset keyword	The attack name or the keywords of the affected asset.
Start time and end time	The time range to query.

4. View the details in the attack list.
5. Click the buttons in the upper-left corner to refresh or export the list.



Security Alerts

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Platform Security Monitoring**. On the Platform Security Monitoring page that appears, click the **Security Alerts** tab.
3. (Optional)Set search conditions.

 **Note** Skip this step if you want to view all security alerts.

Search condition	Description
Data source	The data source.
Level	The alert level. You can select one or more alert levels. Valid values: <ul style="list-style-type: none"> ◦ Urgency ◦ Warning ◦ Reminder
Alert state	The alert status.
Alert type	The alert type, which can be set to All Alert Types or a specific alert type.
Alert name or asset keyword	The alert name or the keyword of the affected asset.

4. View details in the security alert list.
5. Click the buttons in the upper-left corner to refresh or export the list.



Vulnerabilities

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Security Monitoring > Platform Security Monitoring**. On the Platform Security Monitoring page that appears, click the **Vulnerabilities** tab and then click **Platform Baseline**.
3. Set search conditions.

 **Note** Skip this step if you want to view all baseline risks of the platform.

Search condition	Description
Level	The risk level.
Type	The baseline risk type.
State	The processing status.
Risk name or asset name	The risk name or the keywords of the affected asset.

4. Click the buttons in the upper-left corner to refresh or export the list.



28.8.3. Asset Management

28.8.3.1. View tenant assets

This topic describes how to view the assets of tenants, including ECS instances, RDS instances, OSS instances, SLB instances, and Elastic IP addresses (EIPs).

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Asset Management > Tenant Assets**.
3. Select the target service, for example, **Elastic Compute Service (ECS)**.
4. Set search conditions to view a target asset.

 **Note** Skip this step if you want to view all assets.

Search condition	Description
Department	The department to which the asset belongs.
VPC	The VPC to which the asset belongs.
State	The running status of the asset.
New	Specifies whether the asset to query is newly added.
Server name or IP address	The keywords of the asset name.

5. View asset information in the asset list.

28.8.3.2. View platform assets

This topic describes how to view the assets of the platform.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Asset Management > Platform Assets**.
3. Set search conditions to view a target asset.

 **Note** Skip this step if you want to view all assets.

4. View asset information in the asset list.

28.8.4. Log Analysis

28.8.4.1. View log analysis results

This topic describes how to view log analysis results.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Log Analysis**.
3. Set search conditions to view the target information.
4. View log analysis results.

28.8.4.2. Security Audit

28.8.4.2.1. Overview

A security audit refers to the systemic and independent inspection and verification of activities and behavior in the computer network environment. Delegated by property owners and authorized by management authorities, professional auditors give their assessments according to relevant laws and regulations. When the administrator needs to backtrack system operations, the administrator can perform a security audit.

Security audits are long-term security management activities throughout the lifecycle of cloud services. The security audit feature of Apsara Stack Security can collect system security data, analyze weaknesses in system operations, report audit events, and classify audit events into important, moderate, and low risk levels. The security administrator views and analyzes audit events to continuously improve the system and ensure the security and reliability of cloud services.

28.8.4.2.2. View security audit overview

This topic describes how to view the summarized information of security audit.

Context

The **Overview** page provides reports on the raw log trend, audit event trend, audit risk distribution, and security issue distribution. The reports are displayed in the form of a run chart or pie chart to help the security administrator analyze the trend of risks facing your cloud services.

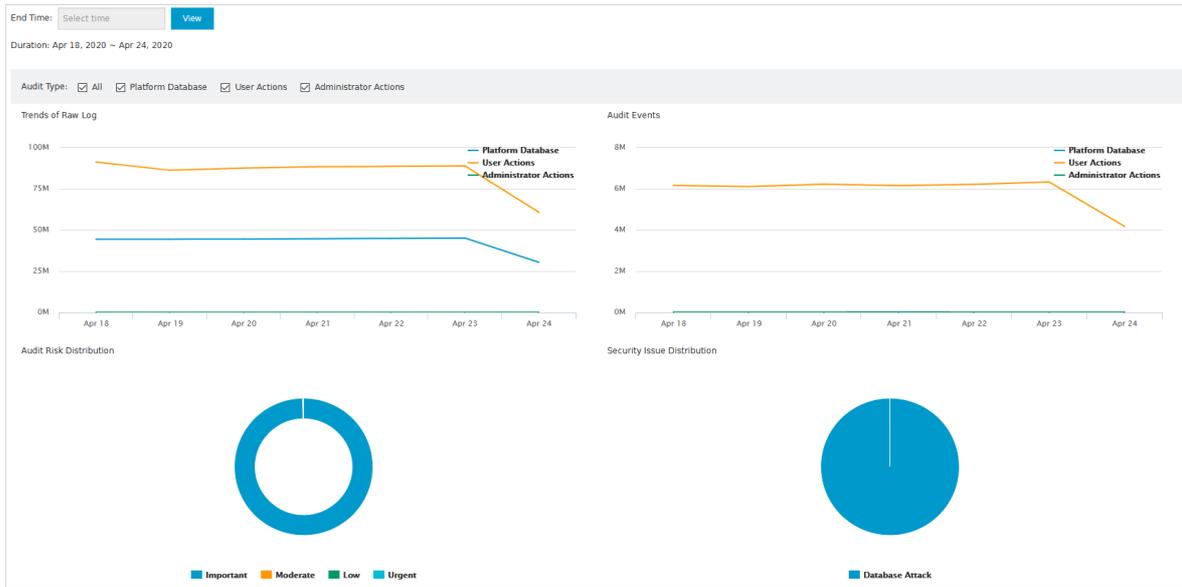
On the **Overview** page, the security administrator can check the number of log entries and the storage usage of each audit type in a specified time range.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**.

On the Security Audit page, click the Overview tab.

3. On the Overview tab, view the audit summary for the last seven days.



○ Trends of Raw Log

This chart displays the trend of logs generated by physical servers, network devices, RDS instances, ECS instances, and the API in the last seven days. The security administrator can analyze the trend in the number of logs to check whether the number of log entries is at a normal level.

○ Audit Events

This chart displays the trend of audit events generated by physical servers, network devices, RDS instances, ECS instances, and the API in the last seven days. The security administrator can analyze this chart to check whether the number of audit events is at a normal level.

○ Audit Risk Distribution

This chart displays the percentage distribution of audit events of different risk levels in the last seven days. Risk levels include important, moderate, and low. The security administrator can analyze this chart to check whether the audit events are at acceptable risk levels.

○ Security Issue Distribution

This chart displays the percentage distribution of different event types in the last seven days. The security administrator can analyze this chart to check for the most frequent audit events and identify potential risks to improve security protection.

○ Log Size

This chart displays the size of online logs and offline logs. If these logs consume many storage resources, we recommend that you back up required audit logs and delete unnecessary logs.

○ Audit Log Size

This chart displays the size of logs for each audit type.

4. View the audit summary in a specified time period.

- i. Specify End Time as the end time of the period to query.
- ii. In Audit Type, select the audit types to query.
- iii. Click View to view the audit summary for the last seven days before the specified end time.

28.8.4.2.3. Query audit events

This topic describes how to query audit events.

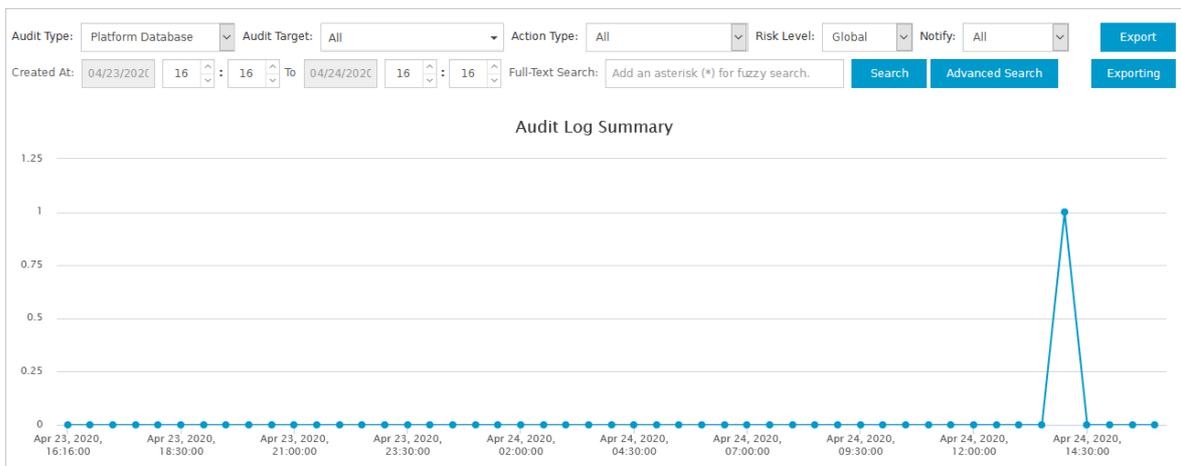
Context

On the **Audit Query** tab, you can view the details of audit events, including log creation time, audit type, audit object, action type, risk level, and log content.

The system matches the logs collected by a security audit module with audit rules. If the log content matches one regular expression in the audit rules, an audit event is reported. For more information about audit rules, see [Add an audit policy](#).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the Security Audit page, click the **Audit Query** tab.
3. On the **Audit Query** tab, configure query conditions to view audit events within a specific time range.



- **Basic query**
 - a. Configure **Audit Type**, **Audit Target**, **Action Type**, **Risk Level**, and **Notify**.
 - b. Specify a time range to query.
 - c. In the **Full-Text Search** search box, enter a keyword.
 - d. Click **Search**.
 - **Advanced query**

In addition to the basic query conditions, you can also configure advanced query conditions.

 - a. Configure basic query conditions.
 - b. Click **Advanced Search**.
 - c. Under **Filter Condition**, specify **User**, **Target**, **Action**, **Result**, and **Cause**.
 - d. Click **Save**.
4. Click **Export** to export the data. Download the exported file for analysis. For more information, see [Manage export tasks](#).

28.8.4.2.4. View raw logs

This topic describes how to view raw audit logs.

Context

On the **Raw Log** tab, you can view the raw logs generated by a running audit object. Raw logs contain information required for debugging. Security administrators can use raw logs to troubleshoot system failures.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the Security Audit page, click the **Raw Log** tab.
3. On the **Raw Log** tab, configure query conditions to view the log summary chart and raw logs within a specific time range.
 - i. Specify **Audit Type** and **Audit Target**.
 - ii. Enter a keyword.
 - iii. Specify a time range to query.
 - iv. Click **Search**.
4. Click **Export** to export the data. Download the exported file for analysis. For more information, see **Manage export tasks**.

28.8.4.2.5. Manage log sources

This topic describes how to view and manage log sources.

Context

You can view the number of log entries and specify whether to show logs by log type or log source.

- The **Log Types** sub-tab provides the number of all log entries for a specific audit object.
- The **Log Sources** sub-tab provides the number of log entries for all audit objects of a specific device instance.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the Security Audit page, click the **Log Sources** tab.
3. Click the **Log Types** sub-tab and view the number of log entries from each audit object.

You can view the number of log entries recorded on the current day and the number of log entries recorded during the last 30 days from each audit object.

If you do not want to show the log data from a specific audit log, perform the following steps:

- i. Find the target audit object and click **Hide** in the **Actions** column.
- ii. In the Note message, click **Confirm**.

 **Note** The procedure used to show log data is similar to that used to hide log data.

4. Click the **Log Sources** sub-tab and view the number of log entries from each device instance.

You can view the number of log entries recorded on the current day and the number of log entries recorded during the last 30 days from each device instance.

If you do not want to show the log data from a specific device instance, perform the following steps:

- i. Find the target device instance and click **Hide** in the **Actions** column.
- ii. In the Note message, click **Confirm**.

 **Note** The procedure used to show log data is similar to that used to hide log data.

28.8.4.2.6. Policy settings

28.8.4.2.6.1. Manage audit rules

This topic describes how to add, modify, or remove an audit rule.

Context

If a log entry matches an audit rule, an audit event is reported. You can set regular expressions in an audit rule to match log entries. A regular expression defines a character string matching pattern, which can be used to check whether a string contains a specific substring. Example:

Regular expression	Description
<code>^\d{5,12}\$</code>	Matches the consecutive numbers from the fifth number to the twelfth number.
<code>load_file\{</code>	Matches the "load_file{" string.

The security audit module defines the default audit rule based on the string generated in the log when an audit event is reported. The security administrator can also customize audit rules based on the string generated in the log when the system encounters an attack.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the Security Audit page, click the **Policies** tab.
3. On the **Policies** tab, click the **Audit Rules** sub-tab.
4. Add an audit rule.
 - i. Click **New** in the upper-right corner.
 - ii. In the **Add Policy** dialog box, configure parameters.

Add Policy
✕

Policy Name

Audit Type: Platform Database ▼

Audit Target: Global ▼

Action Type: Resource Management ▼
Risk Level: Important ▼

Notify: Enable Alert ▼

Filter Condition:

Field	Operator	Value	Action
User	Equals	Enter a user	✕
		+	
Target	Equals	Enter a target	✕
		+	
Action	Equals	Enter a command	✕
		+	
Result	Equal	Search by result keyword	

Add Cancel

- iii. Click **Add**.

After you add an audit rule, if one string in an audit log of the specified audit type, audit object, or risk level matches the regular expression of the audit rule, an alert email is sent to the specified recipient.

5. Manage audit rules. You can add, query, disable, enable, and remove audit rules.
 - Query audit rules
Specify **Audit Type** and **Audit Target**. Enter a keyword in the search bar and click **Search**.
 - Disable an audit rule
Find the target audit rule and click **Disable** in the **Actions** column.
 - Enable an audit rule
Find the target audit rule that has been disabled and click **Enable** in the **Actions** column.
 - Remove an audit rule
Find the target audit rule and click **Delete** in the **Actions** column.

 **Note** You can remove only custom rules.

28.8.4.2.6.2. Configure alert recipients

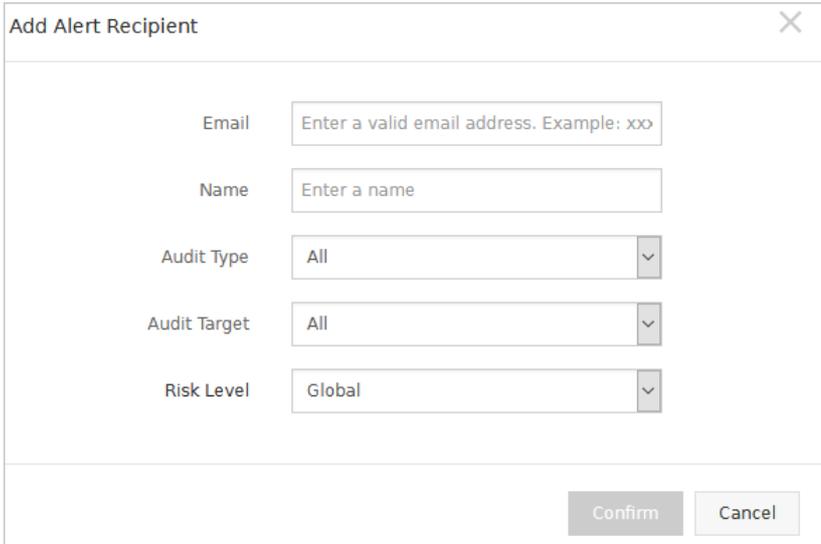
This topic describes how to configure the recipients of alerts on audit events.

Context

You can add an alert recipient by entering an email address that can be used to receive alerts on audit events.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the Security Audit page, click the **Policies** tab.
3. On the **Policies** tab, click the **Alert Settings** sub-tab.
4. Add an alert recipient.
 - i. Click **New** in the upper-right corner.
 - ii. In the **Add Alert Recipient** dialog box, configure information about the target alert recipient.



- iii. Click **Confirm**.
5. Manage alert recipients.
 - Search for alert recipients

Specify **Audit Type**, **Audit Target**, and **Risk Level**, enter a keyword of the email address, and click **Search**.

- Remove alert recipients

Find the target email address and click **Delete** in the **Actions** column.

28.8.4.2.6.3. Manage archives of events and logs

This topic describes how to query and download the archives of audit events and raw logs.

Context

To ensure the security of the Apsara Stack environment, you can download the archives of events and logs to analyze audit events.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the **Security Audit** page, click the **Policies** tab.
3. On the **Policies** tab, click the **Archiving** sub-tab.
4. Query the archives of events and logs.
 - i. Specify **Audit Type** and **Archiving Type**.
 - ii. Specify a time range to query.
 - iii. Click **Search**.
5. Find the target file where the archive information is stored and click **Download** in the **Actions** column to save the archive file to your local machine.

28.8.4.2.6.4. Manage export tasks

This topic describes how to download or delete exported audit events and logs.

Context

After you export audit events or logs on the **Audit Query** or **Raw Log** tab of the **Security Audit** page, you can manage the export tasks on the **Exporting** sub-tab.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit**. On the **Security Audit** page, click the **Policies** tab.
3. On the **Policies** tab, click the **Exporting** sub-tab.
4. View the created export tasks.

Created At	Export Task ID	Task Type	Filter Condition	Task Status	Format	Actions

5. Click **Download** to download audit events or log files to your local machine.
6. Click **Delete** to delete the export task.

28.8.4.2.6.5. Modify system settings

This topic describes how to configure system parameters for security audit.

Context

By configuring system parameters, you can configure the maximum number of system alerts per day and the maximum number of audits per day for raw logs.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Log Analysis > Security Audit.** On the Security Audit page, click the **Policies** tab.
3. On the **Policies** tab, click the **System Settings** sub-tab.
4. Find the target configuration item and click **Edit** in the **Actions** column.

Audit Rules				
Alert Settings Archiving Exporting <u>System Settings</u>				
ID	Description	Updated At	Value	Actions
1	Maximum Alerts per Day	Nov 20, 2019, 00:44:19	1000	Edit
2	Total Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	500	Edit
3	Database Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	-1	Edit
4	Server Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	-1	Edit
5	Network Device Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	-1	Edit
6	User Operation Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	-1	Edit
7	Administration Logs Audited per Day (GB/day)	Nov 20, 2019, 00:44:19	-1	Edit

5. Enter a required value in the **Value** column and click **Confirm** in the **Actions** column.

28.8.5. Rules

28.8.5.1. Create traffic monitoring IPS rules

This topic describes how to add Intrusion Prevention System (IPS) rules to Cloud Firewall. Cloud Firewall has built-in IPS rules. This topic describes how to customize IPS rules based on your business requirements and network environment.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules.**
3. On the **Rules** page, click the **Cloud Firewall IPS Rules** tab.
4. Click **Create Rule** in the upper-right corner of this tab.
5. In the **Create Rule** pane, configure rule-related parameters.

Parameter	Description
Rule Name	The name of the IPS rule. We recommend that you enter a name that indicates the purpose of the rule.
Rules Engine	The rules engine. Valid values: Basic Policies and Virtual Patches .
Attack Type	The attack type to be detected by the rule.
Severity	The severity. Valid values: Low , Medium , and High .
CVE	The CVE ID of the vulnerability listed in the rule. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ? Note Common Vulnerabilities and Exposures (CVE) provides the publicly known information-security vulnerabilities. CVE IDs are allocated by a CVE Numbering Authority (CNA). </div>
Application	The name of the attacked application.

Parameter	Description
Rule Mode	The rule mode. Valid values: Packet and Traffic.
Direction	The direction of traffic to be monitored by IPS. Valid values: Inbound and Outbound, Inbound, and Outbound.
Rule Content	The rule content that is specified by using the Snort syntax. <div style="background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <p> Note To avoid negative impacts on your business, make sure that you enter the correct rule content.</p> </div>
Rule Description	The rule description. We recommend that you enter information such as the purpose and impact of the rule.
Description	The remarks about the rule. We recommend that you enter information such as the purpose and impact of the rule.

6. Click **OK**.

28.8.5.2. Manage IPS rules of Cloud Firewall

This topic describes how to view, enable, and disable the IPS rules of Cloud Firewall.

Context

On the **Cloud Firewall IPS Rules** tab of the **Rules** page in the Apsara Stack Security console, you can view the built-in and custom IPS rules, and enable or disable the rules based on your business requirements.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. On the **Rules** page, click the **Cloud Firewall IPS Rules** tab.
4. **Manage IPS rules of Cloud Firewall.** In the list of IPS rules, you can view rule details, enable rules, and disable rules.
 - **View rule details**
Find the target rule and click **Details** in the **Actions** column to view the rule details.
 - **Enable a rule**
Find the target rule and click the toggle in the **Enable or not** column to change the rule state from **Disable** to **Enable**.
 - **Disable a rule**
If a rule is not suitable for your business, you can disable it.
Find the target rule and click the toggle in the **Enable or not** column to change the rule state from **Enable** to **Disable**.

28.8.5.3. Create traffic monitoring IDS rules

This topic describes how to create traffic monitoring IDS rules.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.

3. On the Rules page that appears, click the **Traffic Monitoring IDS Rules** tab.
4. Click **Create Rule** in the upper-right corner of this tab.
5. In the **Create Rule** pane, configure rule-related parameters.

Parameter	Description
Rule Name	The name of the IPS rule. We recommend that you enter a name that indicates the purpose of the rule.
Rules Engine	The rules engine. Valid values: Basic Policies and Virtual Patches .
Attack Type	The attack type to be detected by the rule.
Severity	The severity. Valid values: Low , Medium , and High .
CVE	<p>The CVE ID of the vulnerability listed in the rule.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Note Common Vulnerabilities and Exposures (CVE) provides the publicly known information-security vulnerabilities. CVE IDs are allocated by a CVE Numbering Authority (CNA).</p> </div>
Application	The name of the attacked application.
Rule Mode	The rule mode. Valid values: Packet and Traffic .
Direction	The direction of traffic to be monitored by IPS. Valid values: Inbound and Outbound , Inbound , and Outbound .
Rule Content	<p>The rule content that is specified by using the Snort syntax.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Note To avoid negative impacts on your business, make sure that you enter the correct rule content.</p> </div>
Rule Description	The rule description. We recommend that you enter information such as the purpose and impact of the rule.
Description	The remarks about the rule. We recommend that you enter information such as the purpose and impact of the rule.

6. Click **OK**.

28.8.5.4. Manage traffic monitoring IDS rules

This topic describes how to view, enable, and disable traffic monitoring IDS rules.

Context

On the **Traffic Monitoring IDS Rules** tab, you can view the built-in and custom IDS rules, and enable or disable the rules based on your business needs.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. On the Rules page that appears, click the **Traffic Monitoring IDS Rules** tab.
4. Manage traffic monitoring IDS rules. In the IDS rule list, you can view rule details, enable rules, and disable rules.

- View rule details
Find the target rule and click **Details** in the **Actions** column to view the rule details.
- Enable a rule
Find the target rule and click the toggle in the **Enable or not** column to change the rule state from **Disable** to **Enable**.
- Disable a rule
If a rule is not suitable for your business, you can disable it.
Find the target rule and click the toggle in the **Enable or not** column to change the rule state from **Enable** to **Disable**.

28.8.5.5. Customize DDoS traffic scrubbing policies and traffic rerouting thresholds

This topic describes how to customize DDoS traffic scrubbing policies and traffic rerouting thresholds. Default traffic rerouting thresholds have been configured. You can perform the following steps to customize the thresholds as needed.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules**.
3. Customize the DDoS traffic scrubbing thresholds.
 - i. Choose **AliGuard Rules > Scrubbing Policy**.
 - ii. In the **Actions** column for the target rule, click **Modify Threshold**.
 - iii. In the **Modify Threshold** dialog box that appears, enter a threshold value.
 - iv. Click **OK**.
4. Customize the traffic rerouting thresholds.
 - i. Choose **AliGuard Rules > Reroute Threshold**.
 - ii. In the **Actions** column for the target rule, click **Modify Threshold**.
 - iii. In the **Modify Threshold** dialog box that appears, enter a threshold.
 - iv. Click **OK**.

28.8.5.6. View Server Guard rules

This topic describes how to view the operation status of Server Guard, including the list of vulnerabilities, baselines, and host exceptions.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Rules > Server Guard Rules**.
3. In the overview section, you can view the total number of vulnerability libraries, number of baselines, and number of host exceptions as well as the available engines.
4. View the vulnerability list.
 - i. Click the **Vulnerabilities** tab.
 - ii. In the overview section, you can view the total number of Linux vulnerabilities, total number of Windows vulnerabilities, total number of Web-CMS vulnerabilities, and total number of emergency vulnerabilities.

- iii. Set search conditions to view the vulnerability list.

 **Note** Skip this step if you want to view all vulnerabilities.

In the vulnerability list, you can view the vulnerability name, CVE ID, vulnerability type, system, update time, and status.

5. View the baseline list.

- i. Click the **Baselines** tab.
- ii. In the overview section, you can view the number of baseline types and check items.
- iii. Set search conditions to view the baseline list.

 **Note** Skip this step if you want to view all baselines.

In the baseline list, you can view the baseline type, check item category, check item name, risk level, update time, and status.

6. View the host exception list.

- i. Click the **Host Exceptions** tab.
- ii. In the overview section, you can view the number of rule alert subcategories, number of webshells, and number of malicious viruses.
- iii. Set search conditions to view the host exception list.

 **Note** Skip this step if you want to view all exceptions.

In the host exception list, you can view the subcategory name, rule category, risk level, update time, source, and status.

28.8.6. Create a report task

This topic describes how to create a report task. After you create a report task, the system sends reports on a regular basis.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > Report Management**.
3. On the Report Management page that appears, click **Create Report**.
4. In the **Create Report** dialog box that appears, specify the parameters.

Parameter	Description
Report Name	The name of the report task. We recommend that you enter information such as the report purpose for easier identification and management.
Task Type	The task type. Valid values: Daily Report, Weekly Report, and Monthly Report.
Department	The department related to the report.
Email Box	The email address of the report recipient. Separate multiple email addresses with commas (,).

5. Click **OK**.

Result

In the report task list, you can view, edit, and delete the created report tasks.

28.8.7. System Configurations

28.8.7.1. Alert settings

28.8.7.1.1. Set alert recipients

This topic describes how to add and manage alert recipients.

Context

Apsara Stack Security sends alerts to alert recipients by SMS, email, or DingTalk. When the detected information matches an alert rule, an alert notification is sent to the alert recipient.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings > Alert Recipient**.
3. Click **Add Recipient**.
4. Enter the recipient information and click **OK**.

Recipient Name	Mobile Number	Email	DingTalk	Actions
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add Recipient"/> <input type="button" value="OK"/> <input type="button" value="Cancel"/>

5. **Manage alert recipients.** In the recipient list, find a recipient and click **Edit** in the **Actions** column to edit the recipient information.

28.8.7.1.2. Set alert notifications

This topic describes how to set the alert notification method for security events on tenants or platforms.

Context

In the **Alerts** section, the security administrator can set the alert notification method for security events. When a security event occurs, the system notifies the alert recipients by email, SMS, or DingTalk. For more information about how to set alert recipients, see [Set alert recipients](#).

Alerts on tenants

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings > Tenant Alerts**.
3. In the **Alerts** section, select a notification method for each security event.

Alerts		
	<input type="checkbox"/> All	<input type="checkbox"/> All
Security Events	Notification Method	
Logon Security: Unusual Logon The account has been logged on in an disapproved location.	<input type="checkbox"/> Mobile Number	<input type="checkbox"/> Email
Emergency Alerts	Notification Method	
Website Defacement An attack that changes the visual appearance of the site, which can adversely affect SEO performance and cause the site to be flagged as malicious by the search engine.	<input type="checkbox"/> Mobile Number	<input type="checkbox"/> Email
Zombie Attack If a server launches DDoS attacks or brute-force attacks on other servers, it may have been controlled by attackers.	<input type="checkbox"/> Mobile Number	<input type="checkbox"/> Email

4. **Click Confirm.**

Alerts on the platform

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Alert Settings > Platform Alerts**.
3. In the **Alerts** section, select a notification method for each security event.

Platform Security Monitoring	Alert Level	Notification Method
Security Alerts The detected invasions on the platform, including webshells, malicious code, and suspicious processes.	Urgent	<input checked="" type="checkbox"/> DingTalk <input checked="" type="checkbox"/> Mobile Number <input checked="" type="checkbox"/> Email
Attack Protection The handled attacks, including Web attacks, CC attacks, port scanning, brute-force attacks, and DDoS attacks.	Important	<input checked="" type="checkbox"/> DingTalk <input checked="" type="checkbox"/> Mobile Number <input checked="" type="checkbox"/> Email
Vulnerabilities (Platform Inspection) The detected vulnerability configurations of the platform, such as inappropriate network access control.	Moderate	<input checked="" type="checkbox"/> DingTalk <input checked="" type="checkbox"/> Mobile Number <input checked="" type="checkbox"/> Email

[Confirm](#)

4. Click **Confirm**.

28.8.7.2. Updates

28.8.7.2.1. Overview of the system updates feature

The system updates feature allows you to manually or automatically update the Apsara Stack Security and rule libraries for up-to-date protection.

The supported package import method depends on the Apsara Stack network environment.

- If Apsara Stack is connected to the Internet, you can choose **Automatically Download Update Packages**.
- If Apsara Stack is not connected to the Internet, you can choose **Manually Import Update Packages**.

The following table lists the update statuses of a rule library.

Update statuses of a rule library

Status	Description
To Be Updated	Indicates that a new version of the rule library is available for update.
Updating	Indicates that the rule library is being downloaded from Alibaba Cloud for update.
Updated	Indicates that the rule library has been updated.
Update Failed	Indicates that the rule library failed to be updated.

28.8.7.2.2. Automatically download an update package

This topic describes how to automatically download update packages and update the services.

Context

If the Apsara Stack environment can connect to the Internet, you can enable automatic download of update packages to update the rule libraries.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Turn on **Auto Update Configuration** to enable automatic download of update packages. After this switch is turned on, the system automatically downloads update packages on a regular basis.

4. Update a rule library. You can update one or more rule libraries at a time.
 - Update multiple rule libraries at a time
Click **Batch Update** in the upper-right corner to update all rule libraries.
 - Update a single rule library
 - a. Click the tab of the rule type you want to update. For example, click **Server Security**.
 - b. Click **Update** in the **Actions** column.

28.8.7.2.3. Manually import an update package and update the services

This topic describes how to manually import an update package and update the services.

Prerequisites

The security administrator has obtained the offline update package.

Context

If the Apsara Stack environment cannot connect to the Internet, you can update a rule library after you import an offline update package.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Manually import an update package.
 - i. Click **Import Update Package** next to **Manual Update** in the upper-left corner.
 - ii. In the **Import Update Package** dialog box that appears, click **Browse** to select an offline update package that is downloaded to your local device.
 - iii. Click **Confirm**.
4. Update a rule library. You can update one or more rule libraries at a time.
 - Update multiple rule libraries at a time
Click **Batch Update** in the upper-right corner to update all rule libraries.
 - Update a single rule library
 - a. Click the tab of the rule type you want to update. For example, click **Server Security**.
 - b. Click **Update** in the **Actions** column.

28.8.7.2.4. Roll back a rule library

This topic describes how to roll a rule library back to a previous version.

Context

If an error occurs with an updated rule library, you can roll it back to a previous version to avoid service interruption.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Click the tab of the target rule library, such as **Server Security**.

4. In the Actions column for the target rule library, choose **More > Roll Back**.
5. In the **Version Rollback** dialog box that appears, click **Confirm**.

28.8.7.2.5. View update history of a rule library

This topic describes how to view the update history of a rule library.

Context

You can view the update history of a rule library. If an error occurs with the latest version, you can locate the problem and roll the rule library back to an earlier version.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Updates**.
3. Click the tab of the target rule library, such as **Server Security**.
4. In the Actions column for a rule library, click **History**. On the **Previous Updates** page, you can view the update history of the rule library. Click **Details** to view the details of an update package.

28.8.7.3. Global Settings

28.8.7.3.1. Set CIDR blocks for traffic monitoring

28.8.7.3.1.1. Add a CIDR block for traffic monitoring

This topic describes how to add a Classless Inter-Domain Routing (CIDR) block for traffic monitoring. Network Traffic Monitoring System of Apsara Stack Security monitors the traffic of a specified CIDR block.

Context

CIDR blocks are configured for Network Traffic Monitoring System. The security administrator can change the CIDR blocks for monitoring as needed. The settings of CIDR blocks apply only to a data center that is deployed in the region to which the specified CIDR block belongs.

Note

Changes to CIDR block settings take effect immediately without the intervention of security administrators. If you add the same CIDR block on the traffic collection CIDR block setting page and region setting page, ensure that you select the same region on both pages.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings > Traffic Collection IP Range**.
3. Click **Add**.
4. In the **Add CIDR Block for Monitoring** dialog box that appears, specify a CIDR block.

- **CIDR Block:** Enter a CIDR block for traffic monitoring.

 **Note** Note that the CIDR block that you entered must be valid and unique.

- **Region:** Specify the region of the data center.

5. Click **Confirm**.

28.8.7.3.1.2. Manage CIDR blocks for traffic monitoring

This topic describes how to modify or delete Classless Inter-Domain Routing (CIDR) blocks for traffic monitoring.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings > Traffic Collection IP Range**.
3. Select a region, enter the target CIDR block, and then click **Search**.

View the information about the CIDR block for traffic monitoring and the region in the search result.

4. In the **Actions** column, manage a CIDR block for traffic monitoring by clicking a button.
 - **Modify the CIDR block for traffic monitoring.**
Click **Modify** to modify the region of the CIDR block for traffic monitoring.
 - **Delete the CIDR block for traffic monitoring.**
Click **Delete** to delete the CIDR block for traffic monitoring.

28.8.7.3.2. Region settings

28.8.7.3.2.1. Add a CIDR block for a region

This topic describes how to add Classless Inter-Domain Routing (CIDR) blocks for regions that are detected and reported by Server Guard.

Context

Region settings are used for region detection of Server Guard agents. Server Guard servers automatically detect and match the regions of servers based on the IP address information reported by Server Guard agents.

 **Note** You can change the region of a CIDR block. After modification, you must modify the region for all assets in the CIDR block on the Asset Overview page.

Procedure

1. [Log on to Apsara Stack Security Center.](#)

2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings > Region**.
3. Click **Add**.
4. In the **Add CIDR Block** dialog box that appears, set the parameters.

- **CIDR Block:** Enter a CIDR block for the region.

 **Note** Enter a valid CIDR block. You cannot enter a CIDR block that is configured in the system.

- **Region:** Specify the region.

5. Click **Confirm**.

28.8.7.3.2.2. Manage CIDR blocks for a region

This topic describes how to modify or delete Classless Inter-Domain Routing (CIDR) blocks for a region.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings > Region**.
3. Select a region, enter the target CIDR block, and then click **Search**. View the information about the CIDR block for the region in the search result.
4. In the **Actions** column, manage a CIDR block for the region by clicking a button.
 - **Modify the CIDR block for the region.**
Click **Modify** to modify the CIDR block for the region.
 - **Delete the CIDR block for the region.**
Click **Delete** to delete the CIDR block for the region.

28.8.7.3.3. Configure whitelists

This topic describes how to configure the brute-force attack blocking whitelist in Server Guard and the following whitelists in Threat Detection Service (TDS): IP addresses allowed by server brute-force attack blocking, IP addresses allowed by application attack blocking, and IP addresses allowed by web attack blocking.

Context

If a normal request is regarded as an attack by the attack blocking function of TDS or the unusual logon detection function of Server Guard, you can add the source IP address to the whitelist to avoid further false positives.

 **Note** Make sure that the IP addresses in the whitelist can be trusted.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the Global Settings page, click the **Whitelist** tab.
3. Click **Add**.
4. In the **Add to Whitelist** dialog box, configure the following parameters.

Parameter	Description
Source IP	Enter a source IP address or Classless Inter-Domain Routing (CIDR) block.
Username	The name of the user who creates the whitelist.
Type	<ul style="list-style-type: none"> ◦ Brute-Force Attack Blocking Whitelist: Server Guard does not alert you on brute-force attacks or unusual logons that are started by the IP addresses in this whitelist. ◦ Beaver WAF Whitelist: The attack blocking function does not alert you on the web attacks that are started by the IP addresses in this whitelist. ◦ Servers with Brute-Force Attack Permissions: The attack blocking function does not alert you on brute-force attacks that are started by the IP addresses in this whitelist. ◦ IPs with Application Attack Permissions: The traffic from the IP addresses in this whitelist is not detected as suspicious application attack traffic.

5. Click **Confirm**.
If you want to delete an existing whitelist, click **Delete** in the **Actions** column. In the **Delete Whitelist** message, click **Confirm**.

28.8.7.3.4. Configure request blocking policies

This topic describes how to enable web attack blocking and brute-force attack blocking.

Context

The attack blocking functions protect your servers against web attacks and brute-force attacks.

Procedure

1. [Log on to Apsara Stack Security Center.](#)

2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the Global Settings page, click the **Policy Configuration** tab.
3. Turn on or off the switches in the **Actions** column to enable or disable **Web Attack Blocking** or **Brute-Force Attack Blocking**.

Category	Status	Description	Actions
Web Attack Blocking	Disabled	 Web attack blocking is disabled. Only the warning function is provided.	
Brute-Force Attack Blocking	Disabled	 Brute-Force attack blocking is disabled. Only the warning function is provided.	

 **Note**

In the **Actions** column, a red switch indicates a disabled function and a green switch indicates an enabled function.

After you disable the blocking function for an attack type, Apsara Stack Security Center provides only the alert function on the attacks.

28.8.7.3.5. Block IP Addresses

This topic describes how to manually block requests from a specified IP address based on traffic analysis results provided by Cloud Firewall.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the Global Settings page, click the **Block IP Addresses** tab.
3. Click **Add**.
4. In the **Add** dialog box that appears, specify the IP address to block.

Add ✕

Source IP

Destination IP

Destination Port

Blocking Duration --Select-- ▼

Type Blacklist ▼

Note: The whitelist mechanism has precedence over the blacklist.

Confirm
Cancel

Parameter	Description
Source IP	Enter the source IP address that you want to block.
Destination IP	Enter the destination IP address to which the requests need to be blocked.

Parameter	Description
Destination Port	Enter the destination port to which the requests need to be blocked.
Blocking Duration	Select a time range for which you want to block requests. You can select 1 Day, 7 Days, or 30 Days.
Type	Set the blocking mode to Whitelist or Blacklist.

5. Click Confirm.

28.8.7.3.6. Configure custom IP addresses and locations

28.8.7.3.6.1. Add custom IP addresses and locations

This topic describes how to add custom IP addresses and locations. You can customize internal IP addresses based on your network planning. After that, IP addresses from the public address library do not match the addresses outside of China.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the Global Settings page, click the **Custom IP Location** tab.
3. Click **Add**. If you want to add multiple IP addresses and locations at a time, click **Batch Upload (.txt)** to import multiple IP addresses and locations as a template.
4. In the Add dialog box, specify the custom IP address and location.
5. Click **Confirm**.

28.8.7.3.6.2. Manage custom IP addresses and locations

This topic describes how to modify and delete custom IP addresses and locations.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Global Settings**. On the Global Settings page, click the **Custom IP Location** tab.
3. In the **Actions** column, manage custom IP addresses and locations.
 - To modify a custom IP address and a location:
Click **Modify** to modify the custom IP address and location.
 - To delete a custom IP address and a location:
Click **Delete** to delete the custom IP address and location.

28.8.7.4. System Monitoring

28.8.7.4.1. Configure CIDR blocks for traffic redirection in Cloud Firewall

Before you use Cloud Firewall, configure Classless Inter-Domain Routing (CIDR) blocks for traffic redirection.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configuration > System Monitoring**. On the System Monitoring tab, click the ICFW tab.
3. On the ICFW tab, click **Add** on the right of **ISW Traffic Diversion Settings - Business CIDR Block**.
4. In the **Add** dialog box, specify **CIDR Block for Traffic Diversion** and **Type**.
5. Click **OK**.
After the configuration is complete, you can view related information in **ISW Traffic Diversion Settings - Business CIDR Block**, **Service Self-check Status**, and **Interface Status**.

28.8.7.5. Remote operations

28.8.7.5.1. Enable Remote O&M

This topic describes how to enable remote operations.

Context

The remote operations function provides remote security operations and rules operations.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Remote O&M**.
3. On the **Remote O&M** page, turn on the switch to enable remote operations.

Note The function is enabled if the switch turns green.

4. Select the departments for which you want to enable remote operations.
 - i. Click **Update** on the right of **Enabled Departments**.
 - ii. In the **Update Enabled Departments** dialog box, click the **Department** drop-down list to select the departments for which you want to enable remote operations.

iii. Click OK.

Note After you perform these operations, the security logs of Enabled Departments are encrypted and uploaded to Apsara Stack Security Center.

5. Select fields to encrypt and upload for remote operations.
 - If you select **Required Nonsensitive Fields**, the system encrypts and uploads the data.
 - If you select **Available Fields and Masking**, the system masks the data before encrypting and uploading it.

28.8.7.6. Account management

28.8.7.6.1. View and modify Apsara Stack accounts

This topic describes how to view and modify the information of Apsara Stack accounts bound to the system.

Context

Note All assets in Apsara Stack Security are bound to Alibaba Cloud accounts. Proceed with caution when you modify the account information.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Accounts**. On the Accounts page, click the **Apsara Stack Account** tab.
3. Modify the information of your Apsara Stack account.
 - i. In the Actions column, click **Modify**.
 - ii. In the **Change Account** dialog box, modify the account information.

The screenshot shows a 'Change Account' dialog box with the following fields:

- Apsara Stack Account
- User ID
- Access Key
- Access Secret (masked with dots)

Buttons: Confirm, Cancel

iii. Click **Confirm**.

4. View the details of your Apsara Stack account.
 - i. In the Actions column, click **Details**.
 - ii. View the account details.

The details include the license expiration date and the number of Server Guard licenses. The information is obtained based on the user ID and the AccessKey pair.

Details	
Apsara Stack Account:	[Redacted]
User ID:	[Redacted]
Access Key:	[Redacted]
Access Secret:	*****
License Due Date:	05/16/2020
Server Guard Licenses:	0

[Confirm](#)

28.8.7.6.2. Add a public cloud account

This topic describes how to add a public cloud account in Apsara Stack Security Center to use features in a hybrid cloud.

Context

After you add a public cloud account in Apsara Stack, you can manage the Anti-DDoS Pro, Anti-DDoS Premium, and Web Application Firewall (WAF) instances under this public cloud account in Apsara Stack. This allows you to use features in a hybrid cloud.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Security Operations Center (SOC) > System Configurations > Accounts**. On the Accounts page, click the **Public Cloud Account** tab.
3. Click **Add**.
4. In the **Add Account** dialog box, enter the public cloud account information and select public cloud services.
 - Enter the **AccessKey ID** and **AccessKey** secret of the public cloud account.
 - Select the public cloud services to use. You can select **Anti-DDoS Pro**, **Web Application Firewall**, or both.
5. Click **Confirm**.

Result

After an account is added, it is displayed on the **Public Cloud Account** tab. To modify or delete an account, you can click **Modify** or **Delete** in the **Actions** column.

28.9. Optional security products

28.9.1. Anti-DDoS settings

28.9.1.1. Overview

In Distributed Denial of Service (DDoS) attacks, attackers exploit the client-server model to combine multiple computers into a platform that can launch attacks on one or more targets. This greatly increases the threat of attacks.

Common DDoS attack types include:

- **Network-layer attacks:** A typical example is UDP reflection attacks, such as NTP flood. These attacks use heavy traffic to congest the network of the victim, disabling proper responses to user requests.
- **Transport-layer attacks:** Typical examples include SYN flood and connection flood. These attacks consume a large number of connection resources of a server to cause denial of service.
- **Session-layer attacks:** A typical example is SSL flood. These attacks consume the SSL session resources of a server to cause denial of service.
- **Application-layer attacks:** Typical attack types include DNS flood, HTTP flood, and game zombie attacks. These attacks consume a large amount of application processing resources of a server to cause denial of service.

Apsara Stack Security can redirect, scrub, and re-inject attack traffic to protect your server against DDoS attacks and ensure normal business operations.

 **Note** Apsara Stack Security cannot scrub the traffic between internal networks.

28.9.1.2. View and configure anti-DDoS policies

This topic describes how to view and configure anti-DDoS policies. Anti-DDoS provides default anti-DDoS policies and distributed denial of service (DDoS) traffic scrubbing policies.

Context

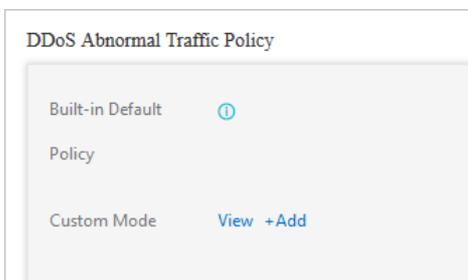
After an alert threshold of DDoS traffic is set for an IP address, an alert is triggered if the traffic to the IP address reaches the threshold. The alert thresholds for an IP address must be set based on the traffic volume. An abnormally large traffic volume indicates a possible DDoS attack. We recommend that you set an alert threshold to a value slightly higher than the peak traffic volume.

Apsara Stack Security supports global alert thresholds or alert thresholds for a specific Classless Inter-Domain Routing (CIDR) block or IP address.

- **Global alert threshold:** You cannot set a global alert threshold. It is set when the service is initialized.
- **Alert threshold for a specific CIDR block:** You can set an alert threshold for a specific CIDR block based on its traffic volume. Compared with global alert thresholds, CIDR block-specific alert thresholds allow you to control the traffic to each CIDR block.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > DDoS Defense Policy.**
3. **View and configure anti-DDoS policies.**



Option	Description
View default policies.	Move the pointer over the position indicated by 1 in the preceding figure to view the default anti-DDoS policies.
Customize an anti-DDoS policy.	Click View to view CIDR block-specific policies, and click Add to customize a policy for a CIDR block.

To customize a policy for a CIDR block, perform the following steps:

- i. Click **+Add** next to **Custom Mode**.
- ii. In the **Set Thresholds for Alerts** dialog box, configure parameters.

Parameter	Description
CIDR Block	The CIDR block for which the alert thresholds are used.
Bandwidth Threshold	The alert threshold for bandwidth usage in a data center. When the sum of inbound and outbound throughput reaches this threshold, DDoS detection is triggered. Set this parameter to a value slightly higher than the peak traffic rate. We recommend that you set the value to 100 or higher. Unit: Mbit/s.
Packets Threshold	The alert threshold for the packet rate of a data center. When the sum of the inbound and outbound packet rates reaches this threshold, DDoS detection is triggered. Set this parameter to a value slightly higher than the peak packet rate. We recommend that you set the value to 20,000 or higher. Unit: packets per second (PPS).

- iii. Click **OK**.

4. In the **DDoS Scrubbing Defense Strategy** section, click **View** to view DDoS traffic scrubbing policies.



28.9.1.3. View DDoS events

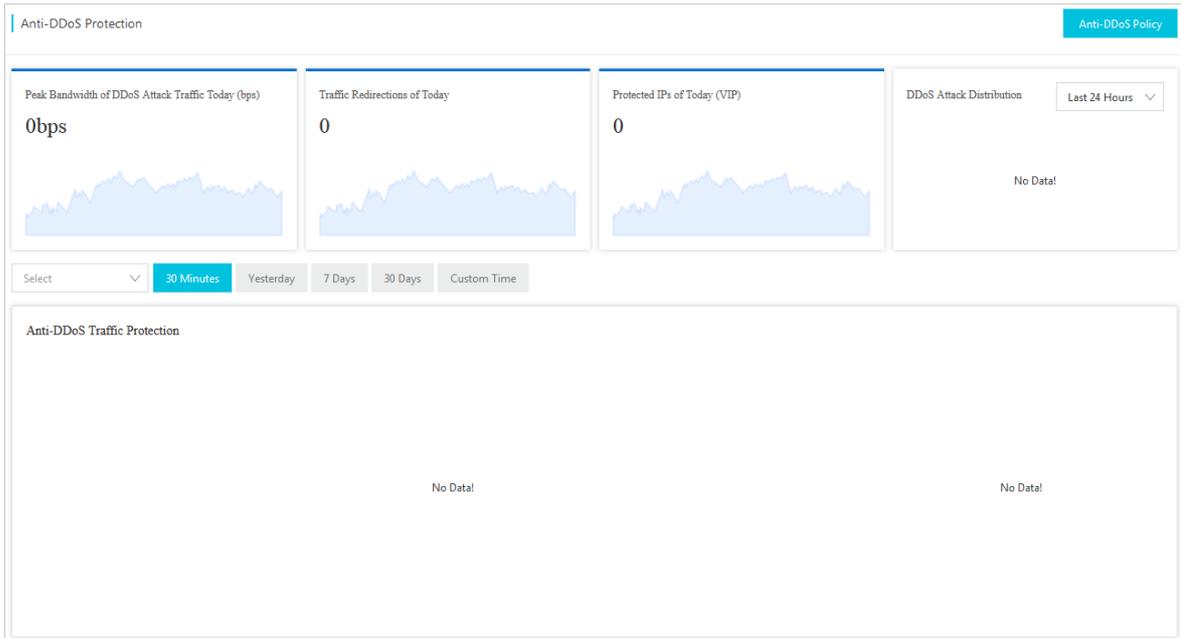
This topic describes how to view distributed denial of service (DDoS) events.

Context

During or after traffic scrubbing, Apsara Stack Security reports security events to Apsara Stack Security Center.

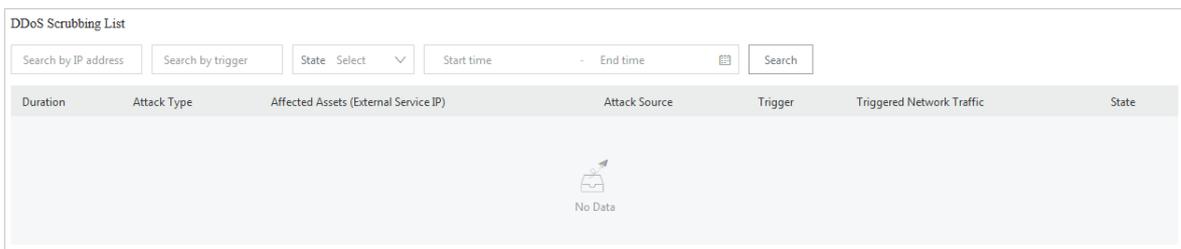
Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > DDoS Defense Policy**.
3. View anti-DDoS statistics.



4. (Optional) In the DDoS Scrubbing List section, specify search conditions and click Search.

Note Skip this step if you need to view all traffic scrubbing events.



Search condition	Description
Search by IP address	The IP address that was under a DDoS attack.
Search by trigger	The metric that exceeds the configured alert threshold in the DDoS attack traffic.
State	<ul style="list-style-type: none"> Scrubbing: indicates that traffic scrubbing is in progress. Scrubbing Complete: indicates that traffic scrubbing is complete.
Start time and End time	The start time and end time of DDoS traffic scrubbing.

5. In the DDoS Scrubbing List section, view details about DDoS traffic scrubbing events.

28.9.2. Cloud Firewall

28.9.2.1. Access control

28.9.2.1.1. Configure the Internet firewall switch policy

This topic describes how to enable and disable the Internet firewall switch policy for assets. You can manually synchronize assets and manage the protection status of target assets.

Manually synchronize assets

If new public IP addresses are not in the list, click Update Assets to update the public IP address list.

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Firewall Switch Policy.**
3. Click the **Internet Firewall** tab. In the upper-right corner of the page, click **Update Assets.**
4. In the **Update Assets** message, click **OK.**
After the assets are synchronized, the system automatically updates the public IP address list.

28.9.2.1.2. Create a VPC firewall

A virtual private cloud (VPC) firewall is a distributed firewall that can detect and control traffic between VPCs. Cloud Firewall can be used to analyze and control traffic between two VPCs only after a VPC firewall is created and enabled.

Context

A VPC firewall can be created only between two VPCs that are connected. VPCs can be connected by using Express Connect or Cloud Enterprise Network (CEN).

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Firewall Switch Policy.**
3. On the **Firewalls** page, click the **VPC-VPC** tab. Find the target VPC, In the **Actions** column, click **Create.**

Parameter	Description
Instance Name	Enter a name for the VPC firewall. We recommend that you enter a unique name that indicates the specific business to make it easy to identify the VPC firewall. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
Route Table	When you create a VPC, the system automatically creates a default route table. You can add system routes to the route table to manage VPC traffic. VPC allows you to create multiple route tables. When you create a VPC firewall in the Cloud Firewall console, Cloud Firewall automatically reads your VPC route tables. Express Connect supports multiple route tables. When you create a VPC firewall for an Express Connect, you can view multiple VPC route tables and can select the route tables that you want to protect.
Destination CIDR Block	After you select a route table from the Route Table drop-down list, the default destination Classless Inter-Domain Routing (CIDR) block of the route table is displayed in the Destination CIDR Block section. If you need to protect traffic to other CIDR blocks, you can manually modify destination CIDR blocks. You can add multiple CIDR blocks that are separated with commas (,).
Peer Route Table	Confirm the region and name of the peer VPC, and select the target route table.
Peer Destination CIDR Blocks	The destination CIDR block of the peer VPC.

Parameter	Description
Firewall Mode	<ul style="list-style-type: none"> ◦ Test: This mode is used to test the health status of the CIDR block or the IP address to ensure that the link is normal. ◦ Active: This mode is used to redirect and protect traffic. ◦ Bypass: The firewall does not redirect the traffic in this mode. If the self-test or the health test fails, the firewall is automatically changed to the Bypass mode. <p>When you create a VPC firewall, you can set Firewall Mode only to Test. After the firewall is created, you can change its mode to Active or Bypass. You cannot directly change the mode from Bypass to Active. You must change the mode from Bypass to Test first, and then to Active.</p> <p>In the Health Test section, enter the 32-bit test IP addresses that are created in the local and peer VPCs.</p>
IPS Mode	<p>Select the working mode of the intrusion prevention system (IPS). Valid values</p> <ul style="list-style-type: none"> ◦ Monitoring Mode: If you select this option, Cloud Firewall monitors traffic. In addition, Cloud Firewall sends alerts if it detects malicious traffic. ◦ Traffic Control Mode: If you select this option, Cloud Firewall intercepts malicious traffic and blocks intrusion attempts.
IPS Capabilities	<p>Select the intrusion prevention policies that you want to enable. Valid values:</p> <ul style="list-style-type: none"> ◦ Basic Policies: This feature provides basic intrusion prevention capabilities such as protection against brute-force attacks and attacks that exploit command execution vulnerabilities. It also allows you to manage and control the connections from infected hosts to a command and control (C&C) server. ◦ Virtual Patches: This feature defends against the most common high-risk application vulnerabilities in real time.
Enable VPC Firewall	<p>After you turn on Enable VPC Firewall, a VPC firewall is enabled automatically after it is created. If you do not require the VPC firewall to be automatically enabled after it is created, turn off this switch.</p>

4. Click **Submit**. When the VPC firewall takes effect, Firewall Status of the VPC firewall changes to **Enabled**.

28.9.2.1.3. Create an IDC-VPC firewall

An IDC-VPC firewall can detect the traffic between an on-premises data center and a virtual private cloud (VPC). Cloud Firewall allows you to control traffic through IDC-VPC firewalls. This topic describes how to create an IDC-VPC firewall.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Firewall Switch Policy**.
3. On the **Firewalls** page, click the **IDC-VPC** tab. In the upper-right corner, click **Create**.

Parameter	Description
Instance Name	<p>Enter a name for the VPC firewall. We recommend that you enter a unique name that indicates the specific business to make it easy to identify the VPC firewall.</p> <p>The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.</p>
VPC Instance	<p>Select the ID of a VPC. The ID is the unique identifier of the VPC. Cloud Firewall automatically synchronizes the VPCs that are connected to the on-premises data center.</p>
VPC Route Table	<p>When you create a VPC, the system automatically creates a default route table. You can add system routes to the route table to manage VPC traffic. VPC allows you to create multiple route tables.</p> <p>When you create a VPC firewall in the Cloud Firewall console, Cloud Firewall automatically reads your VPC route tables. Express Connect supports multiple route tables. When you create a VPC firewall for an Express Connect, you can view multiple VPC route tables and can select the route tables that you want to protect.</p>
VPC Destination CIDR Block	<p>After you select a route table from the VPC Route Table drop-down list, the default destination Classless Inter-Domain Routing (CIDR) block of the route table is displayed in the Destination CIDR Block section. If you need to protect traffic to other CIDR blocks, you can manually modify destination CIDR blocks.</p> <p>You can add multiple CIDR blocks that are separated with commas (,).</p>
IDC Express Connect Circuit (Primary)	<p>Select a leased line ID that you create when you connect the on-premises data center to Apsara Stack.</p> <p>When the customer edge (CE) in the on-premises data center connects to the primary and secondary VSwitches, you must specify a primary leased line. The IDC-VPC firewall automatically synchronizes the primary leased line that you specify. You must specify this parameter when you create the IDC-VPC firewall.</p>

Parameter	Description
VBR (Primary)	Select a virtual border router (VBR) that is bound to the primary leased line in the on-premises data center. The VBR facilitates communication between the VPC and the on-premises data center. The IDC-VPC firewall automatically synchronizes the VBR that you specify. You must specify this parameter when you create the IDC-VPC firewall.
IDC Express Connect Circuit (Secondary)	Select a leased line ID that you create when you connect the on-premises data center to Apsara Stack. You must specify a secondary leased line when the customer edge (CE) in the on-premises data center connects to the primary and secondary VSwitches. The value of this parameter cannot be the same as that of the IDC Express Connect Circuit (Primary) parameter. The IDC-VPC firewall automatically synchronizes the secondary leased line that you specify. You must specify this parameter when you create the IDC-VPC firewall.
VBR(Secondary)	Select a VBR that is bound to the secondary leased line in the on-premises data center. The VBR facilitates communication between the VPC and the on-premises data center. The IDC-VPC firewall automatically synchronizes the VBR that you specify. You must specify this parameter when you create the IDC-VPC firewall.
VPC Destination CIDR Block	The destination CIDR block of the peer VPC.
Firewall Mode	<ul style="list-style-type: none"> ◦ Test: This mode is used to test the health status of the CIDR block or the IP address to ensure that the link is normal. ◦ Active: This mode is used to redirect and protect traffic. ◦ Bypass: The firewall does not redirect the traffic in this mode. If the self-test or the health test fails, the firewall is automatically changed to the Bypass mode. <p>When you create a VPC firewall, you can set Firewall Mode only to Test. After the firewall is created, you can change its mode to Active or Bypass. You cannot directly change the mode from Bypass to Active. You must change the mode from Bypass to Test first and then to Active.</p> <p>In the Health Test section, enter a 32-bit test IP address that is created in the local VPC.</p>
IPS Mode	Select the working mode of the intrusion prevention system (IPS). Valid values: <ul style="list-style-type: none"> ◦ Monitoring Mode: If Cloud Firewall detects malicious traffic, it monitors traffic and sends alerts. ◦ Traffic Control Mode: Cloud Firewall intercepts malicious traffic and blocks intrusion attempts.
IPS Capabilities	Select the intrusion prevention policies that you want to enable. Valid values: <ul style="list-style-type: none"> ◦ Basic Policies: This feature provides basic intrusion prevention capabilities such as protection against brute-force attacks and attacks that exploit command execution vulnerabilities. It also allows you to manage and control the connections from infected hosts to a command and control (C&C) server. ◦ Virtual Patches: This feature defends against the most common high-risk application vulnerabilities in real time.
Enable VPC Firewall	After you turn on Enable VPC Firewall, an IDC-VPC firewall is enabled automatically after it is created. If you do not require the IDC-VPC firewall to be automatically enabled after it is created, turn off this switch.

4. Click **Submit**. When the IDC-VPC firewall takes effect, Firewall Status of the IDC-VPC firewall changes to **Enabled**.

28.9.2.1.4. Manage address books

This topic describes how to create and manage IP address books and port address books. You can use address books to store one or more Classless Inter-Domain Routing (CIDR) blocks or ports.

Context

You can store frequently used IP addresses and ports in address books to facilitate configurations of the Internet firewall.

Create an IP address book

1. Log on to [Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the Access Control page, click the **Internet Firewall** tab.
4. In the upper-right corner, click **Address Books**. In the dialog box that appears, click the **IP Address Books** tab.
5. Click **+Create Address Book** to specify parameters.

Parameter	Description
Address Book Type	Select the type of the address book. Set the value to IP Addresses .
Address Book Name	Specify the name of the address book. The name must be unique. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
IP Address	Enter a CIDR block. Separate multiple CIDR blocks with commas (,).
Description	Enter the content and scenarios of the address book. The description must be 2 to 512 characters in length.

6. Click **Submit**.
After the address book is created, you can view the address book on the **IP Address Books** tab. You can click **Modify** or **Delete** to manage address books.

Create a port address book

1. Log on to [Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the Access Control page, click the **Internet Firewall** tab.

4. In the upper-right corner, click **Address Books**. In the dialog box that appears, click the **Port Address Books** tab.
5. Click **+Create Address Book** to specify parameters.

Parameter	Description
Address Book Name	Specify the name of the address book. The name must be unique. The name can contain letters, digits, underscores (_), and hyphens (-). However, the name cannot contain only digits.
Ports	Enter a port number or port range. Separate multiple port numbers or ranges with commas (,).
Description	Enter the content and scenarios of the address book. The description must be 2 to 512 characters in length.

6. Click **Submit**.
After the address book is created, you can view the address book on the **Port Address Books** tab. You can click **Modify** or **Delete** to manage address books.

28.9.2.1.5. Configure an Internet firewall

The access control feature allows you to configure access control policies for Internet firewalls. You can configure inbound and outbound policies on the Internet Firewall tab to control the traffic between the Internet and your servers.

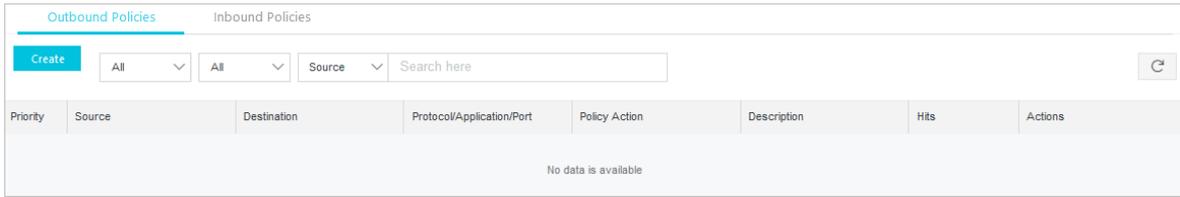
Context

An Internet firewall supports **Outbound Policies** and **Inbound Policies**.

In this topic, IP address books, port address books, and domain address books are used. For more information about how to create these address books, see [Manage address books](#).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internet Firewall** tab.
4. Click the **Outbound Policies** or **Inbound Policies** tab based on the direction of traffic that you want to control.



- **Outbound Policies tab:** You can configure control policies for traffic from your internal network to the Internet.
- **Inbound Policies tab:** You can configure control policies for traffic from the Internet to your internal network.

5. Click **Create**. In the **Create Outbound Policy** dialog box, configure the following parameters.

Parameter	Description
Source Type	<p>Select the type of the traffic source. Valid values: IP and Address Book.</p> <ul style="list-style-type: none"> ○ IP: indicates only one CIDR block. ○ Address Book: indicates a preset IP address book. An IP address book contains multiple CIDR blocks. This allows you to control traffic from multiple IP addresses.
Source	<p>Specify the source addresses that are allowed to access the Internet.</p> <ul style="list-style-type: none"> ○ If you set Source Type to IP, enter a CIDR block. Example: 1.1.1.1/32. ○ If you set Source Type to Address Book, click Select Address Book. In the Select Address Book as Source dialog box, select an IP address book.
Destination Type	<p>Select the type of the traffic destination. Valid values: IP, Address Book, Domain, and Region.</p>

Parameter	Description
Destination	<p>Specify the destination addresses that can be accessed. You must set Destination to addresses on the Internet.</p> <ul style="list-style-type: none"> ◦ If you set Destination Type to IP, enter a CIDR block. Example: 1.1.1.1/32. ◦ If you set Destination Type to Address Book, click Select Address Book. In the Select Address Book as Destination dialog box, select an IP address book. ◦ If you set Destination Type to Domain, enter a domain name. Example: www.example.com. ◦ If you set Destination Type to Region, select one or more destination regions from the drop-down list.
Protocol	<p>Select the protocol for outbound traffic. Valid values: TCP, UDP, ICMP, and ANY. If you are not sure which protocol is used, select ANY.</p>
Port Type	<p>Select the type of ports that are used for the selected protocol. Valid values: Ports and Address Book.</p> <ul style="list-style-type: none"> ◦ Ports: indicates a port range. ◦ Address Book: indicates a preset port address book. A port address book contains multiple ports. This allows you to control traffic on multiple ports.
Ports	<p>Specify the ports on which you want to control traffic. Enter a port number range or select a port address book, depending on the Port Type parameter.</p> <ul style="list-style-type: none"> ◦ If you set Port Type to Ports, enter a port range. ◦ If you set Port Type to Address Book, click Select Address Book. In the Select Ports dialog box, select a port address book.
Application	<p>Select the application to which the policy applies.</p>
Policy Action	<p>Specify whether the policy allows or denies traffic on the Internet firewall.</p> <ul style="list-style-type: none"> ◦ Allow: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is allowed. ◦ Monitor: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is recorded and allowed. ◦ Deny: If outbound traffic meets the preceding conditions that you specify for the policy, the traffic is denied.
Description	<p>Enter a description to identify the policy.</p> <p>The description can contain digits, letters, and underscores (_).</p>

6. Click **Submit**.

Result

After the policy is created, it appears in the policy list. In the **Actions** column that corresponds to the policy, you can click **Modify**, **Delete**, **Insert**, or **Move** to manage the policy.

28.9.2.1.6. Create a policy group

This topic describes how to create a policy group for an internal firewall. You can configure access control policies to restrict unauthorized access between ECS instances.

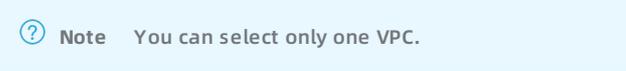
Context

An internal firewall is implemented by leveraging the security group module of ECS. The access control policies that you configure on the **Internal Firewall** tab are automatically synchronized to the security group module of

ECS.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internal Firewall** tab.
4. In the upper-right corner, click **Create Policy Group**.
5. In the **Create Policy Group** dialog box, configure the following parameters.

Parameter	Description	Configuration method
Name	The name of the policy group. The name must be 2 to 128 characters in length.	Enter a name to identify the policy group.
VPC	The VPC to which the policy group is applied.	Select a VPC from the VPC drop-down list. 
Instance ID	The ID of the ECS instance in the selected VPC.	Select an ECS instance ID from the Instance ID drop-down list. 
Description	The description of the policy group. The description must be 2 to 256 characters in length.	Enter a description to identify the policy group.
Template	The template of the policy group.	Select a template from the Template drop-down list. Valid values: <ul style="list-style-type: none"> ◦ default-accept-login: allows all inbound traffic on ports 22 and 3389. ◦ default-drop-all: denies all traffic in the policy group. ◦ default-accept-all: allows all traffic in the policy group.

6. Click **Submit**.

28.9.2.1.7. Manage an internal firewall

This topic describes how to view the policy groups of an internal firewall, configure access control policies in the policy groups, and synchronize the policies to the security group module of ECS.

Context

On the **Internal Firewall** tab, you can view custom policy groups and security groups that are synchronized from ECS.

For more information about how to create a custom policy group, see [Create a policy group](#).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. Click the **Internal Firewall** tab.
4. (Optional)Specify filter conditions, and click **Search** to search for the target policy group.

 **Note** If you want to view all policy groups, skip this step.

The search results are displayed in the policy group list.

5. Find the target policy group and configure its access control policy.

- i. In the **Actions** column, click **Configure Policy**.
- ii. On the **Policies** page, click **Create Policy**.

 **Note** If you want to modify or delete an access control policy, you can perform the following operations: Click the **Inbound** or **Outbound** tab, find the target policy, and then click **Modify** or **Delete**.

iii. In the **Create Policy** dialog box, configure the following parameters.

Parameter	Description	Configuration method
Network Type	The type of the network to which the policy is applied.	The default value is Internal , which indicates that the policy is applied to an internal network.
Direction	The direction of traffic that is controlled by the policy.	Valid values: <ul style="list-style-type: none"> ▪ Inbound: traffic from other ECS instances to the specified ECS instance. ▪ Outbound: traffic from the specified ECS instance to other ECS instances.
Policy Type	Indicates whether the policy allows or denies traffic on the internal firewall.	Valid values: <ul style="list-style-type: none"> ▪ Allow ▪ Deny
Protocol Type	The protocol of traffic that is controlled by the policy.	Select a protocol type from the Protocol Type drop-down list. Valid values: <ul style="list-style-type: none"> ▪ TCP ▪ UDP ▪ ICMP ▪ ANY: If you are not sure which protocol is used, select ANY.
Port Range	The destination port of traffic that is controlled by the policy.	Enter a port range. Example: 22/22.
Priority	The priority of the policy.	Enter a priority number. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note The priority number must be an integer from 1 to 100. Different policies can have the same priority. If an Allow policy and a Deny policy have the same priority, the Deny policy takes precedence. If two Allow policies have the same priority, both policies take effect.</p> </div>

Parameter	Description	Configuration method
Source Type	The type of the traffic source.	<p>Valid values:</p> <ul style="list-style-type: none"> ▪ CIDR Block: The traffic source is a CIDR block. ▪ Policy Group: The traffic source is any ECS instance in a policy group. <p> Note For a policy created in the security group module of ECS, you cannot set Source Type to Policy Group.</p>
Source	The source of the traffic.	<p>Enter a CIDR block or select a policy group, depending on the Source Type parameter.</p> <ul style="list-style-type: none"> ▪ If you set Source Type to CIDR Block, enter only one CIDR block. ▪ If you set Source Type to Policy Group, select a policy group from the Source drop-down list. In this case, the traffic source is any ECS instance in the selected policy group. <p> Note You can select only one policy group from the Source drop-down list.</p>
Destination	The destination of traffic that is controlled by the policy.	<p>Valid values:</p> <ul style="list-style-type: none"> ▪ All ECS Instances: The policy is applied to traffic destined for your ECS instances. ▪ CIDR Block: The policy is applied to traffic destined for the specified CIDR block.
Description	The description that is used to identify the policy.	Enter a description that is 2 to 256 characters in length.

iv. Click **Submit**.

6. Find the target policy group and click **Publish** in the **Actions** column to apply the policy and synchronize it to the security group module of ECS.

28.9.2.1.8. Manage a VPC firewall

A VPC firewall detects and controls the traffic between two VPCs. Cloud Firewall allows you to configure access control policies for VPC firewalls. This topic describes how to configure access control policies for VPC firewalls.

Prerequisites

VPC firewalls are not automatically created. Before you configure access control policies for VPCs, you must create and enable a VPC firewall.

Access control policies take effect only after you enable the VPC firewall.

Context

A VPC firewall allows all traffic by default. If you want to control traffic between VPCs, you can configure access control policies to deny traffic from untrusted sources. You can also allow traffic from trusted sources and deny traffic from all other sources.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Network Security > Access Control.
3. Click the VPC Firewall tab.
4. Click Create.
5. In the Create VPC Firewall Policy dialog box, configure the following parameters.

Create VPC Firewall Policy
✕

Source Type: IP Address Book

* Source: ⓘ

Destination Type: IP Address Book Domain Name

* Destination: ⓘ

* Protocol: ▾

Port Type: Ports Address Book

* Ports: ⓘ

* Application: ▾

* Policy Action: ▾

* Description:

Submit
Cancel

Parameter	Description
Source Type	<p>The type of the traffic source. Valid values: IP and Address Book.</p> <ul style="list-style-type: none"> ◦ If you set Source Type to IP, enter a CIDR block. ◦ If you set Source Type to Address Book, select a preset IP address book. <p>You can add multiple CIDR blocks to an address book to simplify policy configuration.</p>
Source	<p>The source of the traffic.</p> <div style="background-color: #e1f5fe; padding: 5px; border: 1px solid #cfe2f3; margin: 5px 0;"> ? Note If you set Source Type to IP, enter a CIDR block. Example: 1.1.1.1/32. </div> <p>If you set Source Type to Address Book, select a preset address book.</p>

Parameter	Description
Destination Type	<ul style="list-style-type: none"> ◦ IP: Set the destination to a CIDR block. ◦ Address Book: Set the destination to an address book. ◦ Domain Name: Set the destination to a domain name. Wildcard domain names are supported. Example: <i>*.aliyun.com</i>. <p> Note By default, if an HTTP header does not contain the Host field or an HTTPS request does not contain the Server Name Indication (SNI), Cloud Firewall allows the traffic.</p>
Destination	<p>The destination of the traffic. You can enter only one CIDR block.</p> <p>If you set Destination Type to Domain Name, enter a domain name. Wildcard domain names are supported. Example: <i>*.aliyun.com</i>.</p>
Protocol Type	<ul style="list-style-type: none"> ◦ ANY ◦ TCP ◦ UDP ◦ ICMP
Ports	<p>You can specify a port range. 0/0 indicates any port.</p> <p> Note If you set Protocol to ICMP, the port configuration does not take effect. If you set Protocol to ANY, the port configuration does not take effect in ICMP traffic control.</p>
Application	<p>Valid values: ANY, HTTP, HTTPS, Memcache, MongoDB, MQTT, MySQL, RDP, Redis, SMTP, SMTPS, SSH, and VNC.</p> <p>If you set Protocol to TCP, multiple applications are available. If you set Protocol to a value other than TCP, only ANY is available.</p> <p> Note Cloud Firewall identifies applications based on packet characteristics, instead of port numbers. If Cloud Firewall fails to identify the application in a packet, it allows the packet.</p>
Policy Action	<p>Specify whether the policy allows or denies traffic on the VPC firewall.</p> <ul style="list-style-type: none"> ◦ Allow: If traffic meets the preceding conditions that you specify for the policy, the traffic is allowed. ◦ Deny: If traffic meets the preceding conditions that you specify for the policy, the traffic is denied. ◦ Monitor: If traffic meets the preceding conditions that you specify for the policy, the traffic is recorded and allowed. After you observe the traffic for a period of time, you can change the policy action to Allow or Deny.
Description	<p>Enter a description to identify the policy.</p>

6. Click **Submit**.

28.9.2.1.9. Manage an IDC-VPC firewall

An IDC-VPC firewall can detect the traffic between an on-premises data center and a virtual private cloud (VPC). Cloud Firewall allows you to control accesses by using IDC-VPC firewalls. This topic describes access control that is implemented by using IDC-VPC firewalls.

Prerequisites

IDC-VPC firewalls are not automatically created. Before you configure access control policies between an on-premises data center and a VPC, you must create and enable an IDC-VPC firewall.

Access control policies take effect only after you enable the IDC-VPC firewall.

Context

An IDC-VPC firewall allows all traffic by default. If you want to control traffic between an on-premises data center and a VPC, you can configure access control policies to deny traffic from untrusted sources. You can also allow traffic from trusted sources and deny traffic from all other sources.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Network Security > Access Control**.
3. On the page that appears, click the **IDC-VPC Firewall** tab.
4. Click **Create**.
5. In the **Create VPC Firewall Policy** dialog box, configure a policy.

Create VPC Firewall Policy ✕

Source Type: IP Address Book

* Source: ⓘ

Destination Type: IP Address Book Domain Name

* Destination: ⓘ

* Protocol: ▾

Port Type: Ports Address Book

* Ports: ⓘ

* Application: ▾

* Policy Action: ▾

* Description:

Parameter	Description
-----------	-------------

Parameter	Description
Source Type	<p>The type of the traffic source. Valid values: IP and Address Book.</p> <ul style="list-style-type: none"> ◦ If you select IP, enter a Classless Inter-Domain Routing (CIDR) block in the Source field. ◦ If you select Address Book, click Select Address Book next to the Source field and select a pre-configured address book. <p>You can add more than one IP address to an address book to simplify policy configuration.</p>
Source	<p>The source of the traffic.</p> <p> Note If you set Source Type to IP, enter a CIDR block. Example: 1.1.1.1/32.</p> <p>If you set Source Type to Address Book, select a pre-configured address book.</p>
Destination Type	<ul style="list-style-type: none"> ◦ IP: Set the destination to a CIDR block. ◦ Address Book: Set the destination to an address book. ◦ Domain Name: Set the destination to a domain name. Wildcard domain names are supported. Example: <i>*.aliyun.com</i>. <p> Note By default, if an HTTP header does not contain the Host field or an HTTPS request does not contain the Server Name Indication (SNI), Cloud Firewall allows the traffic.</p>
Destination	<p>The destination of the traffic. You can enter only one CIDR block.</p> <p>If you set Destination Type to Domain Name, enter a domain name. Wildcard domain names are supported. Example: <i>*.aliyun.com</i>.</p>
Protocol	<ul style="list-style-type: none"> ◦ ANY: any protocol ◦ TCP ◦ UDP ◦ ICMP
Ports	<p>You can specify a port range. 0/0 indicates all ports can be matched.</p> <p> Note If you set Protocol to ICMP, the destination ports do not take effect. If you set Protocol to ANY, the port configuration does not take effect in ICMP throttling.</p>
Application	<p>Valid values:</p> <p>ANY, HTTP, HTTPS, Memcache, MongoDB, MQTT, MySQL, RDP, Redis, SMTP, SMTPS, SSH, SSL, and VNC.</p> <p>If you set Protocol to TCP, multiple applications are available. If you set Protocol to a value other than TCP, only ANY is available.</p> <p> Note Cloud Firewall identifies applications based on packet characteristics, instead of port numbers. If Cloud Firewall fails to identify the application in a packet, it allows the packet.</p>

Parameter	Description
Policy Action	Specify whether the IDC-VPC firewall allows or denies traffic. <ul style="list-style-type: none"> ○ Allow: Matched traffic is allowed. ○ Deny: Matched traffic is denied, and no notifications are sent. ○ Monitor: Matched traffic is monitored but allowed. After you observe the traffic for a period of time, you can change the policy action to Allow or Deny.
Description	The description of the policy that helps you identify the policy.

6. Click **Submit**.

28.9.2.2. Intrusion prevention

28.9.2.2.1. Configure intrusion prevention policies

Cloud Firewall uses a built-in threat detection engine to defend against intrusions and common cyber attacks. It provides virtual patches against vulnerabilities to intelligently block intrusion attempts.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Network Security > Policy Configuration > Intrusion Prevention Policies**.
3. Select a running mode for the threat engine.



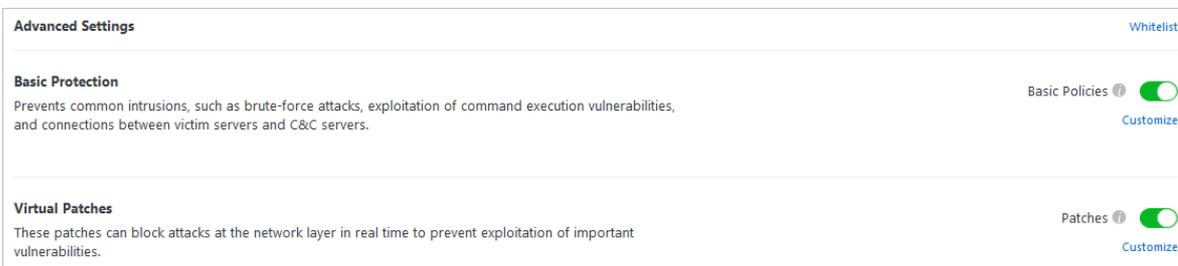
Valid values: **Monitoring Mode** and **Traffic Control Mode**.

- **Monitoring Mode:** In monitoring mode, the system generates alerts on intrusions instead of blocking the malicious traffic.
- **Traffic Control Mode:** In traffic control mode, the system automatically blocks malicious traffic.

To configure **Traffic Control Mode**, perform the following steps:

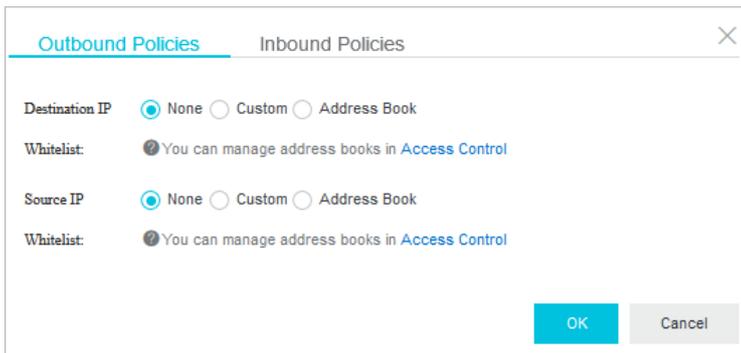
- i. **Select Traffic Control Mode.**
- ii. In the dialog box that appears, click **OK**.

4. Configure the advanced features.

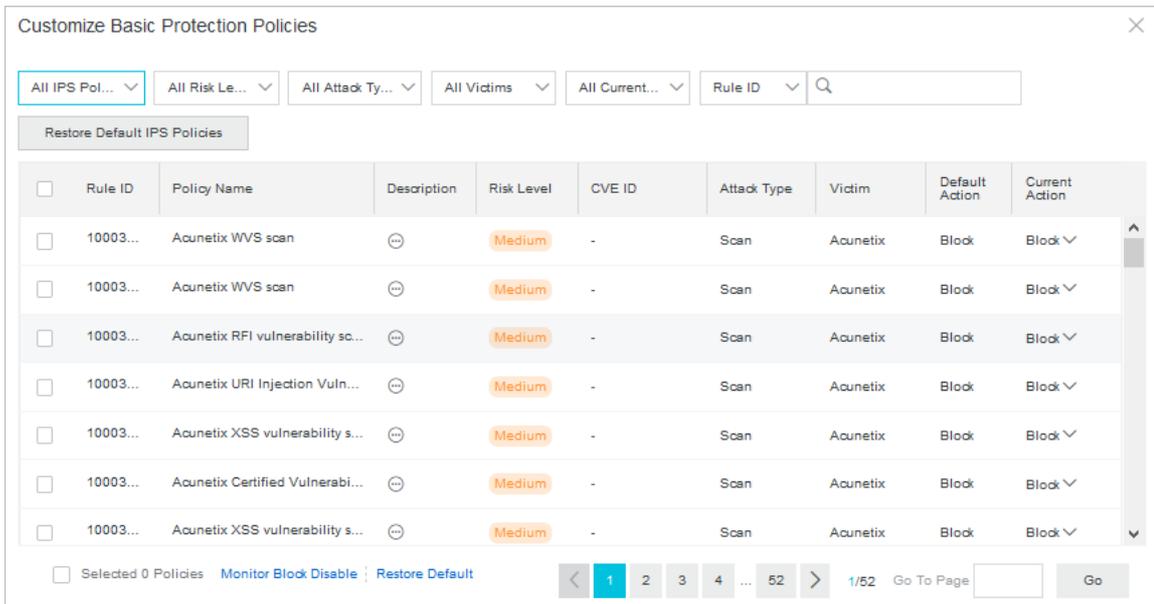


- **Cloud Firewall allows traffic from IP addresses in the whitelist.**
In the **Advanced Settings** section, click **Whitelist** to add trusted IP addresses to a whitelist.

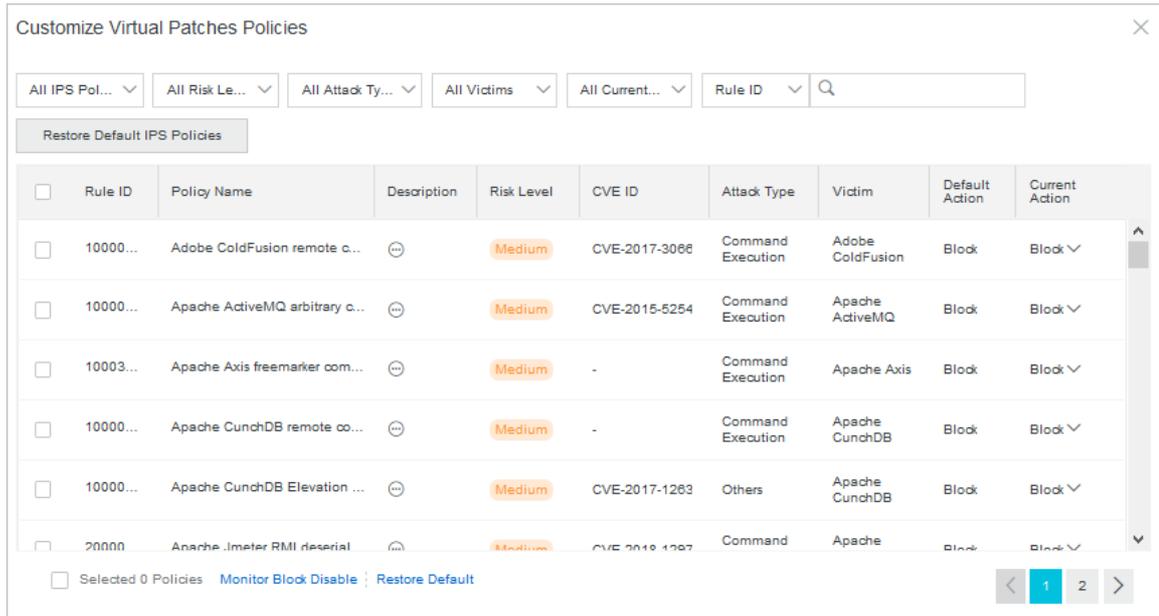
You can add the trusted source IP addresses, destination IP addresses, or address books of both inbound and outbound traffic to the whitelist.



- The basic protection feature defends your network against common intrusions, such as brute-force attacks and command execution vulnerabilities. The basic protection feature also manages connections from infected hosts to a command-and-control (C&C) server.
 - a. Click **Basic Policies** to enable the basic protection feature.
 - b. In the **Basic Protection** section, click **Customize**. In the **Customize Basic Protection Policies** dialog box, you can customize one or more basic protection policies.



- Virtual patches are installation-free. You can use them to defend against high-risk vulnerabilities.
 - a. Click **Patches** to enable the virtual patch feature.
 - b. In the **Virtual Patches** section, click **Customize**. In the **Customize Virtual Patches Policies** dialog box, configure one or more basic virtual patch policies.



28.9.2.2.2. View event logs

All traffic passing through Cloud Firewall is recorded on the Log Audit page. The logs are classified into traffic logs and event logs. You can use the logs to audit your network traffic in real time and take actions accordingly. This topic describes how to view event logs.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Network Security > Log Audit > Event Log.
3. On the Event Logs tab, click Internet Firewall, VPC-VPC Firewall or IDC-VPC Firewall.
4. Specify the search conditions and click Search.

Note If you want to view all event logs, skip this step.

Search condition	Description
Source IP	Specify the source IP address of the event.
Destination IP	Specify the destination IP address of the event.
Type	Select the event type
Action	Select the event action. Valid values: <i>All</i> , <i>Monitor</i> , and <i>Discard</i> .
Time	Set the time range.

5. The event logs record the information of event. This includes the event detection time, threat type, traffic direction (inbound or outbound), source IP address, destination IP address, application type, severity, and policy action.

28.9.2.3. Log analysis

28.9.2.3.1. View traffic logs

All traffic passing through Cloud Firewall is recorded on the Log Audit page. The logs are classified into traffic logs and event logs. You can use the logs to audit your network traffic in real time and take actions accordingly. This topic describes how to view traffic logs.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose **Network Security > Log Audit > Traffic Log**.
3. On the **Event Logs** tab, click **Internet Firewall, VPC-VPC Firewall** or **IDC-VPC Firewall**.
4. Specify the search condition and click **Search**.

Note If you want to view all traffic logs, skip this step.

Search condition	Description
Source IP	Specify the source IP address of the traffic.
Destination IP	Specify the destination IP address of the traffic.
Application	Select the application type.
Time	Set the time range.

To enable the advanced features, perform the following steps:

- **Show Advanced Search**
Click **Show Advanced Search** to configure more search conditions.
 - **List Configuration**
Click **List Configuration** to select items for the traffic list.
5. The traffic logs record the information of access traffic. This includes the start time and end time of the access traffic, traffic direction (inbound or outbound), source IP address, destination IP address, application type, supported protocol, bytes, and packets.

28.9.3. Sensitive Data Discovery and Protection

28.9.3.1. Grant access permissions

This topic describes how to authorize Sensitive Data Discovery and Protection (SDDP) to access data of your department before you use SDDP.

Prerequisites

The department name and AccessKey pair are obtained. For more information, see [Obtain the AccessKey pair of an organization in ASCM Console User Guide](#). To find the topic, choose **Enterprise > Organizations > Obtain the AccessKey pair of an organization**.

Context

Before you use SDDP, you must perform the following operations:

- Authorize SDDP to access the data of your department.
- Authorize SDDP to access data of Apsara Stack services of your department. These services include MaxCompute, Object Storage Service (OSS), and Tablestore.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Data Security > Sensitive Data Protection. The Authorization page appears.

Note If SDDP is not authorized to access the data of your department, configure parameters on the Authorization page to authorize SDDP.

Authorization

Add Authorization

* Department * Department AccessKey ID * Department AccessKey Secret

Authorized Account Information

Department	Department Alibaba Cloud Account	Display Name	Authorization Time
	dtdep-1:		Nov 1, 2019, 10:21:47

Total: 1 < Previous **1** Next >

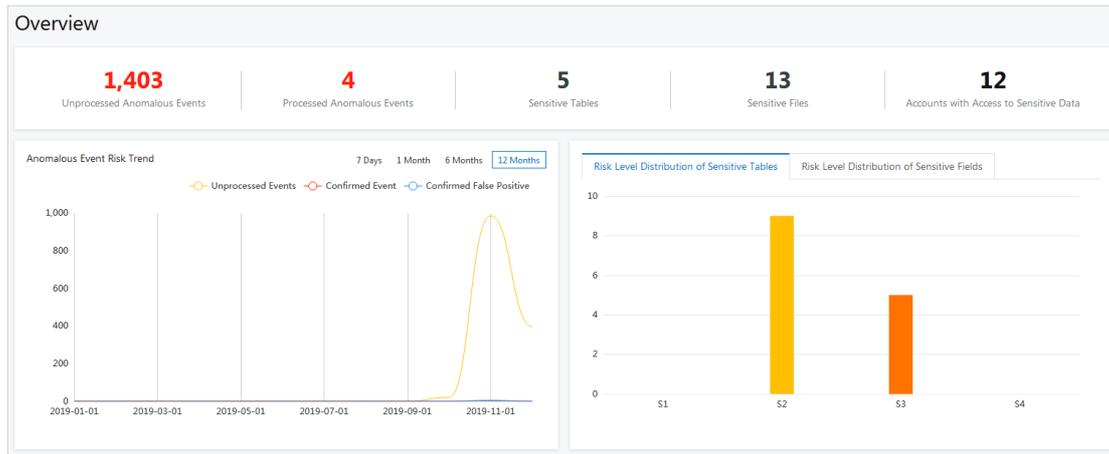
3. In the Add Authorization section, authorize SDDP to access data of your department.
 - i. In the Department drop-down list, enter a keyword and select the department.
 - ii. Specify Department AccessKey ID and Department AccessKey Secret.
 - iii. Click Submit.
4. In the Authorized Account Information section, view the departments that SDDP is authorized to access.

28.9.3.2. Overview

This topic describes the Overview page of Sensitive Data Discovery and Protection (SDDP). This page displays the overall security status of data protected by SDDP, and allows a security administrator to understand the current security status of sensitive data.

SDDP can detect sensitive data in your data assets based on sensitive data detection rules and track the use of sensitive data. SDDP also provides a data overview for you to obtain the security status of your data assets in real time.

In the left-side navigation pane, choose Data Security > Sensitive Data Discovery and Protection. In the left-side navigation tree, click Overview. On the Overview page, view the overall security status of the sensitive data.



- **Overview:** displays the overall information of sensitive data. This includes the number of unprocessed anomalous events, the number of anomalous events confirmed as violations, the total number of sensitive tables, the total number of sensitive objects, and accounts that accessed sensitive data.
- **Abnormal Event risk Trend:** displays the trends of anomalous events in a line chart. You can select 7 Days, 1 Month, 6 Months, or 12 Months to view the trends of Unprocessed Events, Confirmed Events, and Confirmed False Positive.
- **Risk Level Distribution of Sensitive Tables:** displays the distribution of sensitive tables at the S1, S2, and S3 risk levels.
- **Risk Level Distribution of Sensitive Fields:** displays the distribution of sensitive fields at the S1, S2, and S3 risk levels.
- **Data Flow Status:**
 - Displays the dynamic statistics on core data flows in DataHub and Cloud Data Pipeline (CDP).
 - Provides a data flowchart that dynamically displays the data flow status and abnormal output. You can click an anomalous event in the flowchart to go to the **Abnormal data flow** page.

Monitors the data links among entities such as data storage services MaxCompute, , Object Storage Service (OSS), and Tablestore, data transmission services DataHub and CDP, the data flow processing service Blink, external databases, and external files.

28.9.3.3. Detect sensitive data

28.9.3.3.1. Sensitive data overview

This topic describes how to view the overall security status of your data assets.

Choose **Data Security > Sensitive Data Discovery and Protection > Sensitive data identification > Sensitive data overview**. On the **Sensitive data overview** page, you can view the overall security status of your data assets.

- You can view the overall information about sensitive data. The information includes the total numbers of tables, objects, sensitive instances, sensitive tables, and sensitive objects.
- You can search for sensitive data based on conditions such as the risk level, asset type, sensitive data type, and asset name.
- You can view the statistics on the authorization information and sensitive data in Apsara Stack services such as MaxCompute, Object Storage Service (OSS), and Table Store in real time.

28.9.3.3.2. View statistics on sensitive data in MaxCompute

This topic describes how to view statistics on sensitive data in MaxCompute.

Context

MaxCompute is a rapid and fully-managed data warehouse solution that can process terabytes or petabytes of data. MaxCompute provides you with complete data import schemes and various classic distributed computing models. It supports fast computing on a large amount of data, effectively saves costs for enterprises, and guarantees data security.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Data Security > Sensitive Data Protection > Sensitive Data Identification > MaxCompute.
3. View statistics on sensitive data in MaxCompute.



- i. In the Sensitive Data Statistics section, enter a keyword and select the target MaxCompute project from the drop-down list.

Note To view statistics on all MaxCompute projects, select All from the drop-down list.

- ii. On the Proportion of Sensitive Tables and Proportion of Sensitive Fields tabs, view the proportions of sensitive and non-sensitive tables and fields.
 - iii. On the Risk Level Distribution of Sensitive Tables and Risk Level Distribution of Sensitive Fields tabs, view the distribution of sensitive tables and fields at the S1, S2, S3, and S4 risk levels.
4. Query sensitive data in MaxCompute.

By default, the system displays all MaxCompute projects. Different risk levels are indicated by different colors. To view information about a specific project, package, table, or field, follow these steps:

- i. Select a risk level from the Risk Level drop-down list.
- ii. Enter a keyword of the project, package, or table in the search box.
- iii. Click **Project Search**, **Package Search**, or **Table Search**. You can view the relationships among the projects, packages, tables, and fields, and the related authorization information in a tree map.
 - The tree map displays the distribution of sensitive data in MaxCompute.
 - You can view the authorization information of a project, package, table, or field. The system displays the authorization information by category, including the authorized users and violations.
 - You can click Package management under a project to view packages in the project, including the tables and fields in the packages and related authorization information.
- iv. Move the pointer over the project, package, or table to view details.

28.9.3.3.3. View statistics on sensitive data in Tablestore

This topic describes how to view statistics on sensitive data in Tablestore.

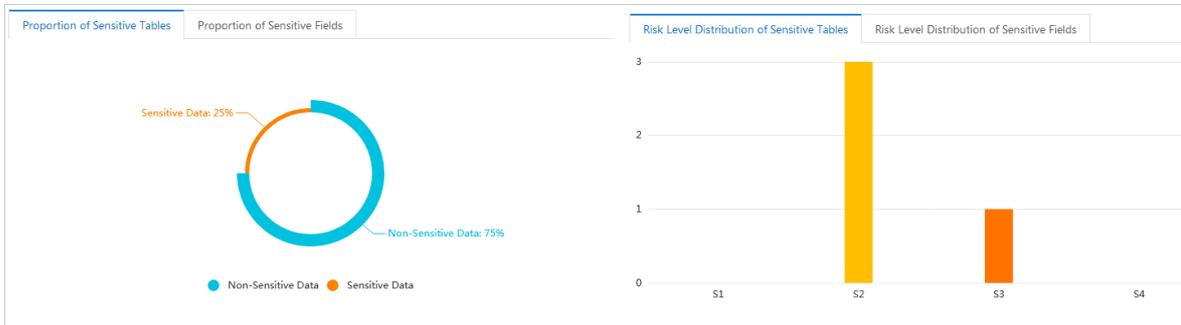
Context

Tablestore is a multi-model NoSQL database service provided by Apsara Stack. It can store a large amount of structured data and supports fast query and analysis. The distributed storage and powerful index-based search engine allow Tablestore to store petabytes of data while ensuring a 10 million TPS and a latency within milliseconds.

Procedure

1. [Log on to Apsara Stack Security Center.](#)

2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > Table Store**.
3. View statistics on sensitive data in Tablestore.



- i. In the **Sensitive Data Statistics** section, enter a keyword and select the target Tablestore instance from the drop-down list.

Note To view statistics on all Tablestore instances, select **All** from the drop-down list.

- ii. On the **Proportion of Sensitive Tables** and **Proportion of Sensitive Fields** tabs, view the proportions of sensitive and non-sensitive tables and fields.
 - iii. On the **Risk Level Distribution of Sensitive Tables** and **Risk Level Distribution of Sensitive Fields** tabs, view the distribution of sensitive tables and fields at the S1, S2, S3, and S4 risk levels.
4. Query sensitive data in Tablestore.

Sensitive Data Search

Risk Level: ▼ Enter a resource name. Fuzzy search is supported. Instance Search Table Search

- S1 LowRisk Level
- S2 MediumRisk Level
- S3 HighRisk Level
- S4 HighestRisk Level

OTS

- msg-bk Authorization Information
- ictx-prod Authorization Information
- databus Authorization Information
- ictx-ots-test Authorization Information
- demo-7cq6 Authorization Information
- test-ots Authorization Information
- S2 yundun-ots2 Authorization Information
- qianyun-ceshi Authorization Information
- S3 yundun-ots1 Authorization Information
- bbbb Authorization Information

By default, the system displays all Tablestore instances. To view information about a specific instance or table, perform the following steps:

- i. Select a risk level from the **Risk Level** drop-down list.
- ii. Enter a keyword of the instance or table in the search field.
- iii. Click **Instance Search** or **Table Search**.
 - The tree map displays the distribution of sensitive data in Tablestore.
 - You can view the authorization information of an instance or table. The system displays the authorization information by category, including the authorized users and violations.
- iv. Move the pointer over the instance or table to view details.

28.9.3.3.4. View statistics on sensitive data in OSS

This topic describes how to view statistics on sensitive data in Object Storage Service (OSS).

Context

OSS is a secure and reliable cloud storage service provided by Apsara Stack. It can store a large amount of data at low costs. OSS can store all types of files and is suitable for various websites, enterprises, and developers.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > OSS**.
3. View statistics on sensitive data in OSS.



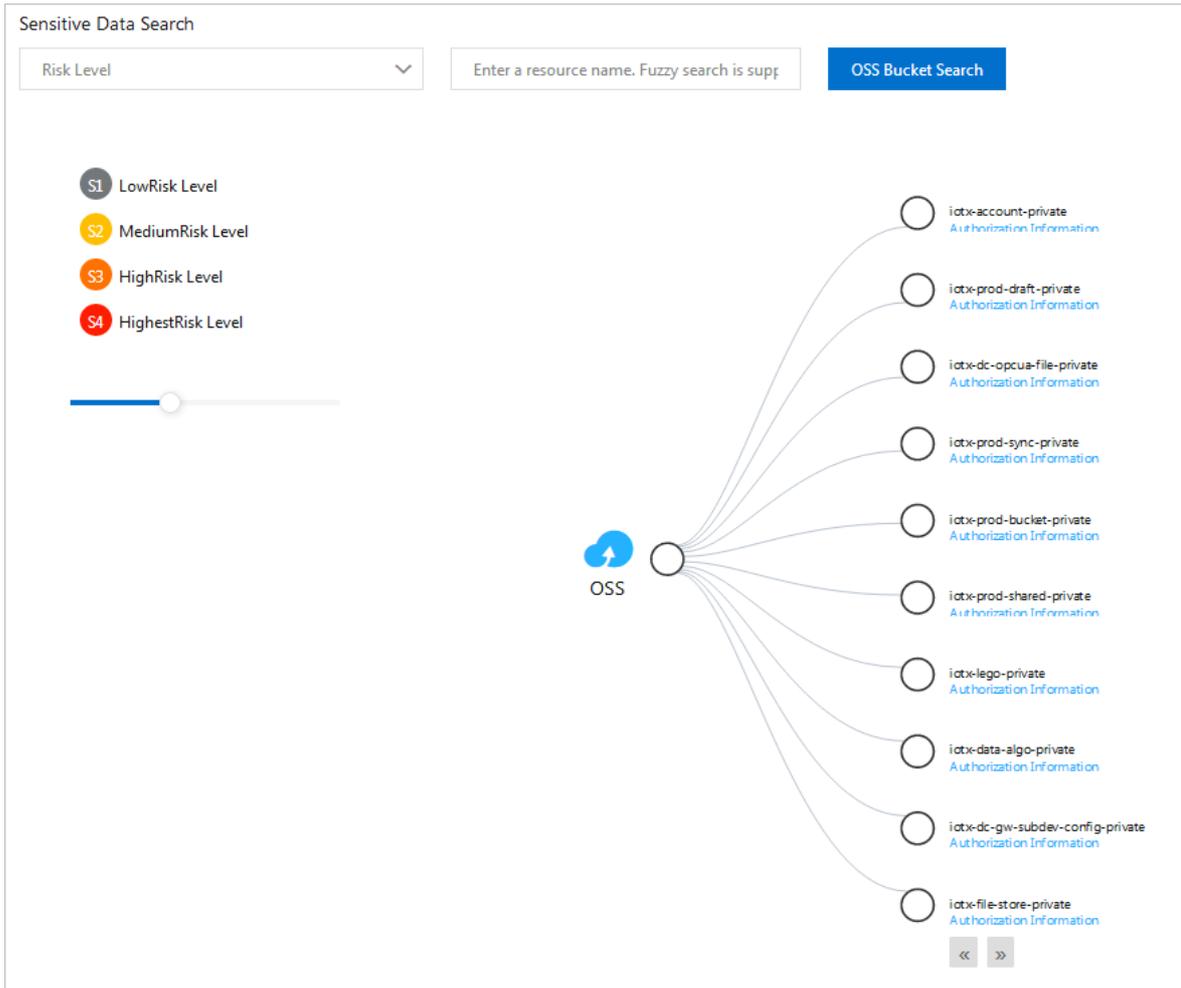
View the charts on the **Proportion of Sensitive File Objects** and **Risk Level Distribution of Sensitive Objects** tabs.

To view the charts for a specific OSS bucket, follow these steps:

- i. In the **Sensitive Data Statistics** section, enter a keyword and select the target OSS bucket from the drop-down list.

Note To view statistics on all OSS buckets, select *All* from the drop-down list.

- ii. On the **Proportion of Sensitive File Objects** tab, view the proportions of sensitive and non-sensitive objects.
 - iii. On the **Risk Level Distribution of Sensitive Objects** tab, view the distribution of sensitive objects at the S1, S2, S3, and S4 risk levels.
4. Query OSS sensitive data.



By default, the system displays all OSS buckets. To view information about a specific bucket, follow these steps:

- i. Select a risk level from the **Risk Level** drop-down list.
- ii. Enter a keyword of the bucket in the search field and click **OSS Bucket Search**.
 - The tree map displays the distribution of sensitive data in OSS.
 - You can view the authorization information of a bucket. The system displays the authorization information by category, including the authorized users and violations.
- iii. Move the pointer over the bucket to view details.

28.9.3.3.5. View statistics on sensitive data in AnalyticDB

This topic describes how to view statistics on sensitive data in AnalyticDB.

Context

AnalyticDB is an Alibaba-developed real-time online analytical processing (OLAP) service that can process a massive amount of data at high concurrency. It can analyze hundreds of billions of data records from multiple dimensions within milliseconds. This provides data-driven insights into your business. AnalyticDB can compute a large amount of data and quickly respond to requests. It enables you to instantly and agilely explore and find data values. You can also embed AnalyticDB into your business system to provide end users with analytics services.

Procedure

1. **Log on to Apsara Stack Security Center.**

2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > ADS**.
3. View statistics on sensitive data in AnalyticDB.

On the **Proportion of Sensitive Tables**, **Proportion of Sensitive Fields**, **Risk Level Distribution of Sensitive Tables**, and **Risk Level Distribution of Sensitive Fields** tabs, view the statistics.

To view the charts for a specific database, follow these steps:

- i. In the **Sensitive Data Statistics** section, enter a keyword and select the target database from the drop-down list.

 **Note** To view statistics on all databases, select *All* from the drop-down list.

- ii. On the **Proportion of Sensitive Tables** and **Proportion of Sensitive Fields** tabs, view the proportions of sensitive and non-sensitive tables and fields.
 - iii. On the **Risk Level Distribution of Sensitive Tables** and **Risk Level Distribution of Sensitive Fields** tabs, view the distribution of sensitive tables and fields at the S1, S2, S3, and S4 risk levels.
4. Query sensitive data in AnalyticDB. All AnalyticDB databases are displayed by default. To view information about a specific database or table, follow these steps:
 - i. Select a risk level from the **Risk Level** drop-down list.
 - ii. Enter a keyword of the database or table in the search box.
 - iii. Click **Database Search** or **Table Search**.
 - The tree map displays the distribution of sensitive data in AnalyticDB.
 - You can view the authorization information of a database or table. The system displays the authorization information by category, including authorized users and violations.
 - iv. Move the pointer over the database or table to view details.

28.9.3.3.6. View statistics on sensitive data in ApsaraDB for RDS

This topic describes how to view statistics on sensitive data in ApsaraDB for RDS.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Sensitive Data Identification > RDS**.
3. View statistics on sensitive data in ApsaraDB for RDS.



On the **Proportion of Sensitive Tables**, **Proportion of Sensitive Fields**, **Risk Level Distribution of Sensitive Tables**, and **Risk Level Distribution of Sensitive Fields** tabs, view the statistics.

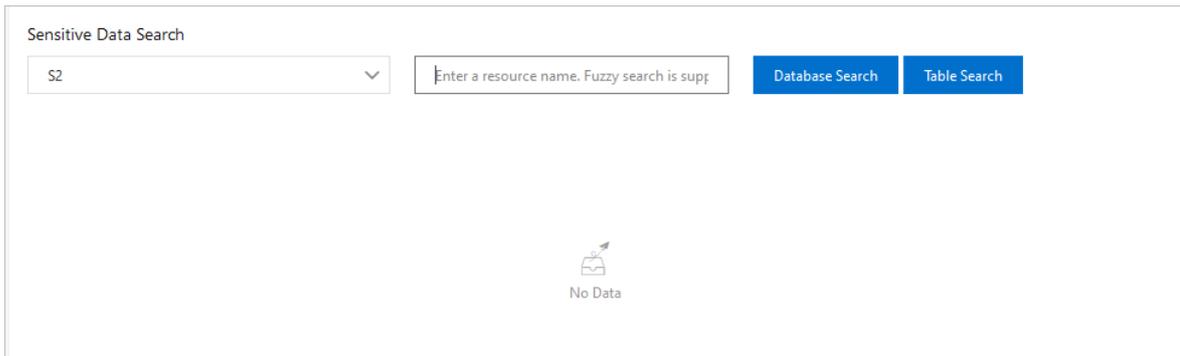
To view the charts for a specific database, follow these steps:

- i. In the **Sensitive Data Statistics** section, enter a keyword and select the target database from the drop-down list.

 **Note** To view statistics on all databases, select *All* from the drop-down list.

- ii. On the **Proportion of Sensitive Tables** and **Proportion of Sensitive Fields** tabs, view the proportions of sensitive and non-sensitive tables and fields.
- iii. On the **Risk Level Distribution of Sensitive Tables** and **Risk Level Distribution of Sensitive Fields** tabs, view the distribution of sensitive tables and fields at the S1, S2, S3, and S4 risk levels.

4. Query sensitive data in ApsaraDB for RDS.



All ApsaraDB for RDS databases are displayed by default. To view information about a specific database or table, follow these steps:

- i. Select a risk level from the **Risk Level** drop-down list.
- ii. Enter a keyword of the database or table in the search box.
- iii. Click **Database Search** or **Table Search**.
 - The tree map displays the distribution of sensitive data in ApsaraDB for RDS.
 - You can view the authorization information of a database or table. The system displays the authorization information by category, including authorized users and violations.
- iv. Move the pointer over the database or table to view details.

28.9.3.4. Check data permissions

28.9.3.4.1. View permission statistics

This topic describes how to view permission statistics.

Context

On the **Permission Management** page, you can view the overall permission distribution of Apsara Stack. You can also quickly identify high-risk accounts and users, and troubleshoot and resolve security issues in a timely manner.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data Permissions > Permission Management**.
3. View the overall permission statistics.



- **Accounts with Access to Sensitive Data:** the number of accounts that can access sensitive data.
- **Departments:** the number of departments in Apsara Stack.
- **Users:** the number of users in Apsara Stack.
- **Accounts:** the number of accounts in Apsara Stack.

4. View the department-level permission statistics.

Department Name	Users	Apsara Stack Console Accounts	Accounts	RAM Users	Permission Anomalous Events	Risk-Confirmed Permission Anomalous Events	Permission Anomalous Events of Yesterday	Permission Anomalous Type with Most Confirmed Violations
[Redacted]	2	2	1	0	0	0	0	--
[Redacted]	1	3	1	0	0	0	0	--
[Redacted]	1	1	1	0	0	0	0	--
[Redacted]	2	2	1	0	3	0	0	Login time is abnormal

You can view the statistics on the users, accounts, and anomalous activities related to permissions for each department.

28.9.3.4.2. View permissions of an account

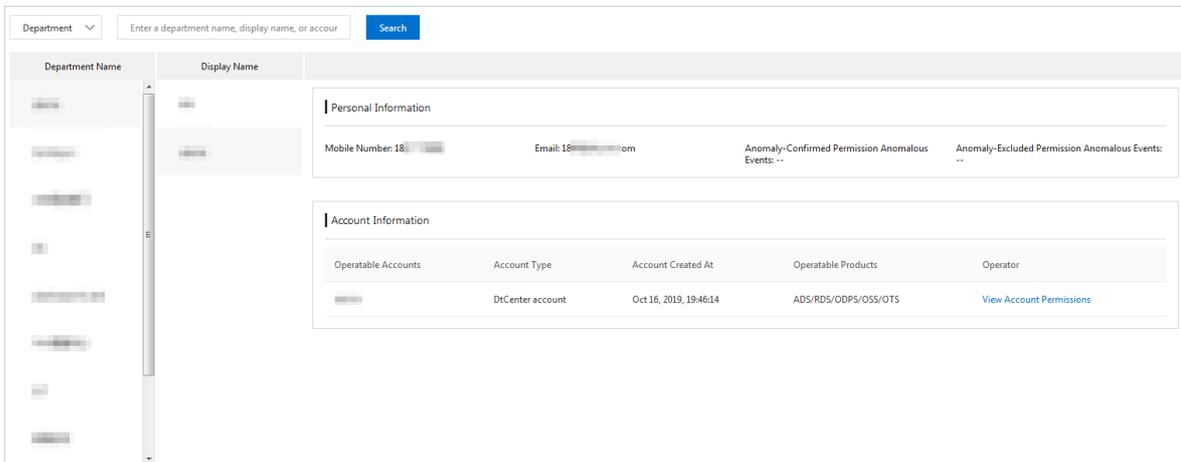
This topic describes how to view permissions of an account.

Context

You can search for an account to view its information so that you can quickly find the owner of sensitive data.

Procedure

1. [Log on to Apsara Stack Security Center.](#)
2. Choose **Data Security > Sensitive Data Protection > Data Permissions > Permission Search.**



3. Search for the target account. To search for an account, perform the following steps:
 - i. Select **Department** or **Employee** from the drop-down list.
 - ii. Enter a keyword in the search field.
 - iii. Click **Search**. The accounts that contain the keyword are listed in the **Display Name** column. You can also click a department in the **Department Name** column. All accounts of the department are listed in the **Display Name** column.
4. In the **Display Name** column, click the target account.
5. View information in the **Personal Information** and **Account Information** sections on the right.

- **Personal Information**

You can view the contact information of the account owner, the number of confirmed anomalous activities related to permission access, and the number of excluded anomalous activities related to permission access.

- **Account Information**

You can view the accounts that the owner can use, the types and creation time of the accounts, and Apsara Stack services that the accounts can access.

You can click **View Account Permissions** in the Actions column of an account to view the resources, resource types, resource paths, and operation permissions.

28.9.3.5. Monitor data flows

28.9.3.5.1. View data flows in DataHub

This topic describes how to view data flows in DataHub.

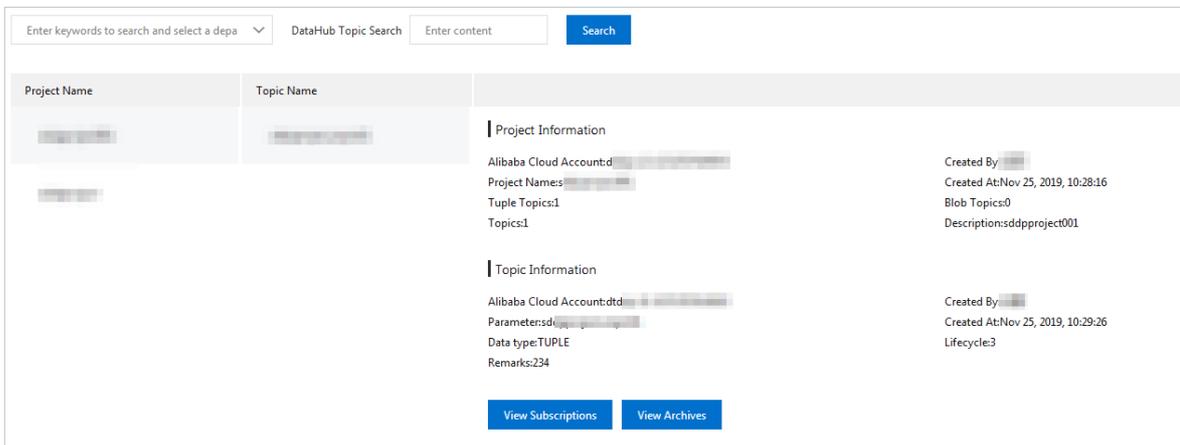
Context

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.

On the **DataHub** page, you can view the details of data flows in DataHub, including the relationships between DataHub projects and topics and the relationships among topics, subscribed applications, and archive sources.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. **Choose Data Security > Sensitive Data Protection > Data Flow Monitoring > DataHub.**
3. **Enter a keyword and select a department from the drop-down list. Enter a topic keyword in the DataHub Topic Search field and click Search.**



Note

You can also click the target project in the **Project Name** column and then click the target topic in the **Topic Name** column.

In the **Project Information** and **Topic Information** sections, you can view information about the project and topic.

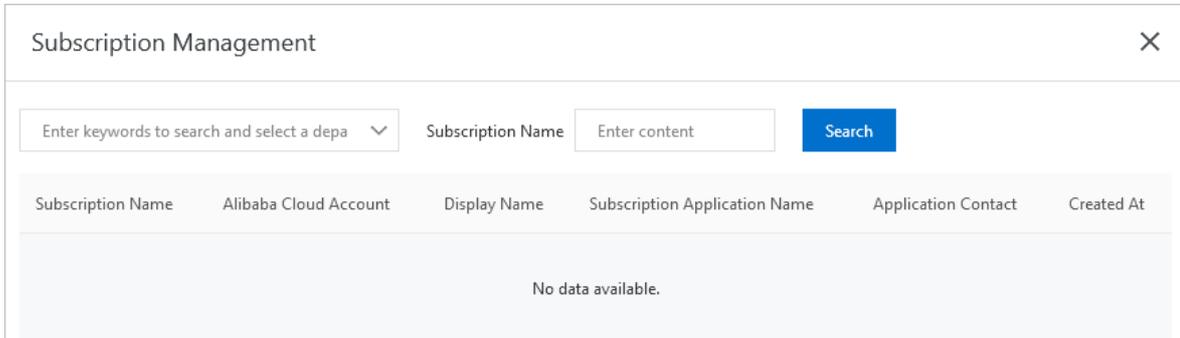
- **Project Information**

Displays information such as the project name, Apsara Stack account, creator, creation time, and number of topics.

- **Topic Information**

Displays information such as the project name, Apsara Stack account, creator, creation date, and type.

4. Click **View Subscriptions** to view the subscription list. The subscription list contains information such as the subscription name, Apsara Stack account of the creator, display name, name of the subscribed application, and application contact.



- i. Enter a keyword and select a department from the drop-down list.
 - ii. Enter a keyword in the **DataHub Topic Search** field.
 - iii. Click **Search** to search for the target DataHub topic.
5. Click **View Archives** to view the archive list. The archive list contains information such as the name of the connected instance, Apsara Stack account of the creator, display name, source service, resource path, and risk level.
 - i. Enter a keyword and select a department from the drop-down list.
 - ii. Enter a keyword in the **DataHub Topic Search** field.
 - iii. Click **Search** to find the target DataHub topic.

28.9.3.5.2. View data flows in Data Integration

This topic describes how to view data flows in Data Integration.

Context

DataWorks is a comprehensive, professional cloud R&D platform for big data. DataWorks serves as an operating system and delivers intelligent, efficient, secure, and reliable big data services. It meets your requirements for data governance and quality management, and allows you to provide data services for external users.

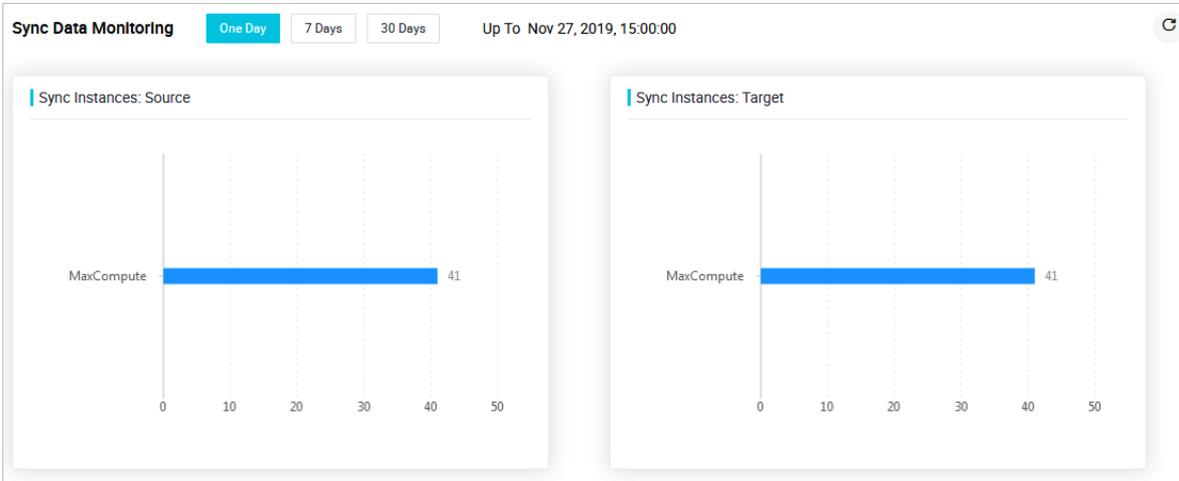
DataWorks provides the Data Integration feature, which is a stable, efficient, and scalable data synchronization platform on Apsara Stack. Data Integration implements fast and stable data transmission and synchronization between various data sources in complicated networks.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Data Flow Monitoring > CDP Flow Monitoring**.
3. On the **Data Integration** page, click **Sync Data Monitoring**.
4. View the statistical graph for the number of synchronized instances.

Note

- The number of synchronized instances is measured from two aspects: Sync Instances: Source and Sync Instances: Target.
- You can view statistics within different periods of time (One Day, 7 Days, and 30 Days).



5. In the Sync Instances section, view information such as the ID, time, node name, data type, and amount of synchronized data.

28.9.3.6. Sensitive data masking

28.9.3.6.1. Add a static desensitization task

This topic describes how to add a static desensitization task and run the task to mask sensitive data.

Procedure

1. Log on to Apsara Stack Security Center.
2. In the left-side navigation pane, choose Data Security > Sensitive Data Protection > Sensitive Data Desensitization > Static Desensitization.
3. In the upper-right corner of the Desensitization Tasks section, click Add Desensitization Task.
4. On the Add Desensitization Task page, configure the parameters.
 - i. In the Basic Task Information step, specify Task Name and Remarks. Then, click Next.
 - ii. In the Desensitization Source Configuration step, configure the parameters and click Next.

Source Type MaxCompute/RDS Table/AnalyticDB Table Bucket File

Source Product

Project

Source Partition

Product	Parameter	Description
	Source Type	Select MaxCompute/RDS Table/AnalyticDB Table.

Product	Parameter	Description
MaxCompute	Source Product	Select MaxCompute.
	Project	Select the project that contains the table with the sensitive data you want to desensitize.
	Table Name	Select the name of the table with the sensitive data you want to desensitize.
	Source Partition	<p>Enter the name of the partition that contains the sensitive data you want to desensitize.</p> <p>You can configure partitions when you create a MaxCompute table. Partitions define different logical divisions of a table to help you efficiently query specific content.</p> <p>Note If you leave Source Partition empty, SDDP desensitizes sensitive data in all partitions of the table.</p>
RDS	Source Type	Select MaxCompute/RDS Table/AnalyticDB Table.
	Source Product	Select RDS.
	Project	Select the database that contains the table with the sensitive data you want to desensitize.
	Table Name	Select the name of the table with the sensitive data you want to desensitize.
	Sample SQL	Optional. Enter an SQL statement to select the data you want to desensitize.
OSS	Source Type	Select Bucket File.
	File Source	<p>Upload a file from a bucket or your local computer.</p> <ul style="list-style-type: none"> ■ Uploaded Local File: If you select this option, click Select File to select a local file. ■ Bucket: If you select this option, select a file from the Source File drop-down list.
	Source File Description	<p>Enter the remarks of the file to quickly identify the task.</p> <p>Note Specify this parameter only when File Source is set to Uploaded Local File.</p>
	Source File	<p>Select a file from an OSS bucket.</p> <p>Note Specify this parameter only when File Source is set to Bucket.</p>
	Source File Name	<p>Optional. Enter the name of the file from the bucket.</p> <p>Note Specify this parameter only when File Source is set to Bucket.</p>

- iii. In the **Desensitization Algorithm Configuration** step, configure the parameters and click **Next**. You must specify the algorithm type, select an algorithm, and turn on the desensitization switch for the source field you want to desensitize.
- iv. In the **Destination Location Configuration** step, configure the parameters and click **Next**.
- v. In the **Confirm Process Logic** step, configure the parameters to confirm the processing logic.

Parameter	Description
Select Trigger Method	<p>The mode in which the desensitization task runs. Valid values:</p> <ul style="list-style-type: none"> ▪ Manual Only: You must manually run the desensitization task on the Static Desensitization page. ▪ Scheduled Only: The desensitization task automatically runs at a specified time on an hourly, daily, or monthly basis. ▪ Manual + Scheduled: You can manually run the desensitization task or enable automatic desensitization at a specified time on an hourly, daily, or monthly basis.
Table Name Conflict Resolution	<p>The solution used when the specified target table name already exists. Valid values:</p> <ul style="list-style-type: none"> ▪ Delete the target table and create a new table with the same name ▪ Insert new data to the target table (recommended)
Row Conflict Resolution	<p>The handling method for a row conflict. Valid values:</p> <ul style="list-style-type: none"> ▪ Keep conflicting rows in the target table and discard the new data (recommended) ▪ Delete conflicting rows from the target table and insert the new data

- vi. Click **Submit**.

After the task is created, you can view it in the **Desensitization Tasks** section.

- 5. In the **Desensitization Tasks** section, turn on the switch and click **Start** to run the created desensitization task.
- 6. In the **Task Execution Query** section, view **Execution Progress** and **Status** of the task.

28.9.3.7. Abnormal activity detection

28.9.3.7.1. Add a custom rule for abnormal activities

This topic describes how to add a custom rule for abnormal activities. If Sensitive Data Discovery and Protection (SDDP) detects an activity that matches the custom rule, it identifies the activity as abnormal.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Anomaly Detection > Anomalous Event Rules**.
3. On the **Anomalous Event Rules** page, click **Add Rule**.
4. In the **Add Rule** dialog box, configure the parameters.

Add Rule
✕

Rule Name

Risk Level

Low
▼

Asset Type

MaxCompute
▼

Filters ?

Select
▼

Select
▼

+ Add

Warning conditions ?

Select
▼

Select
▼

Select
▼

OK

Cancel

Parameter	Description
Rule Name	The name of the rule.
Risk Level	The risk level of the rule. Valid values: Low , Medium , and High .
Asset Type	The type of the service to which the rule applies. Valid values: OSS , MaxCompute , and RDS .
Filters	The filter conditions of the rule.
Warning Conditions	The warning conditions of the rule. Activities that match the filter conditions are checked, and alerts are sent for abnormal activities.

5. Click OK.

What's next

After you create a custom rule, SDDP checks for abnormal activities based on the rule. You can view statistics of the abnormal activities on the **Custom Anomalous Event** tab of the **Anomalous Events** page. For more information, see [Process anomalous activities](#).

28.9.3.7.2. Process abnormal activities

This topic describes how to process abnormal activities in Sensitive Data Discovery and Protection (SDDP). SDDP detects abnormal activities related to sensitive data and generates alerts. On the **Anomalous Events** page, you can confirm abnormal activities as violations or exclude them as false positives.

Context

SDDP divides abnormal activities into the following types:

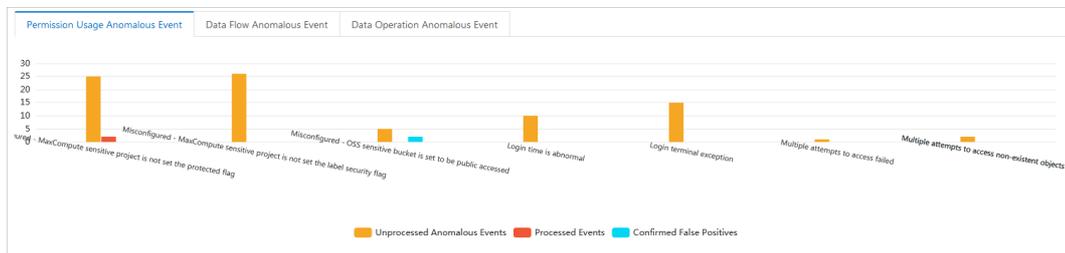
- **Permission Usage Anomalous Event:** Permissions are used in an inappropriate way. For example, a user logs on from an unusual IP address or by using the AccessKey pair of another user.

- **Data Flow Anomalous Event:** Abnormal activities are detected in data flows. For example, a user downloads sensitive data files unnecessarily or during an unusual period.
- **Data Operation Anomalous Event:** Unusual operations are performed on sensitive data. For example, a user modifies sensitive fields.
- **Custom Anomalous Event:** Abnormal activities are detected based on custom rules. For more information, see [Add a custom rule for abnormal activities](#).

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Anomaly Detection > Anomalous Events**.
3. View the statistics on abnormal activities. You can view statistics on different types of abnormal activities in the upper part of the **Anomalous Events** page. The statistics include the types of abnormal activities, number of processed abnormal activities, and number of unprocessed abnormal activities.

Statistics on abnormal activities



- Click the **Permission Usage Anomalous Event**, **Data Flow Anomalous Event**, **Data Operation Anomalous Event**, or **Custom Anomalous Event** tab to view the bar charts of the statistics.
 - Move the pointer over the chart to view the details.
4. Configure search conditions and click **Search**. You can search for abnormal activities by keyword, department, type, status, and alert time.
 5. Process abnormal activities. In the lower part of the **Anomalous Events** page, process abnormal activities detected by SDDP in the abnormal activity list.

Account	Department	Event Type	Event Subtype	Alert Time	Status	Operator
dtdep-13-157-129547170741		Custom exceptions		Dec 7, 2019, 08:03:56	To be processed	View Details Process
dtdep-13-157-129547170741		Custom exceptions		Dec 7, 2019, 08:03:56	To be processed	View Details Process
dtdep-13-157-129547170741		Custom exceptions		Dec 7, 2019, 07:37:48	To be processed	View Details Process
dtdep-13-157-129547170741		Custom exceptions		Dec 7, 2019, 07:02:46	To be processed	View Details Process

- i. Find the target abnormal activity, and click **View Details** in the **Actions** column to view the details of the abnormal activity.
- ii. Find the target abnormal activity, and click **Process** in the **Actions** column.

iii. In the Anomalous Event Processing dialog box, process the abnormal activity.

Parameter	Description
Add Processing Record	Check the abnormal activity and record the verification process.
Anomalous Event Verification	<ul style="list-style-type: none"> ▪ Confirmed and Processed: Confirm that the activity is abnormal. If you select this option without manually processing the abnormal activity, SDDP keeps generating alerts for this activity. ▪ False Positive: Set the abnormal activity as a false positive. If you select this option, SDDP no longer generates alerts for the activity. This activity will no longer appear on the Anomalous Events page.
Anomalous Event Sample-based Enhancement	<p>Specify whether to enhance detection of abnormal activities based on the processing result of the abnormal activity.</p> <div style="background-color: #e1f5fe; padding: 10px;"> <p> Note If you select this option:</p> <ul style="list-style-type: none"> ▪ An abnormal activity that is set as a false positive is returned to the algorithm as a false positive sample. ▪ An abnormal activity that is confirmed as a violation is returned to the algorithm as a positive sample. <p>This improves the accuracy of subsequent abnormal activity detection, but may increase the false negative rate.</p> </div>

iv. Click Completed.

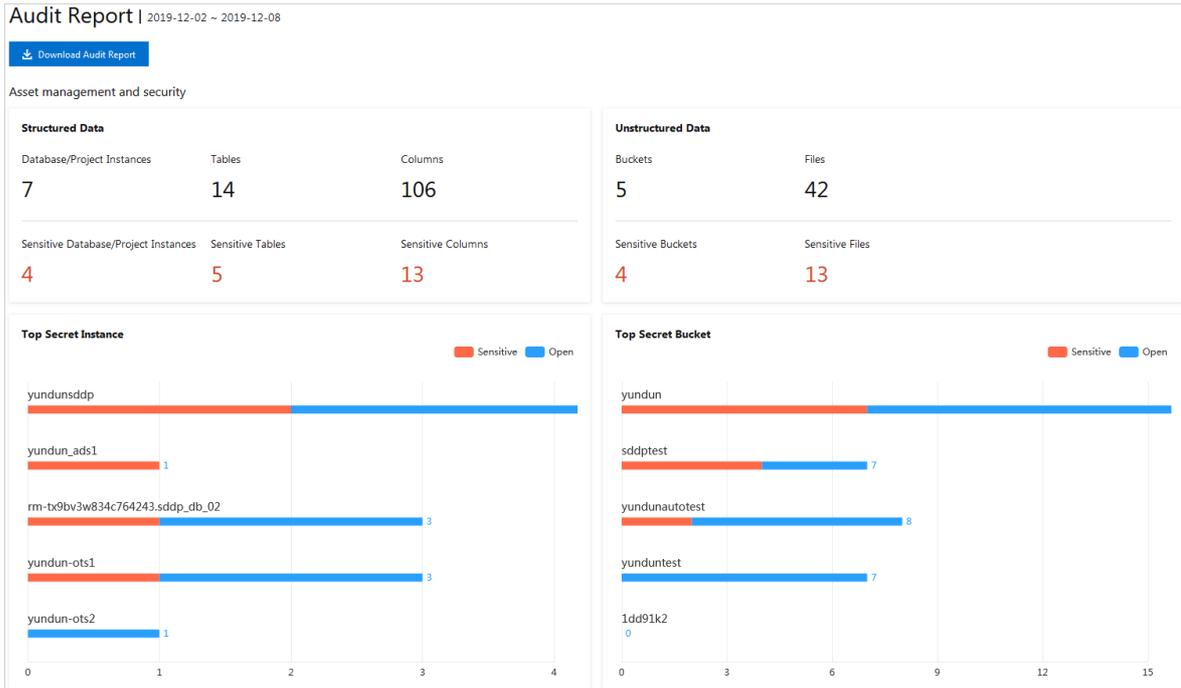
28.9.3.8. Intelligent audit

28.9.3.8.1. View and download audit reports

This topic describes how to view and download audit reports of sensitive data.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Intelligent Audit > Audit Report**.
3. On the **Audit Report** page, view an audit report.



4. Click **Download Audit Report**.

28.9.3.8.2. View audit logs

This topic describes how to view audit logs.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Intelligent Audit > Audit Logs**.
3. (Optional) On the **Audit Logs** page, specify the time period to filter audit logs.

Dec 5, 2019 - Dec 12, 2019 Reset Advanced Search

Select an asset type Search Reset

If you require more detailed filter conditions, follow these steps:

- i. Click **Advanced Search**.
 - ii. Specify filter conditions such as the asset type, asset name, account, and source IP address.
 - iii. Click **Search**.
4. View audit logs in the log list.

28.9.3.8.3. View raw logs

This topic describes how to view raw logs of Object Storage Service (OSS), MaxCompute, and ApsaraDB for RDS.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Intelligent Audit > Raw Logs**.
3. On the **Raw Logs** page, click the required tab. On the **Raw Logs** page, you can view raw logs of OSS,

MaxCompute, and RDS.

4. (Optional) Specify conditions to filter raw logs.
5. View information of raw logs in the log list.

Bucket	Object	Account	Time	Source IP Address	User-Agent	Operation Type	Status Code	Operator
sddptest	xxxxxxx.xlsx	1295xxxxxx	12/Dec/2019:17:32:13 +0800	10.0.0.0	aliyun-sdk-go/1.9.3 (Linux/4.9.1...	GetObject	200	Details
sddptest	xxxxxxx.yid.doc	1295xxxxxx	12/Dec/2019:17:32:13 +0800	10.0.0.0	aliyun-sdk-go/1.9.3 (Linux/4.9.1...	GetObject	200	Details
sddptest	xxxxxxx	1295xxxxxx	12/Dec/2019:17:32:13 +0800	10.0.0.0	aliyun-sdk-go/1.9.3 (Linux/4.9.1...	GetObject	200	Details

6. Find the target raw log, and click **Details** in the **Operator** column.

28.9.3.8.4. Add an audit rule

This topic describes how to add an audit rule. Sensitive Data Discovery and Protection (SDDP) audits raw logs that match the audit rule.

Procedure

1. Log on to [Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Intelligent Audit > Audit Rules**.
3. Click **Add Rule**.
4. In the **Add Rule** dialog box, configure the parameters.

Parameter	Description
Rule Name	The name of the audit rule.
Risk Level	The risk level of the rule. Valid values: Low , Medium , and High .
Asset Type	The type of the asset.
Filters	The filter conditions of the rule. Raw logs that match the filter conditions are audited.

5. Click **OK**.

Result

After you create an audit rule, it is displayed in the rule list. You can view its details or click the operations in the **Operator** column to edit or delete the rule.

28.9.3.9. Security configuration

28.9.3.9.1. Manage rules used to identify sensitive data

This topic describes how to create and manage rules used to identify sensitive data.

Context

Sensitive Data Discovery and Protection (SDDP) can identify and classify sensitive data in Apsara Stack services such as MaxCompute, Object Storage Service (OSS), and Table Store.

SDDP identifies sensitive data based on the rules. You can use built-in rules provided by SDDP or configure custom rules based on your business needs to identify sensitive data.

Procedure

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Rule Configuration.**
3. Click the **Sensitive Data Identification Rules** tab to view the existing rules used to identify sensitive data.

Rule Name	Rule Type	Rule Source	Risk Level	Operator
AccessKeyId	Regular expression	Built-in		<input checked="" type="checkbox"/> Delete Details
AccessKeySecret	Regular expression	Built-in		<input checked="" type="checkbox"/> Delete Details
IPv6 address	Regular expression	Built-in		<input checked="" type="checkbox"/> Delete Details
GPS position	Regular expression	Built-in		<input checked="" type="checkbox"/> Delete Details

Specify **Rule Type**, **Risk Level**, and **Rule Name**. Then, click **Search** to search for rules.

SDDP provides built-in algorithms to identify sensitive data such as ID card numbers, addresses, phone numbers, and bank card numbers. It uses file clustering, deep neural network, and machine learning to identify sensitive images, text, and fields.

4. Create a rule. You can create a rule to identify sensitive data.
 - i. Click **Add Rule**.
 - ii. In the **Add Rule** dialog box, configure the parameters.

* Rule Type

* Rule Name

* Risk Level

* Rule Definition

- **Rule Type:** the type of the rule. Valid values: *Keyword* and *Regular expression*.
- **Rule Name:** the name of the rule. We recommend that you name the rule based on its purpose.
- **Risk Level:** the risk level of the rule. Valid values: *S1 (low)*, *S2 (medium)*, *S3 (high)*, and *S4 (critical)*.
- **Rule Definition:** the content of the rule.

iii. Click **Submit**.

5. Manage the rules. In the rule list, click an operation in the **Operator** column to disable, enable, or delete a rule.

Note

- You can delete rules but cannot modify them. After you delete a rule, SDDP no longer uses it to identify sensitive data. Proceed with caution.
- A rule is enabled by default after it is created. If you do not want to identify sensitive data that matches a rule, you can disable the rule. After you disable a rule, SDDP no longer uses it to identify sensitive data. We recommend that you enable all rules to reduce risks.
- Rules marked as *Built-in* in the Rule Source column are default rules. If no custom rules are configured, SDDP identifies sensitive data based on these default rules. The default rules cannot be modified or deleted.

28.9.3.9.2. Manage thresholds and rules used to detect abnormal activities

This topic describes how to manage the thresholds and rules used to detect abnormal activities.

Context

On the Rule Configuration page, you can customize the thresholds and rules used to detect abnormal activities.

Procedure

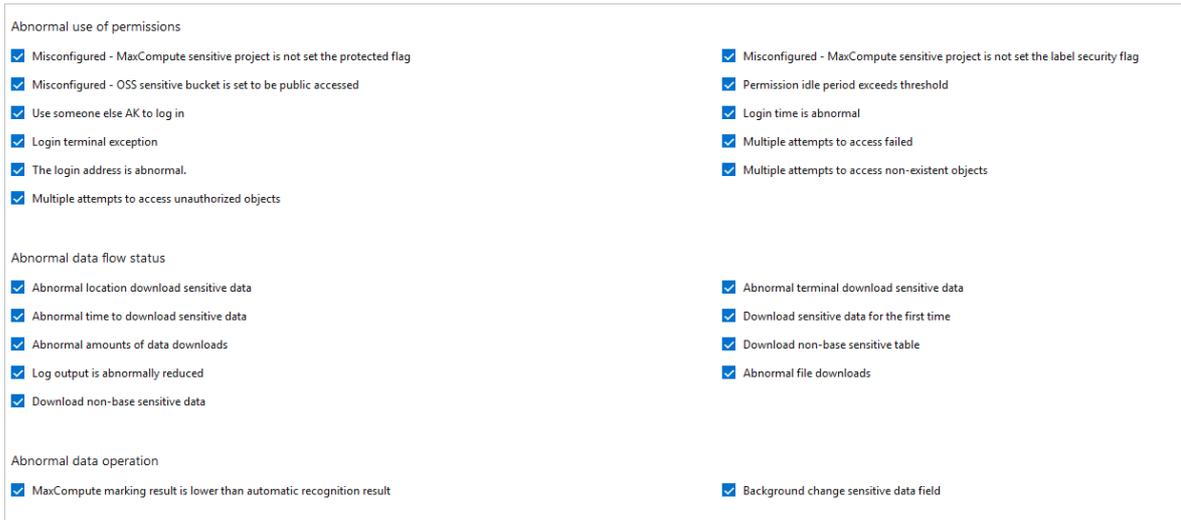
1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Rule Configuration.**
3. Click the **Anomaly Alert Configuration** tab to view the thresholds and rules used to detect abnormal activities.
4. Configure the thresholds used to detect abnormal activities.



Sensitive Data Discovery and Protection (SDDP) provides default thresholds that can be customized.

- i. In the **General Configuration for Anomaly Alerts** section, click **Modify** next to a threshold.
 - ii. Enter a value and click **Submit**.
5. Configure the rules used to detect abnormal activities.

In the **Enable Anomaly Alerts** section, select the types of abnormal activities that you want SDDP to detect.



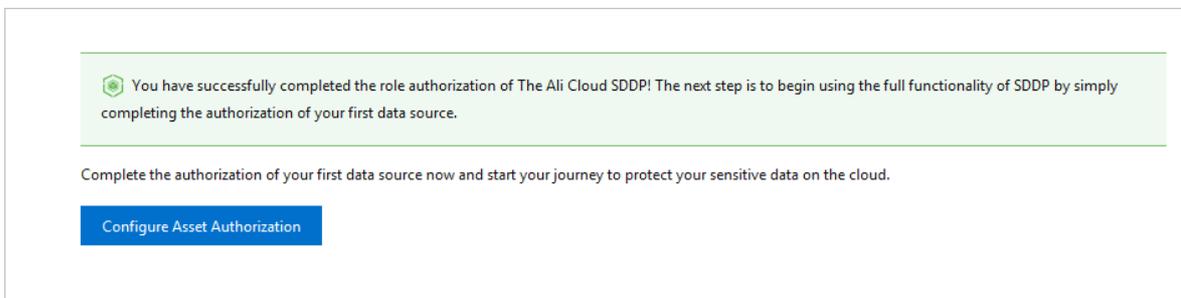
After you configure the rules, you can view statistics on suspicious events on the **Anomalous Events** page.

28.9.3.9.3. Configure an authorized asset

This topic describes how to configure an authorized asset for Sensitive Data Discovery and Protection (SDDP). After the configuration, SDDP scans the authorized asset. You can view abnormal activities of data leaks or sensitive data on the **Anomalous Events** page.

Procedure

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Authorization Configuration**.
3. On the **Authorization Configuration** page, click **Configure Asset Authorization**.



4. Select a region from the **Region** drop-down list and a database from the **Database Name** drop-down list.

5. Click **Complete Authorization**.

Result

After the configuration is complete, SDDP scans the authorized asset.

28.9.3.9.4. Configure desensitization algorithms

This topic describes how to configure desensitization algorithms.

Context

Sensitive Data Discovery and Protection (SDDP) supports the following desensitization methods:

- **Hashing:** performs tokenization or masks passwords. Raw data cannot be retrieved after it is desensitized by using this method. The MD5, Secure Hash Algorithm 1 (SHA-1), SHA-256, and hash-based message authentication code (HMAC) salted algorithms are supported.
- **Masking:** replaces targeted information in sensitive data with asterisks (*) or number signs (#) in any of the following ways:
 - Keep the first N characters and the last M characters.
 - Keep characters from the Xth position to the Yth position.
 - Mask the first N characters and the last M characters.
 - Mask characters from the Xth position to the Yth position.
 - Mask characters before a special character.
 - Mask characters after a special character.
- **Replacement:** uses a mapping table or interval to randomly replace or map entire or partial field values. Raw data can be retrieved after it is desensitized by using this method. SDDP provides multiple built-in mapping tables in the .txt and .rtf formats and allows you to add custom replacement algorithms. This method is suitable for fields with a fixed format, such as ID card numbers.
- **Transformation:** rounds or offsets field values to desensitize them. You can round numbers and dates or offset characters in text based on specified parameters. Raw data can be retrieved after it is offset but cannot be retrieved after it is rounded.
- **Encryption:** uses the Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), or Advanced Encryption Standard (AES) algorithm to encrypt data. Raw data can be retrieved after it is desensitized by using this method.
- **Shuffling:** shuffles values of a field within a specified range of a source table. The values can be shuffled randomly or be offset in a specified way. Raw data can be retrieved after it is offset but cannot be retrieved after it is shuffled.

Hashing

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration.**
3. Click the **Hashing** tab.
4. Set a salt value for each algorithm.

 **Note**

In cryptography, you can insert a specific string to a fixed position of a password to generate a hash value that is different from that of the original password. This process is called salting.

A salt value is the specific string that you insert.

MD5	<input type="text" value="15ef85cd"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
SHA1	<input type="text" value="Enter a salt value"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
SHA256	<input type="text" value="Enter a salt value"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
HMAC	<input type="text" value="Enter a salt value"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>

5. Click **Test** for an algorithm. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.

Desensitization Algorithm Test ✕

Enter an original value

Desensitization Result

6. After the test, close the **Desensitization Algorithm Test** dialog box. Then, click **Save**.

Masking

1. [Log on to Apsara Stack Security Center.](#)
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration.**
3. Click the **Masking** tab.
4. Configure the parameters.

Select Source Type *

* #

Keep the First N Characters and the Last M Characters

n m

Keep Characters from the Xth Place to the Yth Place

x y

Mask the First N Characters and the Last M Characters

n m

Mask Characters from the Xth Place to the Yth Place

x y

Special character front cover (for the first time the character appears)

@ & .

After masking of special characters (for the first appearance of the character)

@ & .

5. Click **Test** for an algorithm. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the test, close the **Desensitization Algorithm Test** dialog box. Then, click **Save**.

Replacement

1. [Log on to Apsara Stack Security Center](#).
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration**.
3. Click the **Replacement** tab.
4. Configure the parameters.

[Add Replacement Desensitization Algorithm](#)

ID Card Number Mapping Replacement
[Random Administrative Region Code Table](#)

Algorithm validation check (ID, Bankcards)

[Save](#) [Test](#)

ID Card Number Random Replacement
[Random Administrative Region Code Table](#)

Jan 1, 1920 - Jan 1, 2130

Algorithm validation check (ID, Bankcards)

[Save](#) [Test](#)

Military ID Random Replacement
[Random Administrative Region Code Table](#)

Random Military ID Interval 0 - 999999

[Save](#) [Test](#)

Passport Number Random Replacement
[Purpose Field Random Code](#)

Random Passport Number Interval 1 - 99999999

[Save](#) [Test](#)

Random Replacement for Hong Kong & Macao Exit-Entry Permit Number
[Purpose Field Random Code](#)

Random Hong Kong & Macao Exit-Entry Permit Number Interval 100 - 99999999

[Save](#) [Test](#)

Note By default, SDDP provides multiple common replacement algorithms, such as ID Card Number Mapping Replacement and Telephone Number Random Replacement.

- If you want to customize a mapping table, click the required mapping table, replace the original content with your own mapping table, and click **Save**.
 - If you need to customize an algorithm, click **Add Replacement Desensitization Algorithm** and specify the interval and mapping table.
5. Click **Test** for an algorithm. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
 6. After the test, close the **Desensitization Algorithm Test** dialog box. Then, click **Save**.

Transformation

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration.**
3. Click the **Transformation** tab.
4. Configure the parameters.

Number Rounding	Deciman rounding level	<input type="text" value="1"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
Date Rounding	Date rounding level	<input type="text" value="Month"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
Character Offset	Number of cyclical bits offset	<input type="text" value="0"/>	<input type="radio"/> Left <input type="radio"/> Right	<input type="button" value="Test"/> <input type="button" value="Submit"/>

5. Click **Test** for an algorithm. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the test, close the **Desensitization Algorithm Test** dialog box. Then, click **Save**.

Encryption

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration**.
3. Click the **Encryption** tab.
4. Specify a key for an algorithm.

DES	<input type="text" value="1"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
3DES	<input type="text" value="2"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
	<input type="text" value="2"/>		
AES	<input type="text" value="3"/>	<input type="button" value="Test"/>	<input type="button" value="Submit"/>
	<input type="text" value="1"/>		

5. Click **Test** for an algorithm. In the **Desensitization Algorithm Test** dialog box, enter the original value and click **Test** to check whether the algorithm works.
6. After the test, click the close icon in the **Desensitization Algorithm Test** dialog box. Then, click **Save**.

Shuffling

1. **Log on to Apsara Stack Security Center.**
2. In the left-side navigation pane, choose **Data Security > Sensitive Data Protection > Security Configuration > Desensitization Algorithm Configuration**.
3. Click the **Shuffling** tab.
4. Select a shuffling method.

Randomly Shuffle	Shuffling Method	<input checked="" type="radio"/> Reset <input type="radio"/> Random Selection	<input type="button" value="Submit"/>
------------------	------------------	---	---------------------------------------

5. Click **Save**.

29. Key Management Service (KMS)

29.1. Manage keys in the KMS console

29.1.1. Log on to the KMS console

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

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Security > Key Management Service**.

29.1.2. Create a CMK

This topic describes how to create a CMK for subsequent encryption and decryption operations.

Procedure

1. **Log on to the KMS console.**
2. On the **Keys** page, click **Create Key**.
3. On the **Create Key** page, select an organization from the **Organizations** drop-down list. Then, **Resource Set** and **Region** are automatically specified.
4. In the **Basic Settings** section, specify **Key Spec**, **Purpose**, and **Protection Level**.
5. Specify **Alias** and **Description**.
6. In the **Advanced Settings** section, specify **Rotation Period**.
 - **Disable:** The CMK is not automatically rotated.
 - **Enable:** The CMK is automatically rotated. You can select or customize a rotation period.
7. Specify **Key Material Source**.
 - **Alibaba Cloud KMS:** Use KMS to generate key material.
 - **External:** Manually import external key material.

 **Note** If you select **External**, it means that you have read and understood the implications of importing external key material.

8. Click **Submit**.

29.1.3. View CMK details

After you create a CMK, you can view the details about the CMK, such as the key ID, status, purpose, and creation time.

Procedure

1. [Log on to the KMS console](#).
2. On the **Keys** page, find the target CMK, and click its alias or choose **More > Key Details** in the **Actions** column.
3. In the **Key Details** section, view the details about the CMK, such as the key ID, status, purpose, and creation time.

29.1.4. Enable a CMK

This topic describes how to enable a CMK.

Procedure

1. [Log on to the KMS console](#).
2. On the **Keys** page, find a CMK that is in the **Disabled** state and click **Enable** in the **Actions** column.
3. In the **Enable Key** message, click **OK**.

29.1.5. Disable a CMK

This topic describes how to disable a CMK.

Context

If a CMK is disabled, it cannot be used for encryption or decryption. The ciphertext encrypted by using the CMK cannot be decrypted until the CMK is enabled again.

Procedure

1. [Log on to the KMS console](#).
2. On the **Keys** page, find a CMK that is in the **Enabled** state and click **Disable** in the **Actions** column.
3. In the **Disable Key** message, click **OK**.

 **Note** After you disable the CMK, its state changes from **Enabled** to **Disabled**.

29.1.6. Schedule the deletion of a CMK

You can schedule the deletion of a CMK by specifying a waiting period. After the period elapses, the CMK is automatically deleted.

Context

You must specify a waiting period of 7 to 30 days.

 **Warning**

- During the waiting period, the CMK is in the Pending Deletion state. It cannot be used to encrypt data, decrypt data, or generate data keys.
- Deleting a CMK may have a severe impact on data availability. Therefore, in most cases, we recommend that you disable a CMK instead of deleting the CMK.
- After a CMK is deleted, it cannot be recovered. The data encrypted by using this CMK and the ciphertext data keys generated by using this CMK cannot be decrypted. Therefore, KMS only allows you to schedule the deletion of a CMK. You cannot immediately delete a CMK.
- KMS deletes the CMK within 24 hours after the specified waiting period elapses.

For example, if you submit a deletion application at 14:00, September 10, 2017 and specify a waiting period of seven days, KMS deletes the CMK within 24 hours after 14:00, September 17, 2017.

Procedure

1. [Log on to the KMS console](#).
2. On the **Keys** page, find the target CMK and choose **More > Schedule Key Deletion** in the **Actions** column.
3. In the **Schedule Key Deletion** dialog box, specify **Delete In (7-30 days)**.
4. Click **OK**.

The state of the CMK becomes **Pending Deletion**.

Before the specified period ends, you can perform the following operations to cancel the deletion: Choose **More > Cancel Key Deletion** in the **Actions** column. In the **Cancel Key Deletion** message, click **OK**.

29.1.7. Configure automatic rotation of a CMK

This topic describes how to configure automatic rotation of a CMK.

Procedure

1. [Log on to the KMS console](#).
2. On the **Keys** page, find a symmetric key and choose **More > Key Details** in the **Actions** column. You can determine whether a CMK is a symmetric key based on the value of **Key Spec**.
3. In the **Key Version** section, click **Set Rotation Policy**.
4. In the **Set Rotation Policy** dialog box, specify **Rotation Period**.The following options are available:
 - **30 Days, 90 Days, 180 Days, 365 Days**: If you select one of these options, the next rotation date is displayed.
 - **Disable**: If you select this option, the CMK is not automatically rotated.
 - **Customize**: If you select this option, you must specify **Days (7–730)**. The next rotation date is displayed.
5. Click **OK**.

Related information

- [Automatic key rotation](#)
- [Manual key rotation](#)

29.2. Use a CLI to manage CMKs

KMS allows you to use a command-line interface (CLI) to create, delete, enable, disable, and view CMKs. This topic describes how to use a CLI to manage CMKs.

Prerequisites

The CLI tool required for your operating system is installed.

- **Windows**: <https://www.alibabacloud.com/help/doc-detail/121510.htm>

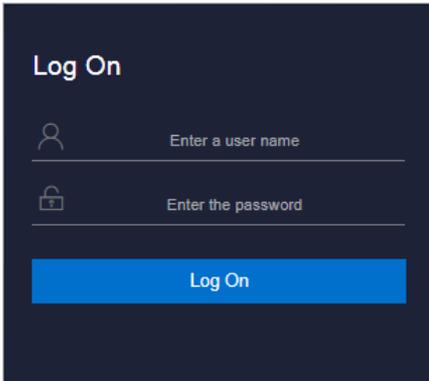
- Linux: <https://www.alibabacloud.com/help/doc-detail/121541.htm>
- macOS: <https://www.alibabacloud.com/help/doc-detail/121544.htm>

Context

CMKs are basic resources of KMS. A CMK is composed of a key ID, basic metadata (such as key state), and key material that is used to encrypt and decrypt data. You can create a CMK and call API operations to encrypt data, decrypt data, and generate data keys. You can also manage the lifecycle of CMKs.

Obtain the domain name and region of KMS

1. Open the browser. Enter the ASO access address `http://region-id.aso.intranet-domain-id.com` in the address bar and then press Enter.



Note You can select the language from the drop-down list in the upper-right corner to change the language of ASO.

2. Enter the correct username and password.
 - The system has three default users:
 - Security officer: manages other users or roles.
 - Auditor officer: views audit logs.
 - System administrator: used for other functions except those of the security officer and auditor officer.
 - You must modify the password of your username as instructed when you log on to ASO for the first time. To improve security, the password must be 10-20 characters long and can contain English uppercase/lowercase letters (A-Z or a-z), numbers (0-9), and special characters, such as exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%), which meets the minimum complexity requirements.
3. Click **Log On** to log on to ASO.
4. In the left-side navigation pane, click **Products > Product List > Apsara Infrastructure Management Framework**.
5. On the **All Reports** page that appears, search for **Registration Vars of Services**. Click **Registration Vars of Services**.
6. On the **Registration Vars of Services** page, move the pointer over **Service** and click the  icon that appears. In the dialog box that appears, search for **kms**.
7. Right-click the value in the **Service Registration** column that corresponds to **kms** and select **Show More** from the shortcut menu.
8. In the **Details** dialog box, check the value of **kms-intranet-domain**. This value is the domain name of KMS.

Note The following information is used in the subsequent sections of this topic:

- Domain name: kms.example.com
- Region: cn-qingdao
- AccessKey ID: \${please-replace-your-access-key-id}
- AccessKey secret: \${please-replace-your-access-key-secret}

Create a CMK

Call the CreateKey operation to create a CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms CreateKey --Description "this is a test key"
```

Information of the created CMK is returned. KeyId is the globally unique identifier of the CMK. In all subsequent operations that require the CMK, you need to use KeyId to specify the CMK.

Sample success responses

```
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167: :key/769472ec-07d6-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "133418746167",
    "DeleteDate": "",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "f0bd45b8-d7ec-4e5d-a54c-527db2b7e784"
}
```

Query the details about a specified CMK

Call the DescribeKey operation to query the details about a specified CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms DescribeKey --KeyId 769472ec-07d6-47e8-a32a-bc163f66****
```

Sample success responses

```
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167:key/769472ec-07d6-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "133418746167",
    "DeleteDate": "",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "22756da2-a295-4ad4-a532-e739e35c48dd"
}
```

Query CMKs

Call the ListKeys operation to query CMKs.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms ListKeys --PageNumber 1 --PageSize 10
```

Sample success responses

```
{
  "Keys": {
    "Key": [
      {
        "KeyArn": "acs:kms:cn-neimeng-poc2-d01:176448502412:key/090ea1aa-12db-4f6f-63f7ff612426",
        "KeyId": "090ea1aa-12db-4f6f-b2d3-63f7ff61"
      },
      {
        "KeyArn": "acs:kms:cn-neimeng-poc2-d01:176448502412:key/0cf4ed90-de6d-46be-561d7638f943",
        "KeyId": "0cf4ed90-de6d-46be-ad12-561d7638"
      },
      {
        "KeyArn": "acs:kms:cn-neimeng-poc2-d01:176448502412:key/1fa4613a-6927-45bf-ff0fe0ddb4e4",
        "KeyId": "1fa4613a-6927-45bf-9672-ff0fe0dd"
      },
      {
        "KeyArn": "acs:kms:cn-neimeng-poc2-d01:176448502412:key/1fd9ba48-a535-4ce2-d2a78bcb00af",
        "KeyId": "1fd9ba48-a535-4ce2-b1aa-d2a78bcb"
      },
      {
        "KeyArn": "acs:kms:cn-neimeng-poc2-d01:176448502412:key/2e588255-93b1-475b-6b7b307534cf",
        "KeyId": "2e588255-93b1-475b-9b74-6b7b3075"
      }
    ]
  },
  "PageNumber": 1,
  "PageSize": 5,
  "RequestId": "3668ea48-9dbb-4d82-b0ca-5d961f8f9031",
  "TotalCount": 17
}
```

Disable a CMK

Call the DisableKey operation to disable a CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms DisableKey --KeyId 769472ec-07d6-47e8-a32a-bc163f66****
```

After you disable a CMK, you can call the DescribeKey operation to check whether the CMK state becomes Disabled.

Sample success responses

```
[root@a34h05001.cloud.h05.amtest4 /apsarapangu/disk4/tmp/xtermupload/kmstest]
#./aliyun --skip-secure-verify --region cn-qingdao-env25-d01 --endpoint kms-intranet.cn-qingdao-env25-d01.intra.env25.shuguang.com --access-key-id b59F0cNyeq0isdJ7 --access-key-secret wdmybEGDoPrvgFZtNHGXoX2nn2x1j3 kms DescribeKey --KeyId 769472ec-07d6-47e8-a32a-bc163f66-47e8-a32a-bc163f66
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167:iam::key/769472ec-07d6-47e8-a32a-bc163f66-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "13341874616704",
    "DeleteDate": "",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Disabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba-1111",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "7f3c97a1-ae00-4e8e-9475-0ff9f87f1661"
}
```

Enable a CMK

Call the EnableKey operation to enable a CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms EnableKey --KeyId 769472ec-07d6-47e8-a32a-bc163f66****
```

After you enable a CMK, you can call the DescribeKey operation to check whether the CMK state becomes Enabled.

Sample success responses

```
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167:iam::key/769472ec-07d6-47e8-a32a-bc163f66-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "13341874616704",
    "DeleteDate": "",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba-1111",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "3bfac037-7026-4748-b5be-de4c2cd1fe44"
}
```

Schedule the deletion of a CMK

Call the ScheduleKeyDeletion operation to schedule the deletion of a CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms ScheduleKeyDeletion --KeyId 769472ec-07d6-47e8-a32a-bc163f66**** --PendingWindowInDays 7
```

After you schedule the deletion of a CMK, you can call the DescribeKey operation to check the state and deletion date of the CMK. If the deletion is scheduled, the CMK state is PendingDeletion, and the deletion date is the current date plus the value of PendingWindowInDays. The value range of PendingWindowInDays is 7 to 30 days.

Sample success responses

```
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167:example.com:key/769472ec-07d6-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "133418746167",
    "DeleteDate": "2020-05-20T07:52:35Z",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "PendingDeletion",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "c23ef950-3622-46ac-92ff-89205c4ae4ff"
}
```

Cancel the scheduled deletion of a CMK

Call the `CancelKeyDeletion` operation to cancel the scheduled deletion of a CMK.

```
./aliyun --skip-secure-verify --region cn-qingdao --endpoint kms.example.com --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms CancelKeyDeletion --KeyId 769472ec-07d6-47e8-a32a-bc163f66****
```

After you cancel the scheduled deletion of a CMK, you can call the `DescribeKey` operation to check whether the CMK state becomes `Enabled`.

Sample success responses

```
{
  "KeyMetadata": {
    "Arn": "acs:kms:cn-qingdao-env25-d01:133418746167:example.com:key/769472ec-07d6-47e8-a32a-bc163f66",
    "AutomaticRotation": "Disabled",
    "CreationDate": "2020-05-13T07:45:52Z",
    "Creator": "133418746167",
    "DeleteDate": "",
    "Description": "this is a test key",
    "KeyId": "769472ec-07d6-47e8-a32a-bc163f66",
    "KeySpec": "Aliyun_AES_256",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT/DECRYPT",
    "LastRotationDate": "2020-05-13T07:45:52Z",
    "MaterialExpireTime": "",
    "Origin": "Aliyun_KMS",
    "PrimaryKeyVersion": "0b0b989b-7601-4e54-aa7d-5ae280ba",
    "ProtectionLevel": "SOFTWARE"
  },
  "RequestId": "8c931766-ff2d-45b0-a829-981b2d9b49ad"
}
```

29.3. Use RAM for access control

Key Management Service (KMS) uses Resource Access Management (RAM) to control access to resources. This topic describes the resource types, actions, and policy conditions in KMS.

Alibaba Cloud accounts have full operation permissions on their own resources. RAM users and roles are granted operation permissions on resources based on RAM authorization.

Resource types in KMS

The following table lists all resource types and corresponding Alibaba Cloud Resource Names (ARNs) in KMS. They can be used in the `Resource` parameter of a RAM policy.

Resource type	ARN
Key container	acs:kms:\${region}:\${account}:key
Secret container	acs:kms:\${region}:\${account}:secret

Resource type	ARN
Alias container	acs:kms:\${region}:\${account}:alias
Key	acs:kms:\${region}:\${account}:key/\${key-id}
Secret	acs:kms:\${region}:\${account}:secret/\${secret-name}
Alias	acs:kms:\${region}:\${account}:alias/\${alias-name}

Actions defined in KMS

KMS defines actions used in RAM policies for each API operation that requires access control. Actions must be in the `kms:${api-name}` format.

 **Note** The DescribeRegions operation requires no access control. It can be called by Alibaba Cloud accounts, RAM users, or RAM roles after they pass RAM authentication.

The following table lists the relationship between KMS API operations, RAM actions, and resource types.

- Key API operations

Operation	Action	Resource type
ListKeys	kms:ListKeys	Key container
CreateKey	kms:CreateKey	Key container
DescribeKey	kms:DescribeKey	Key
UpdateKeyDescription	kms:UpdateKeyDescription	Key
EnableKey	kms:EnableKey	Key
DisableKey	kms:DisableKey	Key
ScheduleKeyDeletion	kms:ScheduleKeyDeletion	Key
CancelKeyDeletion	kms:CancelKeyDeletion	Key
GetParametersForImport	kms:GetParametersForImport	Key
ImportKeyMaterial	kms:ImportKeyMaterial	Key
DeleteKeyMaterial	kms>DeleteKeyMaterial	Key
ListAliases	kms:ListAliases	Alias container
CreateAlias	kms:CreateAlias	Alias and key
UpdateAlias	kms:UpdateAlias	Alias and key
DeleteAlias	kms>DeleteAlias	Alias and key
ListAliasesByKeyId	kms:ListAliasesByKeyId	Key
CreateKeyVersion	kms:CreateKeyVersion	Key
DescribeKeyVersion	kms:DescribeKeyVersion	Key
ListKeyVersions	kms:ListKeyVersions	Key

Operation	Action	Resource type
UpdateRotationPolicy	kms:UpdateRotationPolicy	Key
Encrypt	kms:Encrypt	Key
Decrypt	kms:Decrypt	Key
ReEncrypt	<ul style="list-style-type: none"> ◦ kms:ReEncryptFrom ◦ kms:ReEncryptTo ◦ kms:ReEncrypt* 	Key
GenerateDataKey	kms:GenerateDataKey	Key
GenerateDataKeyWithoutPlaintext	kms:GenerateDataKeyWithoutPlaintext	Key
ExportDataKey	kms:ExportDataKey	Key
GenerateAndExportDataKey	kms:GenerateAndExportDataKey	Key
AsymmetricSign	kms:AsymmetricSign	Key
AsymmetricVerify	kms:AsymmetricVerify	Key
AsymmetricEncrypt	kms:AsymmetricEncrypt	Key
AsymmetricDecrypt	kms:AsymmetricDecrypt	Key
GetPublicKey	kms:GetPublicKey	Key

- Secrets Manager API operations

Operation	Action	Resource type
CreateSecret	kms:CreateSecret	Secret container
ListSecrets	kms:ListSecrets	Secret container
DescribeSecret	kms:DescribeSecret	Secret
DeleteSecret	kms>DeleteSecret	Secret
UpdateSecret	kms:UpdateSecret	Secret
RestoreSecret	kms:RestoreSecret	Secret
GetSecretValue	kms:GetSecretValue	Secret
PutSecretValue	kms:PutSecretValue	Secret
ListSecretVersionIds	kms:ListSecretVersionIds	Secret
UpdateSecretVersionStage	kms:UpdateSecretVersionStage	Secret
GetRandomPassword	kms:GetRandomPassword	None

- Tag management API operations

Operation	Action	Resource type
ListResourceTags	kms:ListResourceTags	Key or secret
UntagResource	kms:UntagResource	Key or secret
TagResource	kms:TagResource	Key or secret

Policy conditions in KMS

You can add conditions to RAM policies to control access to KMS. RAM authentication will be successful only when the specified conditions are met. For example, you can use `acs:CurrentTime` to control the time period when a RAM policy is valid.

In addition to global conditions, you can use tags as filters to restrict the use of cryptographic API operations such as Encrypt, Decrypt, and GenerateDataKey. Filters must be in the `kms:tag/${tag-key}` format.

RAM policy examples

- A RAM policy that allows users to access all KMS resources

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

- A RAM policy that allows users to only query keys, aliases, and key usage permissions

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*", "kms:Describe*",
        "kms:Encrypt", "kms:Decrypt", "kms:GenerateDataKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

- A RAM policy that allows users to use keys that contain the following tag to perform cryptographic operations:
 - Tag key: Project
 - Tag value: Apollo

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt", "kms:Decrypt", "kms:GenerateDataKey"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEqualsIgnoreCase": {
          "kms:tag/Project": [
            "Apollo"
          ]
        }
      }
    }
  ]
}
```

29.4. Use aliases

Aliases are optional to CMKs.

Aliases must be unique in a region for each Alibaba Cloud account. An Alibaba Cloud account can have identical aliases in different regions. An alias can be bound to only one CMK in a region, but a CMK can have multiple aliases.

Although aliases are bound to CMKs, aliases are resources independent of CMKs. Take note of the following points about aliases:

- You can call the UpdateAlias operation to bind an alias to a different CMK. This operation does not affect the CMK.
- If you delete an alias, the CMK to which the alias is bound is not deleted.
- RAM users must be authorized before they can perform operations on an alias. For more information, see [Use RAM for access control](#).
- Aliases cannot be modified. To change the alias of a CMK, you must delete the old alias and create a new one for the CMK.

You can replace the CMK ID in the request parameters for the following API operations with an alias that is bound to the CMK:

- DescribeKey
- Encrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext

 **Note** For more information about the API operations, see *Developer Guide* in the manual *API referencetopic*.

To specify an alias instead of a CMK ID in the request parameters for the preceding operations, a RAM user must have the relevant permissions on the CMK. The RAM user does not need to have permissions on the alias.

You can perform the following alias-related operations:

- [Create an alias](#)
- [Bind an alias to a different CMK](#)
- [Delete an alias](#)
- [Query all aliases](#)
- [Query aliases bound to a specific CMK](#)

When you use an alias, you must make sure that the alias is complete. Example:

```
//A complete alias must have the alias/ prefix.  
alias/example
```

Create an alias

- An alias must contain the `alias/` prefix. An alias (excluding the prefix) can contain letters, digits, underscores (`_`), hyphens (`-`), and forward slashes (`/`). An alias (excluding the prefix) must be 1 to 255 characters in length.
- To create an alias, a RAM user must have permissions on both the alias and the CMK to which the alias is bound.
- Creating a new alias for a CMK does not affect the existing aliases of the CMK.
- You can call the CreateAlias operation to create an alias.

```
//Grant RAM user 123456 the permissions to create the alias/example alias for the 08ec3bb9-034f-485b-b1cd-3459baa8
**** CMK.
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateAlias"
      ],
      "Resource": [
        "acs:kms:cn-hangzhou:123456:key/08ec3bb9-034f-485b-b1cd-3459baa8****",
        "acs:kms:cn-hangzhou:123456:alias/example"
      ]
    }
  ]
}

//Create an alias.
aliyun kms CreateAlias --KeyId 08ec3bb9-034f-485b-b1cd-3459baa8**** --AliasName alias/example
```

Bind an alias to a different CMK

- You can call the UpdateAlias operation to bind an existing alias to a different CMK.
- To bind an alias to a different CMK, a RAM user must have permissions on the alias, the CMK to which the alias is currently bound, and the CMK to which you want to bind the alias.

```
//Grant RAM user 123456 the permissions to bind the alias/example alias to the 127d2f84-ee5f-4f4d-9d41-dbc1aca2878  
8 CMK. The alias is originally bound to the 08ec3bb9-034f-485b-b1cd-3459baa8**** CMK.
```

```
{  
  "Version": "1",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:UpdateAlias"  
      ],  
      "Resource": [  
        "acs:kms:cn-hangzhou:123456:key/08ec3bb9-034f-485b-b1cd-3459baa8****",  
        "acs:kms:cn-hangzhou:123456:key/127d2f84-ee5f-4f4d-9d41-dbc1aca2****",  
        "acs:kms:cn-hangzhou:123456:alias/example"  
      ]  
    }  
  ]  
}
```

```
//Bind an alias to a different CMK.
```

```
aliyun kms UpdateAlias --AliasName alias/example --KeyId 127d2f84-ee5f-4f4d-9d41-dbc1aca2****
```

Delete an alias

- You can call the DeleteAlias operation to delete an alias. Deleting an alias does not affect the CMK to which the alias is bound.
- To delete an alias, a RAM user must have permissions on both the alias and the CMK to which the alias is bound.

```
//Grant RAM user 123456 the permissions to delete the alias/example alias of the 127d2f84-ee5f-4f4d-9d41-dbc1aca2**
** CMK.
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DeleteAlias"
      ],
      "Resource": [
        "acs:kms:cn-hangzhou:123456:key/127d2f84-ee5f-4f4d-9d41-dbc1aca2****",
        "acs:kms:cn-hangzhou:123456:alias/example"
      ]
    }
  ]
}

//Delete an alias.
aliyun kms DeleteAlias --AliasName alias/example
```

Query all aliases

- You can call the ListAliases operation to query all aliases under your Alibaba Cloud account in the current region.
- To query all aliases, a RAM user must have permissions on alias resources.

```
//Grant RAM user 123456 the permissions to query all aliases.
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListAliases"
      ],
      "Resource": [
        "acs:kms:cn-hangzhou:123456:alias"
      ]
    }
  ]
}

//Query all aliases.
aliyun kms ListAliases
```

Query aliases bound to a specific CMK

- You can call the ListAliasesByKeyId operation to query all aliases bound to a specific CMK.
- To query the aliases bound to a specific CMK, a RAM user must have permissions on the CMK.

```
//Grant RAM user 123456 the permission to query all aliases bound to the 127d2f84-ee5f-4f4d-9d41-dbc1aca2**** CMK.
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DeleteAlias"
      ],
      "Resource": [
        "acs:kms:cn-hangzhou:123456:key/127d2f84-ee5f-4f4d-9d41-dbc1aca2****"
      ]
    }
  ]
}

//Query the aliases bound to a specific CMK.
aliyun kms ListAliasesByKeyId --KeyId 127d2f84-ee5f-4f4d-9d41-dbc1aca2****
```

29.5. CMK overview

You can host different types of customer master keys (CMKs) in Key Management Service (KMS) based on your business requirements. For example, you can use a CMK to encrypt and decrypt data. You can also use a CMK to generate and verify a signature.

Key-based cryptographic algorithms

KMS allows you to use various algorithms to support cryptographic operations. The algorithms are classified into two types: symmetric key algorithms and asymmetric key algorithms, as listed in the following table.

Algorithm class	Algorithm subclass	Support encryption and decryption	Support signature generation and verification
Symmetric key algorithm	AES	Yes	No
Symmetric key algorithm	SM4 ^{Note}	Yes	No
Asymmetric key algorithm	RSA	Yes	Yes
Asymmetric key algorithm	ECC	No	Yes
Asymmetric key algorithm	SM2 ^{Note}	Yes	Yes

Symmetric keys are mainly used to encrypt or decrypt data. If you do not specify the KeySpec parameter during key creation, KMS creates a symmetric key. You can call the Encrypt or Decrypt operation to encrypt or decrypt data without the need to obtain the plaintext of a symmetric key. For more information, see [Overview of symmetric encryption](#).

Asymmetric keys can be used to encrypt data, decrypt data, generate a signature, or verify a signature. An asymmetric CMK in KMS consists of a public key and a private key, which are cryptographically related to each other. The public key can be made available for anyone to use, but the private key must be kept secure. To keep private keys secure, KMS does not provide an API operation for you to export the private key of an asymmetric key pair. You can use a private key to decrypt data or generate a signature by calling the related operations. Anyone with a public key can use it to encrypt data or verify the signature generated by the corresponding private key. For more information, see [Overview of asymmetric keys](#).

 **Note**

- Only hardware security modules (HSMs) support SM2 and SM4 keys.
- Only KMS of the Advanced edition supports Rivest-Shamir-Adleman (RSA) and elliptic-curve cryptography (ECC) keys.

Protection levels

KMS provides the Managed HSM feature. You can set the protection level of your CMK to HSM to host the CMK in an HSM. Managed HSM uses HSMs as dedicated hardware to safeguard keys. For a CMK whose protection level is HSM, the plaintext of its key material is stored only inside an HSM. KMS calls an HSM-related API operation to perform cryptographic operations. During the operations, no one can have access to the plaintext of the key material. The plaintext of the key material cannot be exported from the HSM.

Key managers

In most cases, you are the manager of your CMK in KMS, and its Creator attribute is set to the ID of your Alibaba Cloud account.

Alibaba Cloud services that are integrated with KMS can implement server-side encryption. In this scenario, an Alibaba Cloud service can automatically host an encryption key in KMS to encrypt and protect your data. This makes it easier for you to use the entry-level data encryption features and reduces your overhead for key lifecycle and permission management. These service-managed keys are called service keys. To facilitate identification, KMS sets the Creator attribute of the service key hosted by an Alibaba Cloud service to the code of this service and assigns an alias in the format of `acs/<Code of the Alibaba Cloud service>` to the service key. For example, the Creator attribute of the service key hosted by Alibaba Cloud Object Storage Service (OSS) is set to OSS and alias `acs/oss` is assigned to the service key.

29.6. Use symmetric keys

29.6.1. Overview of symmetric encryption

This topic describes symmetric encryption, which is the most commonly used data encryption method. KMS provides easy-to-use API operations that allow you to encrypt and decrypt data on the cloud.

 **Note** For more information about the API operations, see API reference in *Developer Guide*.

If you do not specify the KeySpec parameter during key creation, KMS creates a symmetric key. KMS supports popular symmetric key algorithms and provides high-level data security by using strong cryptography.

Classification of symmetric keys

KMS supports the following two types of symmetric keys:

-
- SM4

KMS supports the commercial symmetric algorithm SM4 standardized by China National Standards. If SM4 is used, set the KeySpec parameter to `Aliyun_SM4`. The Encrypt API operation uses Galois/Counter Mode (GCM) for encryption.

Encryption and decryption features

When you call API operations to encrypt data or data keys, you only need to specify a CMK ID or alias. KMS uses the specified CMK for encryption and returns ciphertext. When you call the Decrypt API operation, you only need to specify the ciphertext that you want to decrypt. You do not need to specify a CMK. The Decrypt operation is only available for you to decrypt the ciphertext that is generated by calling the Encrypt, GenerateDataKey, or GenerateDataKeyWithoutPlaintext operation.

AAD

Symmetric keys of KMS use GCM for block ciphers. You can use additional authenticated data (AAD) to provide supplemental protection for the integrity of encrypted data. By encapsulating AAD, KMS enables you to easily customize authentication data. For more information, see [Encryption Context](#).

Envelope encryption

With the GenerateDataKey and GenerateDataKeyWithoutPlaintext API operations, KMS can generate a two-level key hierarchy to accelerate envelope encryption.

For more information about envelope encryption, see the Envelope encryption topic of *Features* in *Technical White Paper*.

Rotation of symmetric keys

Each symmetric CMK that is generated in KMS supports multiple key versions. KMS automatically rotates CMKs by generating new key versions. You can customize the key rotation policy.

If a CMK has multiple versions, the latest version of the CMK is used to encrypt data or data keys in the Encrypt, GenerateDataKey, and GenerateDataKeyWithoutPlaintext operations. When you call the Decrypt operation, you do not need to specify a CMK ID or key version ID. KMS automatically identifies the CMK and its key version with which the corresponding data or data key is encrypted. Then KMS uses the key material of the identified key version to decrypt the ciphertext.

KMS rotates a CMK by generating a new version of the CMK. After a rotation is complete, KMS automatically uses the new key version to encrypt data or data keys. However, the earlier key version is still available for decrypting the ciphertext generated before the rotation. For more information, see [Automatic key rotation](#).

BYOK

KMS allows you to encrypt your data on the cloud by using the Bring Your Own Key (BYOK) feature. This feature helps you meet stringent security and compliance requirements. We recommend that you use Managed HSM to protect your keys. You can import your key material into a CMK whose protection level is HSM. Keys in a managed HSM can only be destroyed, and their plaintext cannot be exported. For more information, see [Overview of symmetric encryption](#).

29.6.2. EncryptionContext

EncryptionContext is a JSON string that can be used in KMS API operations, such as Encrypt, GenerateDataKey, and Decrypt.

Function of EncryptionContext

EncryptionContext is a JSON string. It must be in the string-string format and is used to ensure data integrity.

If this parameter is specified during encryption, you must specify equivalent EncryptionContext for decryption. Encryption is involved in Encrypt and GenerateDataKey, and decryption is involved in Decrypt.

EncryptionContext is related to decryption, but is not included in ciphertext, which corresponds to CipherBlob.

Valid values of EncryptionContext

A valid value of EncryptionContext is a JSON string of up to 8,192 characters only in the string-string format. When you specify EncryptionContext for an API operation, consider the escape characters.

Example of valid EncryptionContext

```

{"ValidKey": "ValidValue"}
{"Key1": "Value1", "Key2": "Value2"}

```

(Partial) Example of invalid EncryptionContext

```

[{"Key": "Value"}] // JSON array
{"Key": 12345} // String-int
{"Key": ["value1", "value2"]} // String-array

```

Equivalent EncryptionContext

In essence, EncryptionContext is a map or hash table in the string-string format. When EncryptionContext is used as a parameter, make sure that the key-value pairs indicated by the JSON string match. This allows you to obtain equivalent EncryptionContext. You can use EncryptionContext that is equivalent to EncryptionContext that you entered during encryption to decrypt ciphertext, rather than retaining the original string.

Example of equivalent EncryptionContext

```

{"Key1": "Value1", "Key2": "Value2"} is equivalent to {"Key2": "Value2", "Key1": "Value1"}.

```

29.6.3. Import and delete key material

This topic describes how to import and delete external key material.

Context

CMKs are basic resources of KMS. A CMK is composed of a key ID, basic metadata (such as key state), and key material that is used to encrypt and decrypt data. When you call the CreateKey operation to create a CMK, you can set the Origin parameter to `Aliyun_KMS` (default value). In this case, KMS generates key material. If you set the Origin parameter to `EXTERNAL`, you must import external key material to the CMK. In this case, KMS does not create key material. You can call the DescribeKey operation to check the key material source of an existing CMK.

 **Note** For more information about the API operations, see *API reference* in *Developer Guide*.

- If the value of Origin in KeyMetadata is `Aliyun_KMS`, the key material is generated by KMS. In this case, the CMK is considered to be a **normal key**.
- If the value of Origin is `EXTERNAL`, the key material is imported from an external source. In this case, the CMK is considered to be an **external key**.

Before you import external key material, take note of the following points:

- Make sure that the source of randomness from which the key material is generated meets security requirements.
- Make sure that the key material is reliable.
 - KMS ensures the high availability of imported key material. However, it cannot ensure that the imported key material has the same reliability as the key material generated by KMS.
 - You can call the DeleteKeyMaterial operation to delete the key material that you have imported. You can also set an expiration period to enable KMS to automatically delete the key material after the expiration period ends. The CMK is not deleted. To delete key material generated by KMS, you can only call the ScheduleKeyDeletion operation to specify a waiting period of 7 to 30 days for deleting the CMK. The key material is deleted with the relevant CMK after the waiting period ends.

- After you delete the imported key material, you can re-import the same key material to make the relevant CMK available again. Therefore, we recommend that you save a copy of the key material.
- Key material is unique for each CMK. When you import key material into a CMK, the CMK is associated with that key material. Even after the key material expires or is deleted, you cannot import different key material into that CMK. If you need to rotate a CMK that uses external key material, you must create a new CMK and then import new key material.
- CMKs are independent. You cannot use a CMK to decrypt data that is encrypted by using another CMK, even if the two CMKs use the same key material.
- The key material to be imported must be a 256-bit symmetric key.

Import key material

1. Create an external key. You can use one of the following methods to create an external key:

- Method 1: Log on to the KMS console and click Create Key. In the Advanced section of the Create Key dialog box, set Key Material Source to External. For more information, see [Create a CMK](#).
- Method 2: Call the CreateKey operation. Set Origin to *EXTERNAL*.

```
./aliyun --skip-secure-verify --region ${please-replace-your-region} --endpoint ${please-replace-your-endpoint} --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms CreateKey --Origin EXTERNAL --Description "External key"
```

2. Obtain parameters that are used to import key material. The parameters include a public key that is used to encrypt the key material and an import token. You can use one of the following methods to obtain the parameters:

- Method 1: Obtain the parameters in the KMS console.
- Method 2: Call the GetParametersForImport operation to obtain the parameters.

```
./aliyun --skip-secure-verify --region ${please-replace-your-region} --endpoint ${please-replace-your-endpoint} --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms GetParametersForImport --KeyId 1339cb7d-54d3-47e0-b595-c7d3dba8**** --WrappingAlgorithm RSAES_OAEP_SHA_1 --WrappingKeySpec RSA_2048
```

3. Import key material.

- You can import key material into an external key that never has key material. You can also reset the expiration time of key material or re-import key material that has expired or been deleted.
- Each import token is bound to a public key that is used to encrypt key material. A CMK is specified when an import token is generated. The import token can only be used to import key material into the specified CMK.
- The lifecycle of an import token is 24 hours. It can be used repeatedly within this period. After it expires, you must obtain a new import token and a new public key.
 - i. Use the public key to encrypt the key material. The public key is a 2048-bit Rivest-Shamir-Adleman (RSA) public key. The encryption algorithm must be consistent with that specified when you obtain the import parameters. Because the public key returned when you call the GetParametersForImport operation is Base64 encoded, you must first decode the public key. Currently, KMS supports the following encryption algorithms: RSAES_OAEP_SHA_1, RSAES_OAEP_SHA_256, and RSAES_PKCS1_V1_5.
 - ii. Base64 encode the encrypted key material.
 - iii. Call the ImportKeyMaterial operation to import the encoded key material and the import token to KMS.

```
./aliyun --skip-secure-verify --region ${please-replace-your-region} --endpoint ${please-replace-your-endpoint} --access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms ImportKeyMaterial --KeyId 1339cb7d-54d3-47e0-b595-c7d3dba8**** --EncryptedKeyMaterial xxx --ImportToken xxxx
```

Delete key material

After you import key material into an external key, you can use the external key just like a normal key. The only difference is that the key material of an external key may expire and can be deleted independently. After the key material of an external key expires or is deleted, the external key can no longer be used, and the ciphertext encrypted by using this external key cannot be decrypted. To enable the external key and related decryption again, you must re-import the same key material.

If an external key is in the Pending Deletion state before its key material expires or is deleted, the key state does not change after the key material expires or is deleted. Otherwise, the key state changes to Pending Import after the key material expires or is deleted.

You can use one of the following methods to delete key material:

- Method 1: Delete key material in the KMS console.
- Method 2: Call the DeleteKeyMaterial operation to delete key material.

```
./aliyun --skip-secure-verify --region ${please-replace-your-region} --endpoint ${please-replace-your-endpoint}
--access-key-id ${please-replace-your-access-key-id} --access-key-secret ${please-replace-your-access-key-secret} kms DeleteKeyMaterial --KeyId 1339cb7d-54d3-47e0-b595-c7d3dba8****
```

Use OpenSSL to encrypt and upload key material

1. Create an external key.
2. Generate key material. The key material must be a 256-bit symmetric key. In this example, OpenSSL is used to generate a 32-byte random number.

```
openssl rand -out KeyMaterial.bin 32
```

3. Obtain parameters that are used to import the key material.
4. Encrypt the key material.
 - i. Base64 decode the public key that is used to encrypt the key material.
 - ii. Use an encryption algorithm such as RSAES_OAEP_SHA_1 to encrypt the key material.
 - iii. Base64 encode the encrypted key material and save it as a text file.

```
openssl rand -out KeyMaterial.bin 32
openssl enc -d -base64 -A -in PublicKey_base64.txt -out PublicKey.bin
openssl rsautl -encrypt -in KeyMaterial.bin -oaep -inkey PublicKey.bin -keyform DER -pubin -out EncryptedKeyMaterial.bin
openssl enc -e -base64 -A -in EncryptedKeyMaterial.bin -out EncryptedKeyMaterial_base64.txt
```

- iv. Upload the encrypted key material and the import token.

Use SDK for Java to encrypt and upload key material.

```
//Use the latest KMS SDK for Java.
//KmsClient.java

import com.aliyuncs.kms.model.v20160120.*;
import com.aliyuncs.profile.DefaultProfile;

//KMS API encapsulation
public class KmsClient {
    DefaultAcsClient client;

    public KmsClient( String region_id, String ak, String secret) {
```

```

        DefaultProfile profile = DefaultProfile.getProfile(region_id, ak, secret);
        this.client = new DefaultAcsClient(profile);
    }

    public CreateKeyResponse createKey() throws Exception {
        CreateKeyRequest request = new CreateKeyRequest();
        request.setOrigin("EXTERNAL"); //Create an external key.
        return this.client.getAcsResponse(request);
    }
    //... Omitted. The remaining operations are the same as those in the API method.
}
//example.java
import com.aliyuncs.kms.model.v20160120.*;
import KmsClient
import java.security.KeyFactory;
import java.security.PublicKey;
import java.security.spec.MGF1ParameterSpec;
import javax.crypto.Cipher;
import javax.crypto.spec.OAEPParameterSpec;
import javax.crypto.spec.PSource.PSpecified;
import java.security.spec.X509EncodedKeySpec;
import java.util.Random;
import javax.xml.bind.DataConverter;

public class CreateAndImportExample {
    public static void main(String[] args) {
        String regionId = "cn-hangzhou";
        String accessKeyId = "**** Provide your AccessKeyId ****";
        String accessKeySecret = "**** Provide your AccessKeySecret ****";
        KmsClient kmsClient = new KmsClient(regionId,accessKeyId,accessKeySecret);
        //Create External Key
        try {
            CreateKeyResponse keyResponse = kmsClient.createKey();
            String keyId = keyResponse.KeyMetadata.getKeyId();
            //Generate a 32-bit random number.
            byte[] keyMaterial = new byte[32];
            new Random().nextBytes(keyMaterial);
            //Obtain parameters that are used to import the key material.
            GetParametersForImportResponse paramResponse = kmsClient.getParametersForImport(keyId,"RSAES_OAEP_
SHA_256");
            String importToken = paramResponse.getImportToken();
            String encryptPublicKey = paramResponse.getPublicKey();
            //Base64 decode the public key.
            byte[] publicKeyDer = DataConverter.parseBase64Binary(encryptPublicKey);
            //Use RSA to parse the public key.
            KeyFactory keyFact = KeyFactory.getInstance("RSA");
            X509EncodedKeySpec keySpec = new X509EncodedKeySpec(publicKeyDer);

```


Algorithm	Key type	Description	Purpose
SM2	EC_SM2	ECC defined by GB/T 32918	<ul style="list-style-type: none"> • Encrypt or decrypt data. • Generate a digital signature.

Data encryption

In data encryption, asymmetric keys are used to transmit sensitive information. The following steps describe a typical scenario:

1. An information receiver distributes a public key to a transmitter.
2. The transmitter uses the public key to encrypt sensitive information.
3. The transmitter sends the ciphertext generated from the sensitive information to the information receiver.
4. The information receiver uses the private key to decrypt the ciphertext.

The private key can be used only by the information receiver. This ensures that the plaintext of sensitive information cannot be intercepted and decrypted by unauthorized parties during transmission. This encryption method is widely used to exchange keys. For example, session keys are exchanged in Transport Layer Security (TLS) handshakes, and encryption keys are exported and imported between different hardware security modules (HSMs).

For more information, see [Encrypt and decrypt data by using an asymmetric CMK](#).

Digital signature

Asymmetric keys are also used to generate digital signatures. Private keys can be used to sign messages or information. Private keys are strictly protected and can be used only by trusted users to generate signatures. After a signature is generated, you can use the corresponding public key to verify the signature to achieve the following purposes:

- Verify data integrity. If the data does not match its signature, the data may be tampered with.
- Verify message authenticity. If a message does not match its signature, the message transmitter does not hold the private key.
- Provide non-repudiation for signatures. If the data matches its signature, the signer cannot deny this signature.

The following operations describe a typical signature verification scenario:

1. A signer sends a public key to a message receiver.
2. The signer uses the private key to sign data.
3. The signer sends the data and signature to the message receiver.
4. After receiving the data and signature, the message receiver uses the public key to verify the signature.

Digital signatures are widely used to defend against data tampering and authenticate identities.

For more information, see [Generate and verify a digital signature by using an asymmetric CMK](#).

Key version

KMS does not support automatic rotation of asymmetric CMKs. You can call the [CreateKeyVersion](#) operation to create a key version in a specific CMK and generate a new pair of public and private keys. If you use a new key version to generate a digital signature or encrypt data, you must also distribute the new version of the public key.

In addition, unlike symmetric CMKs, asymmetric CMKs do not have a primary key version. Therefore, to call the operations related to asymmetric keys in KMS, you must specify the CMK ID or CMK alias and a key version.

Public key operation

In most cases, you can call the `GetPublicKey` operation to obtain a public key and distribute it to users for encryption or verification. Then, the users can use cryptographic libraries such as OpenSSL and Java Cryptography Extension (JCE) on the business end to perform local calculation.

You can also call the `AsymmetricEncrypt` or `AsymmetricVerify` operation to perform public key operations. If you call these operations, KMS records the logs of the calls and allows you to use Resource Access Management (RAM) to put limits on the use of public keys. Compared with local calculation on the business end, KMS offers flexible functions that better suit your needs.

Private key operation

You can call only the `AsymmetricDecrypt` or `AsymmetricSign` operation to use a private key to decrypt data or generate a digital signature.

29.7.2. Encrypt and decrypt data by using an asymmetric CMK

This topic describes how to use an asymmetric customer master key (CMK) to encrypt and decrypt data in Alibaba Cloud CLI.

Asymmetric encryption generally includes the following steps:

1. An information receiver distributes a public key to a transmitter.
2. The transmitter uses the public key to encrypt sensitive information.
3. The transmitter sends the ciphertext generated from the sensitive information to the receiver.
4. The receiver uses the private key to decrypt the ciphertext.

Before you start

Call the `CreateKey` operation to create an asymmetric CMK in KMS. Set the `KeySpec` parameter to a desired key type and the `KeyUsage` parameter to `ENCRYPT/DECRYPT`.

The following code demonstrates how to create an RSA encryption key:

```
$ aliyun kms CreateKey --KeySpec=RSA_2048 --KeyUsage=ENCRYPT/DECRYPT --ProtectionLevel=HSM
```

Obtain the public key

1. Call the `GetPublicKey` operation to obtain the public key of the asymmetric key pair.

```
$ aliyun kms GetPublicKey --KeyId=**** --KeyVersionId=****
```

Sample success responses:

```
{
  "RequestId": "82c383eb-c377-4mf6-bxx8-81hkc1g5g7ab",
  "KeyId": "****",
  "KeyVersionId": "****",
  "PublicKey": "PublicKey-DataBlob"
}
```

2. Save the public key to the `rsa_publickey.pub` file. `PublicKey-DataBlob` is a placeholder. You must replace it with the obtained public key.

```
$ echo PublicKey-DataBlob > rsa_publickey.pub
```

Use the public key to encrypt data

1. Create a sample plaintext file plaintext-file.txt that contains "this is plaintext".

```
echo "this is plaintext" > plaintext-file.txt
```

2. Use OpenSSL to encrypt the file and write the obtained binary ciphertext into the plaintext-file.enc file.

```
openssl pkeyutl -encrypt -in plaintext-file.txt \
-inkey rsa_publickey.pub -pubin \
-pkeyopt rsa_padding_mode:oaep \
-pkeyopt rsa_oaep_md:sha256 \
-pkeyopt rsa_mgf1_md:sha256 \
-out plaintext-file.enc
```

Call the KMS API to decrypt data

You must call the KMS API and use the private key to decrypt data.

1. Before you transmit the encrypted data over the network, encode it in Base64.

```
$ openssl base64 -in plaintext-file.enc
```

The following Base64-encoded ciphertext is returned:

```
5kdCB06HHeAwgfH9ARY4/9Nv5vlpQ94GXZcmaC9FE59Aw8v8RYdozT6ggSbyZbi+
8STKVq9402MEfmUDmWjLuu0qgAZsCe5wU4JWHh1y84Qn6HT068j0qOy5X2Hilrjs
fCdetgtMtVorSgb3bbERk2RV67nHWrdkecNbUaz+6ik4ALZxv2uWrV62eQ9yUBYm
Jb956LbqnfWdCFxUSHH/qB5QCnLpijzvPmfNLZr653H4nF08gpZjnmLF4FJTU3i2
mGLzK4J3Rh/l7PQHiVMdc4hSnXosg68QmMVdZBGLK9/cD9SYngPDiirU7z0q7Git
dleloyCAUDFyuQC6a+SqzA==
```

2. Pass the Base64-encoded ciphertext to KMS to decrypt data.

```
aliyun kms AsymmetricDecrypt \
--KeyId **** \
--KeyVersionId **** \
--Algorithm RSAES_OAEP_SHA_256 \
--CiphertextBlob 5kdCB06HHeAwgfH9ARY4/9Nv5vlpQ94GXZcmaC9FE59Aw8v8RYdozT6ggSbyZbi+8STKVq9402MEfmUDm
wJLuu0qgAZsCe5wU4JWHh1y84Qn6HT068j0qOy5X2HilrjsfCdetgtMtVorSgb3bbERk2RV67nHWrdkecNbUaz+6ik4ALZxv2uWrV
62eQ9yUBYmJb956LbqnfWdCFxUSHH/qB5QCnLpijzvPmfNLZr653H4nF08gpZjnmLF4FJTU3i2mGLzK4J3Rh/l7PQHiVMdc4hSnXo
sg68QmMVdZBGLK9/cD9SYngPDiirU7z0q7GitdleloyCAUDFyuQC6a+SqzA==
```

Example output:

```
{
  "KeyId": "****",
  "KeyVersionId": "****",
  "Plaintext": "dGhpcyBpcyBwbGFpbnRleHQgDQo=",
  "RequestId": "6be7a8e4-35b9-4549-ad05-c5b1b535a22c"
}
```

3. Decode the returned Base64-encoded plaintext in Base64.

```
echo dGhpcyBpcyBwbGFpbnRleHQgDQo= | openssl base64 -d
```

The following decrypted plaintext is returned:

```
this is plaintext
```

29.7.3. Generate and verify a digital signature by using an asymmetric CMK

This topic uses Alibaba Cloud CLI as an example to describe how to use an asymmetric customer master key (CMK) to generate and verify a digital signature. You can also perform this operation by using the KMS SDK.

Asymmetric encryption generally includes the following steps:

1. A signer sends a public key to a receiver.
2. The signer uses the private key to sign data.
3. The signer sends the data and signature to the receiver.
4. After receiving the data and signature, the receiver uses the public key to verify the signature.

Before you start

Call the `CreateKey` operation to create an asymmetric CMK in KMS. Set the `KeySpec` parameter to a desired key type and the `KeyUsage` parameter to `SIGN/VERIFY`.

- Create an RSA signature key:

```
$ aliyun kms CreateKey --KeySpec=RSA_2048 --KeyUsage=SIGN/VERIFY --ProtectionLevel=HSM
```

- Create a NIST P-256 signature key:

```
$ aliyun kms CreateKey --KeySpec=EC_P256 --KeyUsage=SIGN/VERIFY --ProtectionLevel=HSM
```

- Create a secp256k1 signature key:

```
$ aliyun kms CreateKey --KeySpec=EC_P256K --KeyUsage=SIGN/VERIFY --ProtectionLevel=HSM
```

Preprocess signature: compute a message digest

Both RSA and ECC signature operations involve first computing the digest of an unsigned message and then signing the digest.

 **Note** The algorithm used to obtain a message digest must match the algorithm used to call KMS to compute a signature. For example, the `ECDSA_SHA_256` signature algorithm must be used in conjunction with the `SHA-256` digest algorithm. It does not support the `SHA-384` digest algorithm.

The following example uses the SHA-256 digest algorithm.

1. Save the message "this is message" that needs to be signed into the file message-file.txt:

```
$ echo "this is message" > message-file.txt
```

2. Compute the SHA-256 digest of the message and save the binary digest to the file message-sha256.bin:

```
$ openssl dgst -sha256 -binary -out message-sha256.bin message-file.txt
```

Call KMS to compute the signature

You must call the KMS API to compute the signature of a message with the private key.

1. Before you transmit the message digest over the network, encode it in Base64.

```
$ openssl base64 -in message-sha256.bin
```

The following Base64 encoded digest is returned:

```
hRP2cuRFSlfEoUXCGuPyi7kZr18VCTZeVOTw0jbUB6w=
```

2. Pass the Base64 encoded digest to KMS to generate a signature.

 **Note** The parameters passed and the results generated vary depending on key types and signature algorithms. Each signature result generated in the example is stored in a different file.

• RSASSA-PSS

For RSA keys, you can use the RSASSA-PSS signature algorithm and the SHA-256 digest algorithm to create a signature. Run the following command:

```
$ aliyun kms AsymmetricSign --KeyId=**** --KeyVersionId=**** \  
--Algorithm=RSA_PSS_SHA_256 --Digest=hRP2cu...  
{  
  "KeyId": "****",  
  "KeyVersionId": "****",  
  "Value": "J7xmdnZ...",  
  "RequestId": "70f78da9-c1b6-4119-9635-0ce4427cd424"  
}
```

Decode the signature value in Base64 and generate a binary signature. This signature is saved in the file rsa_pss_signature.bin:

```
$ echo J7xmdnZ... | openssl base64 -d -out rsa_pss_signature.bin
```

• RSASSA_PKCS1_V1_5

For RSA keys, you can use the RSASSA_PKCS1_V1_5 signature algorithm and the SHA-256 digest algorithm to create a signature. Run the following command:

```
$ aliyun kms AsymmetricSign --KeyId=**** --KeyVersionId=**** \
--Algorithm=RSA_PKCS1_SHA_256 --Digest=hrP2cu...
{
  "KeyId": "****",
  "KeyVersionId": "****",
  "Value": "qreBkH/u...",
  "RequestId": "4be57288-f477-4ecd-b7be-ad8688390fbc"
}
```

Decode the signature value in Base64 and generate a binary signature. This signature is saved in the file `rsa_pkcs1_signature.bin`:

```
echo qreBkH/u... | openssl base64 -d -out rsa_pkcs1_signature.bin
```

- **NIST P-256**

For NIST curve P-256, you can use the ECDSA signature algorithm and the SHA-256 digest signature to create a signature. Run the following command:

```
$ aliyun kms AsymmetricSign --KeyId=**** --KeyVersionId=**** \
--Algorithm=ECDSA_SHA_256 --Digest=hrP2cu...
{
  "KeyId": "****",
  "KeyVersionId": "****",
  "Value": "MEYCIQD33Y98...",
  "RequestId": "472d789c-d4be-4271-96bb-367f7f0f8ec3"
}
```

Decode the signature value in Base64 and generate a binary signature. This signature is saved in the file `ec_p256_signature.bin`:

```
echo MEYCIQD33Y98... | openssl base64 -d -out ec_p256_signature.bin
```

- **secp256k1**

For SECG curve secp256k1, you can use the ECDSA signature algorithm and the SHA-256 digest algorithm to create a signature. Run the following command:

```
$ aliyun kms AsymmetricSign --KeyId=**** --KeyVersionId=**** \
--Algorithm=ECDSA_SHA_256 --Digest=hrP2cu...
{
  "KeyId": "****",
  "KeyVersionId": "****",
  "Value": "MEYCIQDWuul...",
  "RequestId": "fe41abed-91e7-4069-9f6b-0048f5bf4de5"
}
```

Decode the signature Value in Base64 and generate a binary signature. This signature is saved in the file `ec_p256k_signature.bin`:

```
echo MEYCIQDWuul... | openssl base64 -d -out ec_p256k_signature.bin
```

Obtain the public key

Obtain the public key of the created asymmetric key pair from the KMS. The preceding example assumes that:

- The public key of the RSA key pair is saved to the file `rsa_publickey.pub`.
- The public key of the NIST P-256 key pair is saved to the file `ec_p256_publickey.pub`.
- The public key of the `secp256k1` key pair is saved to the file `ec_p256k_publickey.pub`.

Use the public key to verify the signature

Run the following command lines to verify the signature (the command varies depending on the algorithm used to generate the public key):

- **RSASSA-PSS**

```
$ openssl dgst \  
-verify rsa_publickey.pub \  
-sha256 \  
-sigopt rsa_padding_mode:pss \  
-sigopt rsa_pss_saltlen:-1 \  
-signature rsa_pss_signature.bin \  
message-file.txt
```

- **RSASSA_PKCS1_V1_5**

```
$ openssl dgst \  
-verify rsa_publickey.pub \  
-sha256 \  
-signature rsa_pkcs1_signature.bin \  
message-file.txt
```

- **NIST P-256**

```
$ openssl dgst \  
-verify ec_p256_publickey.pub \  
-sha256 \  
-signature ec_p256_signature.bin \  
message-file.txt
```

- **secp256k1**

```
$ openssl dgst \  
-verify ec_p256k_publickey.pub \  
-sha256 \  
-signature ec_p256k_signature.bin \  
message-file.txt
```

If the verification succeeds, the system displays the following message:

```
Verified OK
```

29.8. Key rotation

29.8.1. Overview

Keys are often used to protect data. The security of data is dependent on the security of its corresponding keys. You can use key versions and the periodic rotation mechanism to improve key security and implement security policies and best practices for data protection.

Security goals

You can use the periodic key rotation mechanism to:

- Reduce the amount of data encrypted by each key

The security of a key is inversely proportional to the amount of data encrypted by it. This amount is usually defined by the total bytes of data or the total number of messages that are encrypted by the same key. For example, National Institute of Standards and Technology (NIST) defines the secure lifecycle of a key in GCM mode as the total number of messages encrypted based on the key. The periodic key rotation mechanism enables each key to remain secure and minimize vulnerability to cryptanalytic attacks.

- Respond in advance to security events

In the early days of system design, key rotation was introduced as a routine O&M method. This provides the system with a method to handle security events when they occur, and complies with the fail early, fail often principle of software engineering. If key rotation is not executed until an emergency event has already occurred, the probability of system failure increases exponentially.

- Provide logical isolation of data

Encrypted data is isolated with each key rotation from other data encrypted using different keys. The impact of key-related security events can be identified quickly and preventive measures can be taken.

- Reduce the window of time to crack keys

Periodic rotation of encryption keys ensures that you can control and reduce the window of time for which the key and its encrypted data are vulnerable to being cracked. Attackers only have a limited period of time between rotation tasks during which they are able to crack the key. This practice greatly increases the security of your data against cryptanalytic attacks.

Regulatory compliance

The periodic key rotation mechanism facilitates compliance with various regulations, which include but are not limited to:

- Payment Card Industry Data Security Standard (PCI DSS)
- Cryptography-related industrial standards issued by State Cryptography Administration, such as GM/T 0051-2016
- Cryptography-related standards issued by NIST, such as NIST Publication 800-38D

29.8.2. Automatic key rotation

This topic describes how to configure automatic rotation of CMKs in KMS.

Key versions

A CMK may have multiple key versions. Each key version represents an independently generated key. Key versions of the same CMK do not have any cryptographic relation to each other. KMS automatically rotates CMKs by generating new key versions.

There are two types of key versions:

- Primary key versions
 - The primary key version of a CMK is an active encryption key. Each CMK has only one primary key version at any point in time.
 - When you call an encryption API operation such as `GenerateDataKey` or `Encrypt`, KMS uses the primary key version of a specified CMK to encrypt the target plaintext.
 - You can call the `DescribeKey` operation to view the `PrimaryKeyVersion` attribute.
- Non-primary key versions

- A non-primary key version of a CMK is an inactive encryption key. Each CMK can have any number of non-primary key versions.
- Each non-primary key version was a primary key version and acted as the active encryption key in the past.
- When a new primary key version is created, KMS does not delete or disable non-primary key versions because they will be used to decrypt data.

 **Note** When you call an encryption API operation, the primary key version of a specified CMK is used. When you call a decryption API operation, the key version that was used to encrypt the ciphertext is used.

You can generate a key version in either of the following ways:

- Create a CMK.

You can call the `CreateKey` operation to create a CMK. If you set the `Origin` parameter to `Aliyun_KMS`, KMS generates an initial key version and sets it as the primary key version.

- Execute an automatic rotation policy.

After you configure an automatic rotation policy, KMS executes the policy on a regular basis to generate new key versions.

Automatic key rotation

Configure a key rotation policy.

When you call the `CreateKey` operation to create a CMK, you can specify an automatic rotation policy for the CMK. You can call the `UpdateRotationPolicy` operation to update the current automatic rotation policy. When you call the API operations, you must configure the following parameters:

- `EnableAutomaticRotation`: specifies whether to enable automatic rotation.
- `RotationInterval`: indicates the time period for automatic rotation.

You can call the `DescribeKey` operation to view the configured automatic rotation policy. The following parameters are returned:

- `AutomaticRotation`: indicates whether automatic rotation is enabled. *Disabled* indicates that automatic rotation is not enabled. *Enabled* indicates that automatic rotation is enabled. *Suspended* indicates that KMS suspends the execution of automatic rotation although automatic rotation is enabled.
- `RotationInterval`: indicates the period for automatic rotation.

Execute an automatic rotation policy.

When automatic rotation is enabled, KMS calculates the time of the next rotation by using the following formula:

$$\${NextRotationTime} = \${LastRotationTime} + \${RotationInterval}$$

where:

- `LastRotationTime` : specifies the time the last key version is created. You can call the `DescribeKey` operation and check the `LastRotationDate` parameter to obtain the time.
- `NextRotationTime` : specifies the time KMS performs the next rotation task to create a new key version. You can call the `DescribeKey` operation and check the `NextRotationDate` parameter to obtain the time.

 **Notice** When you update the `RotationInterval` parameter of an automatic rotation policy, the value of `NextRotationTime` may be a point in time in the past. This does not affect the execution of the automatic rotation policy. If this situation occurs, KMS executes the automatic rotation policy immediately.

Impact of CMK status on automatic rotation

A new key version can be created for a CMK only if this CMK is in the *Enabled* state (the value of `KeyState`). You must take note of the following points:

- If a CMK is in the *Disabled* or *Pending Deletion* state, do not call the `UpdateRotationPolicy` operation to update its automatic rotation policy.
- If a CMK enters the *Disabled* or *Pending Deletion* state after you enable automatic rotation for it, KMS suspends the execution of automatic rotation. In this case, if you call the `DescribeKey` operation, the returned value of the `AutomaticRotation` parameter is *Suspended*. When the CMK enters the *Enabled* state again, its automatic rotation policy becomes active.

Limits

The following keys do not support multiple key versions:

- **Managed keys:** the default keys managed by KMS for specific cloud services. These keys belong to users of the cloud services and are used to provide basic encryption protection for data of the users.
- **Keys based on BYOK:** the keys that you have imported to KMS. For more information, see [Import and delete key material](#). The `Origin` attribute of these keys is *EXTERNAL*. KMS does not generate key material or initiate rotation tasks for these keys.

These two types of keys do not support version-based manual or automatic key rotation. A BYOK-based key does not have multiple versions in KMS. First, users have strong control over the persistence and lifecycle of BYOK-based keys. This makes management of the keys difficult and error-prone. For example, you must have on-premises key management facilities, data must be synchronized between on-premises and cloud facilities, and no grace period is provided for key material deletion on the cloud. The complexity of maintaining multiple versions of BYOK-based keys makes key management much more risky. Second, both primary and non-primary key versions may become unavailable at different points in time. For example, if key versions are deleted by KMS or imported again when they expire, it will be impossible to synchronize the CMKs and protected data or guarantee system integrity.

29.8.3. Manual key rotation

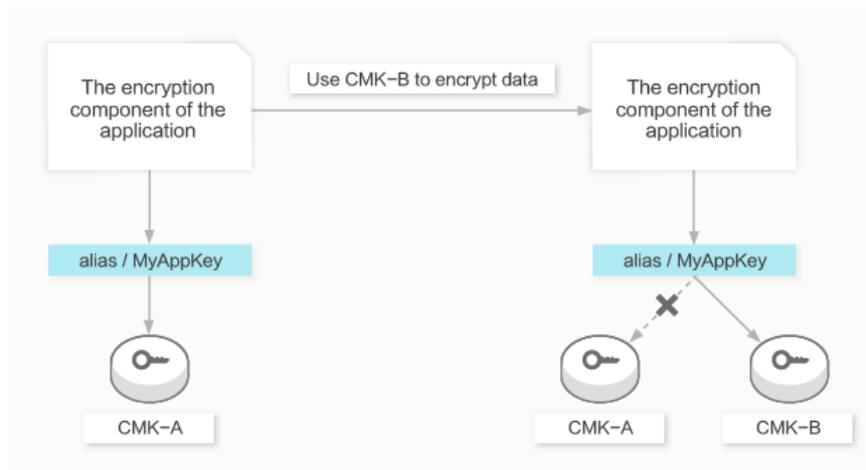
If your Customer Master Keys (CMKs) do not support version-based automatic rotation, you can manually rotate the CMKs. This is an alternative solution that does not depend on whether automatic key rotation is supported.

Custom data encryption scenario

On-premise or cloud applications can call the API operation to implement custom data encryption. Examples:

- Encrypt sensitive data such as ID card numbers, credit card information, and home addresses before writing it to databases
- Encrypt data at the client side before uploading it to OSS
- Encrypt service profiles that contain sensitive data and SSL key certificates such as application profiles

You can use the key alias feature to rotate encryption keys within applications. The ID and alias of the key are not required when you call the `Decrypt` API operation.



In this scenario, you must perform the following steps:

1. Initial configuration

- i. The administrator creates a CMK, whose ID is CMK-A.
- ii. The administrator binds the `alias/MyAppKey` alias to CMK-A.
- iii. When the application encryption module calls the Encrypt API operation, the value of the `KeyId` parameter is `alias/MyAppKey`. KMS finds that `alias/MyAppKey` is bound to CMK-A and then uses CMK-A to encrypt data.
- iv. When the application decryption module calls the Decrypt API operation, the `KeyId` parameter is not used. KMS uses the CMK used to encrypt the data to decrypt the data.

2. Manual rotation

- i. The administrator creates a CMK, whose ID is CMK-B.
- ii. The administrator calls the `UpdateAlias` API operation to bind the `alias/MyAppKey` alias to CMK-B.
- iii. When the application encryption module calls the Encrypt API operation, the value of the `KeyId` parameter is `alias/MyAppKey`. KMS finds that `alias/MyAppKey` is bound to CMK-B and uses CMK-B to encrypt data.
- iv. When the application decryption module calls the Decrypt API operation, the `KeyId` parameter is not used. KMS uses the CMK used to encrypt the data to decrypt the data.

Server encryption scenario

Other cloud services can encrypt their data by integrating KMS API operations. The following situations may occur in key rotation scenarios:

- Automatic rotation policies configured on KMS affect the server encryption of other cloud services

Cause: After CMK encryption is configured for cloud services, they will call the `GenerateDataKey` API operation of KMS to generate data keys. When KMS generates a new primary key version, cloud services use the new version to generate new data key. A typical example of such cloud services is OSS. In this situation, if you want to enable automatic key rotation, you cannot use service managed keys or BYOKs imported to KMS because they do not support automatic rotation.

- Automatic rotation policies configured on KMS do not affect server encryption of other cloud services

Cause: After CMK encryption is configured for cloud services, the services only call the `GenerateDataKey` API operation of KMS once to generate keys to encrypt specific resources. When KMS generates a new primary key version, cloud services will not use it. For example, when encrypting a cloud disk, ECS calls the `GenerateDataKey` API operation of KMS once to generate a volume encryption key. This key will not be updated again after it is created.

If automatic rotation policies configured on KMS do not affect server encryption of other cloud services or you want to rotate data keys when BYOKs are used, you can change configurations and copy data to achieve the same effect as key rotation. These methods depend on the features of different cloud services. For more information, see documentation of respective cloud services.

30. Log Service

30.1. What is Log Service?

Log Service (SLS) is a one-stop logging service developed by Alibaba Cloud that is widely used by Alibaba Group in big data scenarios. You can use Log Service to collect, query, and consume log data.

Without the need to invest in in-house data collection and processing systems. This enables you to focus on your business, improving business efficiency and helping your business to expand.

Log Service provides the following features:

- **Log collection:** Log Service allows you to collect events, binary logs, and text logs in real time by using multiple methods, such as Logtail and JavaScript.
- **Query and analysis:** Log Service allows you to query and analyze the collected log data and view analysis results on charts and dashboards.
- **Status alert:** Log Service can automatically run query statements at regular intervals after you create an alert task. If the query results meet the conditions of the alert task, Log Service sends an alert to the specified recipients in real time.
- **Real-time consumption:** Log Service provides real-time consumption interfaces through which log consumers can consume log data.

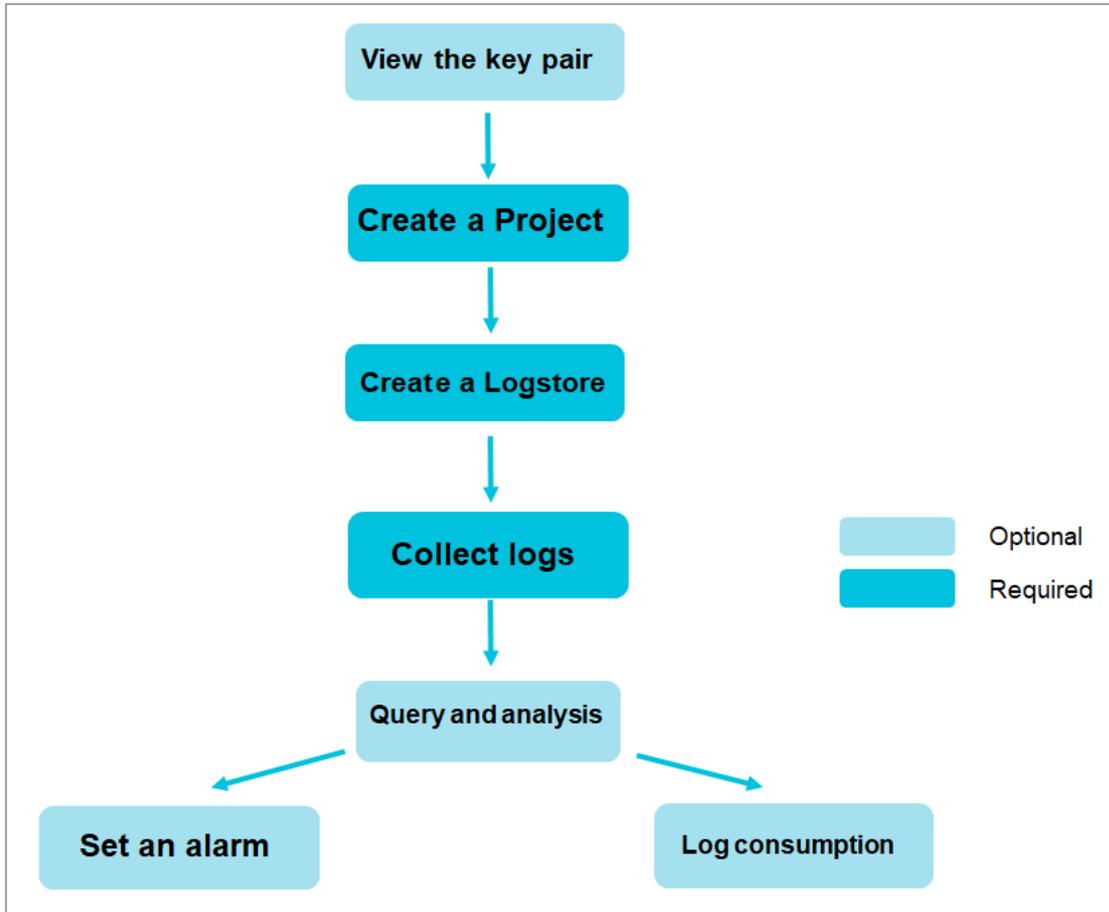
30.2. Quick start

30.2.1. Procedure

This topic provides the basic procedure to use Log Service. You can use this procedure to create projects, create Logstores, and collect log data.

The following figure shows the **Procedure**.

Procedure



1. **Optional. Obtain an AccessKey pair.**
 Before you can use Log Service through APIs or SDKs, you must have an AccessKey pair.
2. **Create a project.**
 Create a project in a specified region and add a description.
3. **Create a Logstore.**
 Create a Logstore for the project and specify the number of shards.
4. **Collect text logs**
 Select a method to collect log data based on your business requirements. Text log collection is used as an example.
5. **Enable the index feature and configure indexes for a Logstore,** and query and analyze logs.
 Log Service supports **real-time log query** and **analysis**. After you enable the indexing feature, you can query and analyze logs and configure **Overview** and **dashboards**.
6. **Configure alerts.**
 Log Service allows you to configure alerts based on log query results. Then, Log Service sends alerts by using multiple methods, such as a custom webhook.
7. **Consume logs in real time.**
 Log Service allows you to consume logs by using multiple methods, such as a **Spark Streaming client**, **Storm spout**, and **Flink connector**.

30.2.2. Log on to the Log Service console

This topic describes how to log on to the Log Service console.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Log Service**.
5. On the page that appears, select the organization and region, and then click **SLS**. The home page of the Log Service console is displayed.

30.2.3. Obtain an AccessKey pair

This topic describes how to obtain an AccessKey pair in the Apsara Stack Cloud Management (ASCM) console.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following operations:

1. Log on to the Apsara Stack Cloud Management (ASCM) console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the **Organizations** navigation tree, click the  icon on the right of the organization to be added.
5. Select **AccessKey**.
6. In the message that appears, view the AccessKey pair of the organization.

 **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations use the same AccessKey pair as their level-1 organization.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following operations:

1. Log on to the ASCM console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture and click **User Information**.
3. In the **Apsara Stack AccessKey Pair** section of the **User Information** page, view your AccessKey pair.

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.

The AccessKey pair including the AccessKey ID and AccessKey secret is the credential to for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.

Region	AccessKey ID	AccessKey Secret
cn-qingdao-env4b-d01	AKIAI44QH8D8DFQ13O9G5	Show

Note The AccessKey pair consists of an AccessKey ID and an AccessKey secret. They allow you to access Apsara Stack resources with full permissions for your account. You must keep them confidential.

30.2.4. Manage projects

This topic describes how to create, modify, and delete projects in the Log Service console.

Context

A project in Log Service is a resource management unit. The resources in each project are isolated from resources in other projects. We recommend that you store the log data of different applications in dedicated projects. You can manage Logstores, Logtail configurations, log sources, log data, and machine groups in a project. Each project provides an endpoint for you to access the resources.

A project provides the following features:

- Allows you to store log data from different sources in different Logstores of a project. You can use Log Service to collect log data from multiple sources such as business projects, products, and environments. You can then store the log data of each source in a separate Logstore. This simplifies the downstream processes such as the consuming, exporting, and indexing of log data. In addition, you can manage access permissions at the project level.
- Provides an endpoint for you to access the resources in the project. Log Service allocates an exclusive endpoint to each project. You can use the endpoint to read, write, and manage the log data in the project.

Create a project

Note

- You can create a project only by using the Log Service console.
- You can create up to 50 projects under each Apsara Stack tenant account.

1. Log on to the Log Service console.
2. In the Projects section, click Create Project.
3. Set the parameters based on your requirements. The following table describes the parameters.

Parameter	Description
Project Name	<p>The name of the project. The project name must be unique across all regions. Use the following naming conventions:</p> <ul style="list-style-type: none"> ◦ The name can contain lowercase letters, digits, and hyphens (-). ◦ The name must start and end with a lowercase letter or digit. ◦ The name must be 3 to 63 characters in length. <p>Note After a project is created, its name cannot be modified.</p>
Description	<p>The description of the project. After the project is created, the description is displayed in the Projects section. If you need to modify the description after the project is created, find the project in the Projects section, and click Edit in the Actions column. The description must be 0 to 64 characters in length and cannot contain the following characters: < > ' \ " \ \ .</p>

Parameter	Description
Region	<p>The region to which the project belongs. We recommend that you select a region that is closer to the log source.</p> <p>After a project is created, its region cannot be modified. This means that projects cannot be migrated across regions.</p>

4. Click **OK**.

Modify the description of a project

To modify the description of a project, perform the following steps:

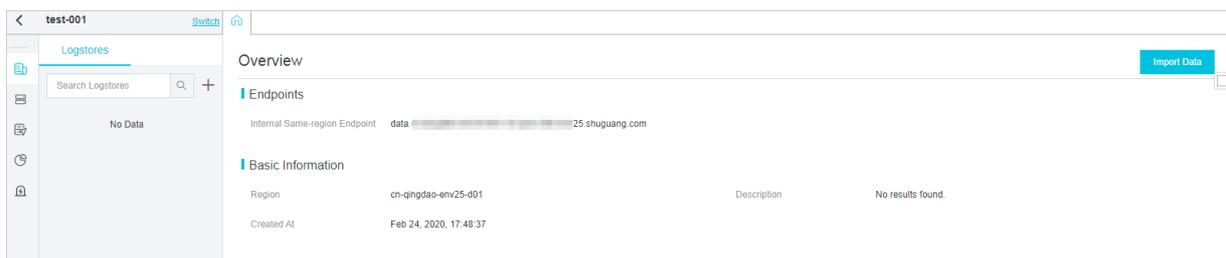
1. In the **Projects** section, find the project.
2. In the **Actions** column, click **Edit**.
3. In the **Modify Project** dialog box, modify the description of the project.

 **Note** You cannot modify the project name or region.

4. Click **OK**.

View the information of a project

To view the information of a project, click the project name in the **Projects** section. On the **Overview** page, you can view the project information such as the endpoint and region.



Delete a project

To delete a project, perform the following steps:

 **Warning** After you delete a project, all logs and configurations in the project are deleted and cannot be restored.

1. In the **Projects** section, find the project.
2. In the **Actions** column, click **Delete**.
3. In the dialog box that appears, select a reason for deletion. If you select **Other issues**, enter the reason in the text box.

Delete Project ×

 You cannot restore the project data after the project is deleted. Are you sure you want to delete the project?

Project Name: test-001

Reason for Deletion The project name is incorrect.

The region of the project is incorrect.

Business issue. Log analysis is no longer required.

The data in the project is for test and must be cleared.

Cost issue.

Do not know how to use Log Service.

Cannot import logs to the project.

Other issues.

4. Click OK.

30.2.5. Manage Logstores

This topic describes how to create, modify, and delete a Logstore in the Log Service console. A Logstore is a collection of resources inside a project. The log data in a Logstore is collected from the same source.

Context

You can create multiple Logstores in a project. We recommend that you create a Logstore for each type of application log.

Logstores provide the following features:

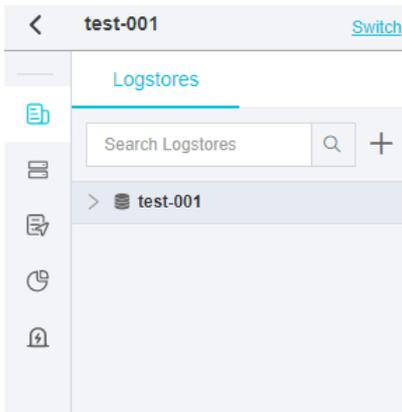
- Real-time log collection
- Log storage and real-time log consumption
- Log indexing and real-time log query

Create a Logstore

 **Note** You can create a maximum of 100 Logstores in each project.

1. [Log on to the Log Service console.](#)
2. In the **Projects** section, click the name of a project.

3. On the page that appears, click next to the search box.



4. In the Create Logstore pane, set the parameters and click OK.

Parameter	Description
Logstore Name	<p>The name of the Logstore. The Logstore name must be unique in the project to which the Logstore belongs.</p> <ul style="list-style-type: none"> ◦ The name can contain lowercase letters, digits, hyphens (-), and underscores (_). ◦ The name must start and end with a lowercase letter or digit. ◦ The name must be 3 to 63 characters in length. <p>Note After a Logstore is created, its name cannot be modified.</p>
WebTracking	Specifies whether to enable the WebTracking feature for the Logstore. You can use WebTracking to collect the log data of HTML websites, HTML5 websites, iOS apps, or Android apps and forward the data to Log Service. This feature is not enabled by default.
Permanent Storage	Specifies whether to permanently store the log data in the Logstore. If you disable this feature, you must specify a retention period for log data.
Data Retention Period	The duration for which log data is stored in the Logstore after the log data is collected. Unit: days. Valid values: 1 to 3000. When this period expires, the log data is deleted.
Shards	The number of shards in the Logstore. You can divide a Logstore into 1 to 10 shards.
Automatic Sharding	Specifies whether to enable the automatic sharding feature. This feature is not enabled by default. If you enable this feature, Log Service automatically splits shards in the Logstore when the data transfer exceeds the capacity of the existing shards.
Maximum Shards	The maximum number of shards. This parameter is required if you enable the automatic sharding feature. Maximum value: 64.

Delete : test-001



You cannot restore the data that has been deleted. Are you sure to delete the data?

OK

Cancel

30.2.6. Manage shards

This topic describes how to split, merge, and delete shards in the Log Service console. Logs are stored on shards in a Logstore. Each Logstore can have multiple shards. When you create a Logstore, you must specify the number of shards in the Logstore. After a Logstore is created, you can split or merge the shards.

Hash key

Log Service uses 128-bit MD5 hashes as the hash key of a Logstore. The entire MD5 hash range is [00000000000000000000000000000000,ffffffffffffffffffffffffffff). The hash key range of a Logstore falls within the entire MD5 hash range. When you create a Logstore, you must specify the number (N) of shards in the Logstore. The hash key range of the Logstore is evenly divided into N parts. Each part is assigned to a shard.

The hash key range of a shard is a left-closed and right-open interval that is specified by the following parameters:

- **BeginKey:** the start of the hash key range. The value of this parameter is included in the range.
- **EndKey:** the end of the hash key range. The value of this parameter is excluded from the range.

If you split a shard, the hash key range of the shard is evenly split. If you merge two shards, the hash key ranges of the shards are also merged. A hash key range determines the scope of a shard. When you push log data to a Logstore, you can specify a hash key for the log data. Log Service then writes the log data to the shard whose hash key range includes the specified hash key. This is called the hash key mode. If you do not specify a hash key for log data, the load balancing mode is used and Log Service writes the log data to a random available shard. However, when you pull log data from a Logstore, you must specify the shard where the log data is stored.

For example, a Logstore is divided into four shards and the hash key range of the Logstore is [00,FF). [Example shards](#) lists the hash key range of each shard.

Example shards

Shard	Hash key range
Shard0	[00,40)
Shard1	[40,80)
Shard2	[80,C0)
Shard3	[C0,FF)

If you set the hash key of log data to 5F, Log Service writes the log data to shard 1 because the hash key range of shard 1 includes 5F. If you set the hash key to 8C, the log data is written to shard 2 because the hash key range of shard 2 includes 8C.

Read/write capacity

Each shard provides an identical read/write capacity. Therefore, the read/write capacity of a Logstore depends on the number of shards in the Logstore. We recommend that you adjust the capacity of a Logstore based on the data traffic. For a Logstore, if the data traffic exceeds the read/write capacity, you can split shards to increase the Logstore capacity. If the data traffic is much less than the read/write capacity, you can merge shards to reduce the Logstore capacity and save costs.

For example, a Logstore consists of two read/write shards and the shards provide a maximum write capacity of 10 MB/s. If log data is written to the Logstore at a rate of 14 MB/s, we recommend that you split one of the shards into two shards. However, if log data is written at a rate of 3 MB/s, you can merge the two shards because the capacity of one shard already meets the read/write requirements.

 **Note**

- If an API operation that writes data to a Logstore constantly returns 403 or 500 errors, you can check the data traffic metrics that are provided by Log Service and determine whether to split shards.
- If the data traffic of a Logstore exceeds the read/write capacity of the Logstore, Log Service provides the best possible service but does not guarantee the service quality.

Shard status

A shard can be in one of the following states:

- Read/write
- Read-only

After a shard is created, the default status of the shard is read/write. If you split or merge shards, the status of the original shards changes to read-only and the new shards are in the read/write state. You can write data to and read data from a read/write shard. However, you can only read data from a read-only shard and cannot write data to the shard.

If you need to split a shard in a Logstore, you must specify the ID of the shard and an MD5 hash. The shard must be in the read/write state. The MD5 hash must be greater than the value of the BeginKey parameter of the shard and less than the value of the EndKey parameter of the shard. After the shard is split, the Logstore has two more shards. The status of the original shard changes from read/write to read-only. You can consume the log data in the original shard but cannot write log data to the shard. The new shards are in the read/write state and are listed below the original shard. The hash key ranges of the new shards cover that of the original shard.

If you need to merge shards in a Logstore, you must specify a read/write shard. The shard cannot be the last read/write shard in the shard list. Log Service finds the shard whose hash key range follows the hash key range of the specified shard, and merges the two shards into a new shard. The status of the original shards changes from read/write to read-only. You can consume the log data in the original shards but cannot write log data to the shards. The new shard is in the read/write state. The hash key range of the new shard covers those of the original shards.

You can perform the following operations on shards in the Log Service console:

- Split a shard.
- Merge shards.

Split a shard

Each shard provides a write capacity of 5 MB/s and a read capacity of 10 MB/s. For a Logstore, if the data traffic exceeds the total read/write capacity of existing shards, we recommend that you split shards to increase the capacity.

1. [Log on to the Log Service console](#).
2. In the **Projects** section, click the name of a project.
3. On the page that appears, click the  icon next to the Logstore, and select **Modify** from the shortcut menu.
4. In the upper-right corner of the **Logstore Attributes** page, click **Modify**.

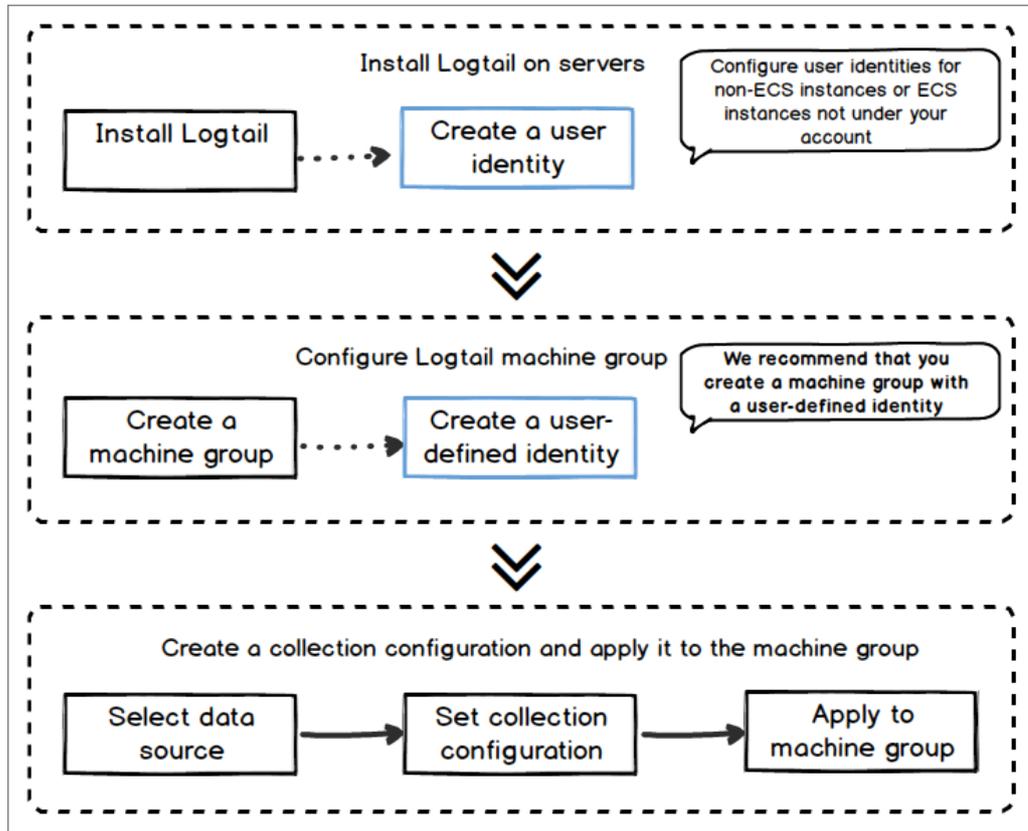
- Provides a comprehensive self-protection mechanism. To minimize the impact of Logtail on the performance of the related servers, Logtail limits the usage of CPU, memory, and network resources.

Processing capabilities and limits

For more information, see [Limits](#).

Configuration process

Configuration process



To collect logs from servers by using Logtail, follow these steps:

1. Install Logtail. For more information about how to install Logtail on a server from which you want to collect logs, see [Install Logtail in Linux](#) and [Install Logtail in Windows](#).
2. Log Service use server groups to manage all servers from which you want to collect logs by using Logtail. Log Service allows you to define server groups by using IP addresses or custom identifiers. You can create a server group as prompted when you apply Logtail configurations to server groups.
3. Create a Logtail configuration and apply it to the server group. For more information about how to create a Logtail configuration, see [Configure text log collection](#).

After the preceding process is complete, logs on the server are automatically collected and sent to the selected Logstore. However, historical logs are not collected. You can use the Log Service console, SDKs, or APIs to query these logs. Log Service allows you to view the status of log collection and check whether errors occur.

For more information, see [Collect logs by using Logtail](#).

Containers

- For information about Alibaba Cloud Container Service for Kubernetes or user-created Kubernetes clusters, see [Collect Kubernetes logs](#).
- For information about other user-created Docker clusters, see [Collect standard Docker logs](#).

Terms

- **Server group:** A server group contains one or more servers from which logs of a specific type are collected. You can apply Logtail configurations to a server group. This enables Log Service to collect logs from all servers in the server group. You can use the Log Service console to manage a sever group. For example, you can create, delete, add, or remove a server. Each server group can contain different versions of Windows servers or Linux servers.
- **Logtail:** Logtail is the agent that collects logs from the servers on which Logtail runs. For more information, see [Install Logtail in Linux](#) and [Install Logtail in Windows](#). After you install Logtail on a server, you must create a Logtail configuration and apply it to the server group to which the server belongs.
 - In Linux, Logtail is installed in the `/usr/local/ilogtail` directory. Logtail initiates two separate processes whose names start with `ilogtail`. One is a collection process and the other is a daemon process. The program running log is `/usr/local/ilogtail/ilogtail.LOG`.
 - In Windows, Logtail is installed in the `C:\Program Files\Alibaba\Logtail` (for 32-bit systems) or `C:\Program Files (x86)\Alibaba\Logtail` (for 64-bit systems) directory. You can choose Administrative Tools > Services to view the two Windows services generated from Logtail. One is LogtailWorker (log collection process) and the other is LogtailDaemon. The program running log is `logtail_*.log` in the installation directory.
- **Logtail configuration:** A Logtail configuration is a set of policies that are used by Logtail to collect logs. You can specify Logtail parameters such as the data source and collection mode. This allows you to customize log collection policies for all servers in a server group. A Logtail configuration determines how to collect a type of logs from a server, parse the logs, and send them to a specified Logstore. You can create a Logtail configuration for a Logstore in the Log Service console. This enables the Logstore to receive logs that are collected by using this Logtail configuration.

Features

Logtail provides the following features:

Feature	Description
Real-time log collection	<p>Logtail dynamically monitors log files and reads and parses incremental logs in real time. In most cases, logs are sent to Log Service within 3 seconds after they are generated.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note Logtail does not collect historical data. If Logtail reads a log later than 12 hours after the log was generated, Logtail drops the log.</p> </div>
Automatic log rotation	<p>Some applications rotate log files based on the file size or date. In the rotation process, the original log files are renamed and empty log files are created. For example, files such as <code>app.LOG.1</code> and <code>app.LOG.2</code> are generated for the <code>app.LOG</code> file after log rotation. You can specify the file (for example, <code>app.LOG</code>) to which collected logs are written. Logtail monitors the log rotation process to ensure that no logs are lost.</p>
Multiple data sources	<p>Logtail can collect text logs, syslogs, HTTP logs, and MySQL binlogs.</p>
Automatic exception handling	<p>If data transmission fails due to exceptions such as Log Service errors, network errors, or quota exhaustion, Logtail retries log collection based on the specific scenario. If the retry fails, Logtail writes the data to the local cache and resends the data after the exception no longer exists.</p>

Feature	Description
Flexible collection policy configuration	<p>You can create a Logtail configuration to specify how logs are collected from an ECS instance. You can select log directories and files by using exact match or wildcard match based on actual scenarios. You can also customize the extraction method of collected logs and the names of extracted fields. Log Service allows you to extract logs by using regular expressions.</p> <p>The log data models of Log Service require that each log has a precise timestamp. Logtail provides custom log time formats. This allows you to extract the required timestamps from log data of different formats.</p>
Automatic synchronization of Logtail configurations	After you create or update a Logtail configuration in the Log Service console, Logtail receives and validates the configuration within 3 minutes. No data loss occurs during the configuration update process.
Automatic upgrade	After you install Logtail on a server, Log Service manages the automatic upgrade of Logtail without manual intervention. No data loss occurs during the Logtail upgrade process.
Status monitoring	Logtail monitors its consumption of CPU and memory in real time. This prevents Logtail from excessively consuming your resources. If the resource consumption exceeds the limit, Logtail restarts to avoid affecting other services on the server. Logtail limits network traffic to avoid excessive bandwidth consumption.
Data transmission with a signature	<p>Logtail obtains your Alibaba Cloud AccessKey pair and uses it to sign all log data packets before they are sent. This prevents data tampering during transmission.</p> <p> Note Logtail obtains your Alibaba Cloud AccessKey pair over HTTPS to ensure the security of your AccessKey pair.</p>

Data collection reliability

During data collection, Logtail stores the collected checkpoints to the local directory on a regular basis. If an exception (for example, an unexpected server shutdown or a process crash) occurs, Logtail restarts and then collects data from the last recorded checkpoint to prevent data loss. Logtail runs based on the **resource limits** specified in the configuration file. If the usage of a resource exceeds the limit for more than 5 minutes, Logtail restarts. After the restart, duplicate data may be generated.

Logtail uses multiple internal mechanisms to improve log collection reliability. However, logs may be lost in the following conditions:

- Logtail is not running but logs are rotated multiple times.
- The log rotation rate is high, for example, one rotation per second.
- The log collection rate is lower than the log generation rate for a long period of time.

30.3.1.1.2. Log collection process of Logtail

This topic describes the process that Logtail uses to collect server logs. The process consists of the following steps: monitor files, read files, process logs, filter logs, aggregate logs, and send logs.

 **Note** After the Logtail configuration is applied to a machine group, unmodified logs on the servers in the machine group are considered as historical logs. Logtail does not collect historical logs in normal running mode. If you want to collect historical logs, see **Import historical logs**.

Monitor files

After you install Logtail on a server and create a Logtail configuration based on a data source, the Logtail configuration is delivered from Log Service to Logtail in real time. Then, Logtail starts to monitor files based on the Logtail configuration.

1. Logtail scans log directories and files that comply with the specified file naming rules based on the specified log path and maximum monitoring directory depth.

Logtail registers event monitoring and periodical polling for the directory from which Logtail collect logs. In Linux, **Inotify** is used. In Windows, **ReadDirectoryChangesW** is used. This method ensures the timeliness and stability of log collection.

2. If the compliant log files in the specified directory are not modified after the configuration is applied, Logtail does not collect these files. If modification events are generated for log files, Logtail triggers the collection process and reads these files.

Read files

After Logtail detects that a log file has been updated, Logtail reads the log file.

- If Logtail reads the log file for the first time, it checks the file size.
 - If the file size is less than 1 MB, Logtail reads the file from the beginning.
 - If the file size is greater than 1 MB, Logtail reads the file from the last 1 MB of data.
- If Logtail has read the file before, Logtail reads the file from the last checkpoint.
- Logtail can read up to 512 KB of data at a time. Therefore, you must limit the log size to 512 KB.

 **Notice** If you have changed the system time on your server, you must manually restart Logtail. Otherwise, the log time is incorrect and logs are dropped.

Process logs

When Logtail reads a log, it divides the log into multiple lines, parses the log, and sets the time field of the log.

- Divide each log into multiple lines

If you specifies **Regex to Match First Line Only** in the Logtail configuration, the log data read by Logtail at one time is divided into multiple lines based on the specified beginning of the line. If the beginning of the line is not specified, each data block is processed as a log.

- Parse each log

Logtail parses each log based on the Logtail configuration, such as regular expressions, delimiters, and JSON.

 **Notice** A complicated regular expression may lead to high CPU usage. Therefore, we recommend that you use an efficient regular expression.

- Handle parsing failures

Logtail determines how to handle parsing failures based on whether the **Drop Failed to Parse Logs** switch is turned on in the Logtail configuration.

- If the **Drop Failed to Parse Logs** is turned on, Logtail drops the logs that fail to be parsed and reports an error.
- If the feature switch is turned off, Logtail uploads the logs that fail to be parsed. In these logs, the key is set to **raw_log** and the value is set to the log content.

- Set the time field of a log

- If the time field of the log is not specified, the log time is the current parsing time.
- If the time field of the log is specified, the following operations are performed:
 - The log time is extracted from the parsed log fields if the difference between the log time and the current time is less than 12 hours.
 - The log is dropped and an error is reported if the difference between the log time and the current time is greater than 12 hours.

Filter logs

After Logtail process logs, it filters the logs based on the filter configuration.

- If the **Filter Configuration** is not specified, Logtail does not filter logs and proceed with the next step.
- If the **Filter Configuration** is specified, Logtail traverses and verifies all the fields of each log.
 - Logtail collects a log if the log matches the filter configuration.
 - Logtail does not collect a log if the log does not match the filter configuration.

Aggregate logs

Logtail sends the logs that match the filter configuration to Log Service. To reduce the number of network requests, Logtail caches the processed and filtered logs for a period of time. Then, Logtail aggregates and packages these logs before sending them to Log Service.

If one of the following conditions is met during caching, logs are immediately packaged and sent to Log Service.

- The log aggregation period exceeds three seconds.
- The number of aggregated logs exceeds 4,096.
- The total size of aggregated logs exceeds 512 KB.

Send logs

Logtail sends the aggregated logs to Log Service. You can set the `max_bytes_per_sec` and `send_request_concurrency` parameters in [Set Logtail startup parameters](#) to adjust the log data sending rate and the maximum concurrent requests. In this case, Logtail ensures that the sending rate and the concurrent requests do not exceed the limits.

If the log data fails to be sent, Logtail retries or stops the operation based on the error message.

Error	Description	Handling method
401	Logtail is not authorized to collect data.	Logtail drops the log packets.
404	The specified project or Logstore does not exist in the Logtail configuration.	Logtail drops the log packets.
403	The shard quota is exhausted.	Logtail waits for three seconds and retries.
500	A Log Service exception has occurred.	Logtail waits for three seconds and retries.
Network timeout	A network connection error has occurred.	Logtail waits for three seconds and retries.

30.3.1.1.3. Logtail configuration files and record files

This topic describes the basic configuration files and record files of Logtail. When Logtail is active, it uses a series of configuration files and generates record files.

The basic configuration files are as follows:

- [Startup configuration file \(ilogtail_config.json\)](#)
- [Account ID configuration file](#)
- [User-defined identifier file \(user_defined_id\)](#)
- [Logtail configuration file \(user_log_config.json\)](#)

The basic record files are as follows:

- [AppInfo record file \(app_info.json\)](#)
- [Logtail operational log file \(ilogtail.LOG\)](#)
- [Logtail plug-in log file \(logtail_plugin.LOG\)](#)
- [Container path mapping file \(docker_path_config.json\)](#)

Startup configuration file (ilogtail_config.json)

This file is used to query or set Logtail runtime parameters. The file is in the JSON format.

After you install Logtail, you can use the startup configuration file to perform the following operations:

- Change the values of the Logtail runtime parameters.

You can change the CPU usage threshold, usage threshold of terminate and stay resident (TSR) programs, and other settings.

- Check whether the installation commands are correct.

The settings of `config_server_address` and `data_server_list` in this file depend on the parameters and installation commands selected when you installed Logtail. If the region specified in `config_server_address` is unreachable or is different from the region where Log Service resides, the selected parameters or commands are incorrect. In this case, Logtail cannot collect logs and must be reinstalled.

Warning

- The file must be a valid JSON file. Otherwise, Logtail cannot be started.
- If you modify the file, you must restart Logtail to validate your modifications.

The following table describes the default parameters in the startup configuration file. You can also add other parameters. For more information, see [Set Logtail startup parameters](#).

Default parameters

Parameter	Description
<code>config_server_address</code>	The address that Logtail uses to receive the configuration file from Log Service. This address depends on the parameters and installation commands that you selected when you installed Logtail. Ensure that the address is reachable and is in the same region as Log Service.
<code>data_server_list</code>	The data server address. This address depends on the parameters and installation commands that you selected when you installed Logtail. Ensure that the address is reachable and is in the same region as Log Service.
<code>cluster</code>	The name of the region where a server resides.
<code>endpoint</code>	The endpoint of Log Service. For more information, see View the information of a project .
<code>cpu_usage_limit</code>	The CPU usage threshold, which is calculated by core.
<code>mem_usage_limit</code>	The TSR usage threshold.
<code>max_bytes_per_sec</code>	The traffic limit on the raw data that is sent by Logtail. If the value of this parameter is greater than 20 Mbit/s, traffic limiting does not take effect.
<code>process_thread_count</code>	The number of threads that Logtail uses to write data to log files.
<code>send_request_concurrency</code>	The number of concurrent requests for sending data packets asynchronously. Logtail sends data packets asynchronously by default. If the write transactions per second (TPS) is high, you can set a greater value for this parameter.

- File path
 - Linux: The file is stored in `/usr/local/ilogtail/ilogtail_config.json`.

- **Container Service:** The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_CONFIG` of the Logtail container. You can run the command `docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_CONFIG` to query the file path. For example, the file path is `/etc/ilogtail/conf/cn-hangzhou/ilogtail_config.json`.
- **Windows:**
 - 64-bit: The file is stored in `C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json`.
 - 32-bit: The file is stored in `C:\Program Files\Alibaba\Logtail\ilogtail_config.json`.
- **Sample file**

```
$cat /usr/local/ilogtail/ilogtail_config.json
{
  "config_server_address": "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
  "data_server_list":
  [
    {
      "cluster": "ap-southeast-2",
      "endpoint": "cn-hangzhou-intranet.log.aliyuncs.com"
    }
  ],
  "cpu_usage_limit": 0.4,
  "mem_usage_limit": 100,
  "max_bytes_per_sec": 2097152,
  "process_thread_count": 1,
  "send_request_concurrency": 4,
  "streamlog_open": false
}
```

Account ID configuration file

This file contains the ID of your Apsara Stack tenant account. The file indicates that the account can collect logs from the server where Logtail is installed. If you want to collect logs from ECS instances that do not belong to your account or from on-premises data centers, you must create an account ID configuration file.

Note

- The file is used only when you collect logs from ECS instances that do not belong to your account and or from on-premises data centers.
- The file can contain only the ID of your Apsara Stack tenant account. It cannot contain the IDs of RAM users under your Apsara Stack tenant account.
- The file name cannot contain a suffix.
- Each Logtail can have multiple account ID configuration files. Each Logtail container can have only one account ID configuration file.

● File path

- **Linux:** The file is stored in `/etc/ilogtail/users/`.
- **Container Service:** The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_USER_ID` of the Logtail container. You can run the command `docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_ID` to query the file path.
- **Windows:** The file is stored in `C:\LogtailData\users\`.

- Sample file

```
$ls /etc/ilogtail/users/  
*****
```

User-defined identifier file (user_defined_id)

This file is used to configure machine groups with user-defined identifiers. For more information, see [Create a machine group based on a custom ID](#).

 Note

- This file is used only when you configure a machine group with user-defined identifiers.
- If you configure multiple user-defined identifiers for a machine group, separate them with line breaks.

- File path

- Linux: The file is stored in `/etc/ilogtail/user_defined_id`.
- Container Service: The file is stored in a Logtail container. The file path is specified in the environment variable `ALIYUN_LOGTAIL_USER_DEFINED_ID` of the Logtail container. You can run the command `docker inspect $(logtail_container_name) | grep ALIYUN_LOGTAIL_USER_DEFINED_ID` to query the file path.
- Windows: The file is stored in `C:\LogtailData\user_defined_id`.

- Sample file

```
$cat /etc/ilogtail/user_defined_id  
aliyun-ecs-rs1e16355
```

Logtail configuration file (user_log_config.json)

This file contains the Logtail configuration that Logtail receives from Log Service. The file is in the JSON format and is updated along with configuration updates. You can use this file to check whether the Logtail configuration is delivered to the server where Logtail is installed. If the Logtail configuration file exists and all contents in the file are up to date, the Logtail configuration is delivered.

 Notice

- We recommend that you do not modify the Logtail configuration file unless you need to specify sensitive information, such as the AccessKey pair and database password.
- You must upload this file when you submit a ticket.

- File path

- Linux: The file is stored in `/usr/local/ilogtail/user_log_config.json`.
- Container Service: The file is stored in `/usr/local/ilogtail/user_log_config.json`.
- Windows
 - 64-bit: The file is stored in `C:\Program Files (x86)\Alibaba\Logtail\user_log_config.json`.
 - 32-bit: The file is stored in `C:\Program Files\Alibaba\Logtail\user_log_config.json`.

- Sample file

```
$cat /usr/local/ilogtail/user_log_config.json  
{  
  "metrics": {  
    "##1.0##k8s-log-c12ba2028****939f0b$app-java": {  
      "aliuid": "16542189****50",  
      "category": "app-java"    }  
  }  
}
```

```

    category : app-java ,
    "create_time" : 1534739165,
    "defaultEndpoint" : "cn-hangzhou-intranet.log.aliyuncs.com",
    "delay_alarm_bytes" : 0,
    "enable" : true,
    "enable_tag" : true,
    "filter_keys" : [],
    "filter_regs" : [],
    "group_topic" : "",
    "local_storage" : true,
    "log_type" : "plugin",
    "log_tz" : "",
    "max_send_rate" : -1,
    "merge_type" : "topic",
    "plugin" : {
      "inputs" : [
        {
          "detail" : {
            "IncludeEnv" : {
              "aliyun_logs_app-java" : "stdout"
            },
            "IncludeLabel" : {
              "io.kubernetes.container.name" : "java-log-demo-2",
              "io.kubernetes.pod.namespace" : "default"
            },
            "Stderr" : true,
            "Stdout" : true
          },
          "type" : "service_docker_stdout"
        }
      ]
    },
    "priority" : 0,
    "project_name" : "k8s-log-c12ba2028c*****ac1286939f0b",
    "raw_log" : false,
    "region" : "cn-hangzhou",
    "send_rate_expire" : 0,
    "sensitive_keys" : [],
    "tz_adjust" : false,
    "version" : 1
  }
}
}

```

AppInfo record file (app_info.json)

This file contains the startup time of Logtail. It also contains the IP address and hostname that Logtail obtains. You must check the IP address obtained by Logtail when you configure an [IP address-based machine group](#).

In most cases, Logtail obtains server IP addresses based on the following rules:

- If the IP address of a server is associated with its hostname in the `/etc/hosts` server file, Logtail obtains the IP address.
- If the IP address of a server is not associated with its hostname, Logtail obtains the IP address of the first network interface card (NIC) on the server.

 Note

- The AppInfo record file contains only the basic Logtail information, which cannot be manually modified.
- If you modify the hostname or other network settings of the server, you must restart Logtail to obtain a new IP address.

Parameters

Parameter	Description
UUID	The serial number of the server.
hostname	The hostname.
instance_id	The unique identifier of Logtail. This identifier is randomly generated.
ip	<p>The IP address that is obtained by Logtail. If this parameter is not specified, Logtail has not obtained the IP address of a server. In this case, Logtail cannot function properly. You must set an IP address for your server and restart Logtail.</p> <div data-bbox="612 1064 1390 1249" style="background-color: #e1f5fe; padding: 5px;"> <p> Note If the machine group is an IP address-based machine group, ensure that the IP address specified for the machine group is the same as the value of this parameter. If the two IP address are different, modify the IP address that you specified for the machine group in the Log Service console. Check the IP addresses again after 1 minute.</p> </div>
logtail_version	The version of Logtail.
os	The version of the operating system.
update_time	The last startup time of Logtail.

- File path
 - Linux: The file is stored in `/usr/local/ilogtail/app_info.json`.
 - Container Service: The file is stored in `/usr/local/ilogtail/app_info.json`.
 - Windows
 - 64-bit: The file is stored in `C:\Program Files (x86)\Alibaba\Logtail\app_info.json`.
 - 32-bit: The file is stored in `C:\Program Files\Alibaba\Logtail\app_info.json`.
- Sample file

```
$cat /usr/local/ilogtail/app_info.json
{
  "UUID" : "",
  "hostname" : "logtail-ds-slpn8",
  "instance_id" : "E5F93BC6-B024-11E8-8831-0A58AC14039E_1**. ***. ***. ***_1536053315",
  "ip" : "1**. ***. ***. **",
  "logtail_version" : "0.16.13",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
  "update_time" : "2018-09-04 09:28:36"
}
```

Logtail operational log file (ilogtail.LOG)

This file contains operational information about Logtail. Log severity levels are ranked as follows in ascending order: `INFO` , `WARN` , `ERROR` . Logs of the `INFO` level can be ignored.

- File path
 - For Linux: The file is stored in `/usr/local/ilogtail/ilogtail.LOG`.
 - Container Service: The file is stored in `/usr/local/ilogtail/ilogtail.LOG`.
 - Windows
 - 64-bit: The file is stored in `C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log`.
 - 32-bit: The file is stored in `C:\Program Files\Alibaba\Logtail\logtail_*.log`.
- Sample file

```
$tail /usr/local/ilogtail/ilogtail.LOG
[2018-09-13 01:13:59.024679] [INFO] [3155] [build/release64/sls/ilogtail/elogtail.cpp:123] change working dir:/usr/local/ilogtail/
[2018-09-13 01:13:59.025443] [INFO] [3155] [build/release64/sls/ilogtail/AppConfig.cpp:175] load logtail config file, path:/etc/ilogtail/conf/ap-southeast-2/ilogtail_config.json
[2018-09-13 01:13:59.025460] [INFO] [3155] [build/release64/sls/ilogtail/AppConfig.cpp:176] load logtail config file, detail:{
  "config_server_address" : "http://logtail.ap-southeast-2-intranet.log.aliyuncs.com",
  "data_server_list" : [
    {
      "cluster" : "ap-southeast-2",
      "endpoint" : "ap-southeast-2-intranet.log.aliyuncs.com"
    }
  ]
}
```

Logtail plug-in log file (logtail_plugin.LOG)

This file contains operational information about plug-ins, such as `stdout`, `binlog`, and `HTTP` plug-ins. Log severity levels are ranked as follows in ascending order: `INFO` , `WARN` , `ERROR` . Logs of the `INFO` level can be ignored.

- File path
 - Linux: The file is stored in `/usr/local/ilogtail/logtail_plugin.LOG`
 - Container Service: The file is stored in `/usr/local/ilogtail/logtail_plugin.LOG`.
 - Windows: The file is not supported.
- Sample file

```

$tail /usr/local/ilogtail/logtail_plugin.LOG
2018-09-13 02:55:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 02:55:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:00:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:00:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##s/s-zc-test-hz-pub$docker-stdout-config,k8s-stdout] open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379573 status:794354-64769-40379963
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout] open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379573 status:794354-64769-40379963
2018-09-13 03:04:26 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##s/s-zc-test-hz-pub$docker-stdout-config,k8s-stdout] close file, reason:no read timeout file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379963 status:794354-64769-40379963
2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout] close file, reason:no read timeout file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379963 status:794354-64769-40379963
2018-09-13 03:05:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:05:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop

```

Container path mapping file (docker_path_config.json)

This file is automatically created only when container files are collected. It records path mappings between container files and actual files. The file is in the JSON format.

 **Note** This file is only an information record file. Modifications to this file do not take effect. If you delete this file, another one is automatically created without service interruptions.

- **File path**
The file is stored in `/usr/local/ilogtail/docker_path_config.json`.
- **Sample file**

```
$cat /usr/local/ilogtail/docker_path_config.json
{
  "detail": [
    {
      "config_name": "##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$nginx",
      "container_id": "df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10",
      "params": "{\n  \"ID\": \"df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10\",\n  \"Path\n\": \"/logtail_host/var/lib/docker/overlay2/947db346695a1f65e63e582ecfd10ae1f57019a1b99260b6c83d00fcd1892874\n/diff/var/log\",\n  \"Tags\": [\n    \"nginx-type\",\n    \"access-log\",\n    \"_image_name_\",\n    \"registry.cn-h\nangzhou.aliyuncs.com/log-service/docker-log-test:latest\",\n    \"_container_name_\",\n    \"nginx-log-demo\",\n    \"_pod_name_\",\n    \"nginx-log-demo-h2lzc\",\n    \"_namespace_\",\n    \"default\",\n    \"_pod_uid_\",\n    \"8\n7e56ac3-b65b-11e8-b172-00163f008685\",\n    \"_container_ip_\",\n    \"172.20.4.224\",\n    \"purpose\",\n    \"test\n\"\n ]\n}\n"
    }
  ],
  "version": "0.1.0"
}
```

30.3.1.2. Installation

30.3.1.2.1. Install Logtail in Linux

This topic describes how to install Logtail on a Linux server.

Supported systems

Logtail supports the following x86-64 (64-bit) Linux operating systems:

- Aliyun Linux
- Ubuntu
- Debian
- CentOS
- openSUSE
- Red Hat

Procedure

Note If you have installed Logtail, the installer will uninstall the existing version of Logtail, delete the `/usr/local/ilogtail` directory, and then reinstall Logtail. By default, Logtail runs after the installation and at startup.

1. Run the following command to download the Logtail installer:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh; chmod 755 logtail.sh
```

Note You must replace `${service:sls-backend-server:sls_data.endpoint}` in the command with the actual endpoint. For more information about endpoints, see [View the information of a project](#).

2. Run the installation command. Start Linux PowerShell and run the following command as an administrator to install Logtail:

```
./logtail.sh install
```

Note If you have installed Logtail, the installer will uninstall the existing version of Logtail, delete the `/usr/local/ilogtail` directory, and then reinstall Logtail.

3. Configure an account ID for a server.

View the version of Logtail

To view the version of Logtail, open the file in the `/usr/local/ilogtail/app_info.json` directory. The `logtail_version` field shows the version of Logtail.

```
$cat /usr/local/ilogtail/app_info.json
{
  "UUID" : "ODF18E97-0F2D-486F-B77F-*****",
  "hostname" : "david*****",
  "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*****_1515129548",
  "ip" : "*****",
  "logtail_version" : "0.16.0",
  "os" : "Linux; 2.6.32-220.23.2.ali1113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013; x86_64",
  "update_time" : "2018-01-05 13:19:08"
}
```

Upgrade Logtail

You can use the Logtail installer (`logtail.sh`) to upgrade Logtail. The installer selects an upgrade method based on the configurations of the existing Logtail.

Note During the upgrade, Logtail is temporarily stopped. Only related files are overwritten. The configuration file, checkpoint file, and logs are retained.

Run the following commands to upgrade Logtail:

```
# Download the Logtail installer.
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh; chmod 755 logtail.sh
# Upgrade Logtail.
sudo ./logtail.sh upgrade
```

Response:

```
# The upgrade is successful.
Stop logtail successfully.
ilogtail is running
Upgrade logtail success
{
  "UUID" : "****",
  "hostname" : "****",
  "instance_id" : "****",
  "ip" : "****",
  "logtail_version" : "0.16.11",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
  "update_time" : "2018-08-29 15:01:36"
}
# The upgrade fails because the current version is the latest version.
[Error]: Already up to date.
```

Start and stop Logtail

- Start Logtail

Run the following command as an administrator:

```
/etc/init.d/ilogtailed start
```

- Stop Logtail

Run the following command as an administrator:

```
/etc/init.d/ilogtailed stop
```

Uninstall Logtail

Run Linux PowerShell as an administrator to download *logtail_installer* and uninstall Logtail:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh
chmod 755 logtail.sh
./logtail.sh uninstall
```

30.3.1.2.2. Install Logtail in Windows

This topic describes how to install Logtail on a Windows server.

Supported systems

Logtail supports the following Windows operating systems:

- Windows 7 (Client) 32-bit
- Windows 7 (Client) 64-bit
- Windows Server 2008 32-bit
- Windows Server 2008 64-bit
- Windows Server 2012 64-bit
- Windows Server 2016 64-bit

Procedure

1. Download the installation package. Run the following command to download the installation package:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/windows/logtail_installer.zip
```

Note You must replace `${service:sls-backend-server:sls_data.endpoint}` in the command with the actual endpoint. For more information about endpoints, see [View the information of a project](#).

2. Decompress the `logtail_installer.zip` package to the current directory.
3. Run the installation command. Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the installation command based on the network type.

```
.\logtail_installer.exe install me-east-1
```

Note You must replace `${region}` in the command with the actual endpoint. For more information about endpoints, see [View the information of a project](#).

4. [Configure an account ID for a server](#).

Installation directory

After you run the installation command, Logtail is installed in the specified directory. The directory cannot be changed. In the directory, you can [View the version of Logtail](#) in the `app_info.json` file or [Uninstall Logtail](#).

The installation directory is as follows:

- 32-bit Windows: `C:\Program Files\Alibaba\Logtail`
- 64-bit Windows: `C:\Program Files (x86)\Alibaba\Logtail`

Note You can run 32-bit or 64-bit applications in a 64-bit Windows operating system. However, the operating system stores 32-bit applications in separate x86 folders to ensure compatibility.

Logtail for Windows is a 32-bit application. Therefore, it is installed in the `Program Files (x86)` folder in 64-bit Windows. If Logtail for 64-bit Windows becomes available in the future, it will be installed in the `Program Files` folder.

View the version of Logtail

To view the version of Logtail, go to the [default installation directory](#), and then use the notepad or another text editor to open the `app_info.json` file. The `logtail_version` field shows the version of Logtail.

In the following example, the version of Logtail is 1.0.0.0:

```
{
  "logtail_version" : "1.0.0.0"
}
```

Upgrade Logtail

- Automatic upgrade
Logtail later than 1.0.0.0 is automatically upgraded in Windows.
- Manual upgrade
Logtail earlier than 1.0.0.0 must be manually upgraded. The manual upgrade procedure is the same as the [installation procedure](#).

 **Note** During a manual upgrade, the files in the original installation directory are deleted. We recommend that you back up the files before you perform a manual upgrade.

Start and stop Logtail

Open the **Control Panel**, choose **System and Security > Administrative Tools**, and then double-click **Services**.

Find the service based on your Logtail version.

- Logtail 0.x.x.x: LogtailWorker.
- Logtail 1.0.0.0 and later: LogtailDaemon.

Perform the following operations as required:

- **Start Logtail**

Right-click the service and select **Start** from the shortcut menu.

- **Stop Logtail**

Right-click the service and select **Stop** from the shortcut menu.

- **Restart Logtail**

Right-click the service and select **Restart** from the shortcut menu.

Uninstall Logtail

Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the following command:

```
.\logtail_installer.exe uninstall
```

After Logtail is uninstalled, the **installation directory** is deleted. However, some residual configuration data is still maintained in the `C:\LogtailData` directory. You can manually delete the data. The residual configuration data includes the following information:

- *checkpoint*: checkpoints of all plug-ins, for example, the Windows event log plug-in.
- *logtail_check_point*: checkpoints of Logtail.
- *users*: IDs of Apsara Stack tenant accounts.

30.3.1.2.3. Set Logtail startup parameters

This topic describes how to set Logtail startup parameters.

Context

You may need to set Logtail startup parameters in the following scenarios:

- You need to collect a large number of log files that consume much memory. You want to maintain the metadata (such as the file signature, collection location, and file name) of each file in the memory.
- The CPU usage is high due to heavy log data traffic.
- The traffic sent to Log Service is heavy due to a large amount of log data.
- You want to collect syslogs or TCP data streams.

Startup configurations

- **File path**

```
/usr/local/ilogtail/ilogtail_config.json
```

- **File format**

JSON

- Sample file (only partial configurations are provided)

```

{
  ...
  "cpu_usage_limit" : 0.4,
  "mem_usage_limit" : 100,
  "max_bytes_per_sec" : 2097152,
  "process_thread_count" : 1,
  "send_request_concurrency" : 4,
  "streamlog_open" : false,
  "streamlog_pool_size_in_mb" : 50,
  "streamlog_rcv_size_each_call" : 1024,
  "streamlog_formats":[],
  "streamlog_tcp_port" : 11111,
  "buffer_file_num" : 25,
  "buffer_file_size" : 20971520,
  "buffer_file_path" : "",
  ...
}
    
```

Startup parameters

Parameter	Description	Example
cpu_usage_limit	The CPU usage threshold for a single core. Data type: double.	For example, the value 0.4 indicates that the CPU usage of Logtail is limited to 40% processing capacity of a single core. In most cases, the processing capacity of a single core is about 24 MB/s in the simple mode and 12 MB/s in the full regex mode.
mem_usage_limit	The usage threshold of the resident memory. Data type: integer. Unit: MB.	For example, the value 100 indicates that the memory usage of Logtail is limited to 100 MB. If the threshold is exceeded, Logtail restarts. If you want to collect more than 1,000 log files, you can increase the threshold value.
max_bytes_per_sec	The traffic limit on the raw data that is sent by Logtail. Data type: integer. Unit: bytes/s.	For example, the value 2097152 indicates that the data transfer rate of Logtail is limited to 2 MB/s.
process_thread_count	The number of threads that Logtail uses to process data.	Default value: 1. Each thread provides a write speed of 24 MB/s in the simple mode and 12 MB/s in the full regex mode. We recommend that you do not modify the default value.
send_request_concurrency	Logtail sends data packets asynchronously by default. If the write transactions per second (TPS) is high, you can set this parameter to a greater value.	Twenty asynchronous concurrencies are provided by default. Each concurrency can provide 0.5 MB/s to 1 MB/s network throughput. The number of concurrencies varies with the network delay.
streamlog_open	Specifies whether to receive syslogs. Data type: Boolean.	The value false indicates that syslogs are not received. The value true indicates that syslogs are received.

Parameter	Description	Example
streamlog_pool_size_in_mb	The size of memory pool that the syslog server uses to cache syslogs. Unit: MB.	Logtail requests memory when it starts. Set the memory pool size based on the server memory size and your business requirements.
streamlog_rcv_size_each_call	The size of the buffer that Logtail uses when the linux socket rcv API is called. Unit: bytes. Valid values: 1024 to 8192.	You can set a greater value if the syslog traffic is high.
streamlog_formats	The method that is used to parse received syslogs.	N/A
streamlog_tcp_addr	The associated address that Logtail uses to receive syslogs. Default value: 0.0.0.0.	N/A
streamlog_tcp_port	The TCP port that Logtail uses to receive syslogs.	Default value: 11111.
buffer_file_num	The maximum number of cached files. If a network exception occurs or the writing quota is exceeded, Logtail writes parsed logs to local files in the installation directory. After the network recovers or a new writing quota is available, Logtail retries to send the logs to Log Service.	Default value: 25.
buffer_file_size	The maximum number of bytes that can be contained in each cache file. The maximum disk space available for cache files is the value of <code>buffer_file_num</code> multiplied by the value of <code>buffer_file_size</code> .	Default value: 20971520 bytes (20 MB).
buffer_file_path	The directory in which cached files are stored. If you modify this parameter, you must move the files (for example, <code>logtail_buffer_file_*</code>) in the old cache directory to the new directory. Then, Logtail can read, send, and delete the cache files.	The default value is null, which indicates that the cached files are stored in the Logtail installation directory <code>/usr/local/ilogtail</code> .
bind_interface	The name of the NIC associated with the local machine, for example, <code>eth1</code> . This parameter is valid only for Logtail that runs in Linux.	By default, the available NICs are automatically associated with the local machine. If you specify this parameter, Logtail will use the specified NIC to upload logs.

Parameter	Description	Example
check_point_filename	The full path in which the checkpoint file is stored. This parameter is used to customize the path to store the checkpoint file of Logtail.	Default value: <code>/tmp/logtail_checkpoint</code> . We recommend that Docker users modify this path and mount the directory where the checkpoint file resides to the host. Otherwise, duplicate collection occurs due to checkpoint data loss when the container is released. For example, you can set <code>check_point_filename</code> to <code>/data/logtail/check_point.dat</code> in Docker and add <code>-v /data/docker1/logtail:/data/logtail</code> to the Docker startup command. Then, the <code>/data/docker1/logtail</code> directory of the host is mounted to the <code>/data/logtail</code> directory of Docker.

 Note

- The preceding table lists only the common startup parameters. If the `ilogtail_config.json` file contains parameters that are not listed in the table, the default settings are used for these parameters.
- We recommend that you do not add unnecessary parameters to the `ilogtail_config.json` file.

Modify configurations

1. Configure the `ilogtail_config.json` file as needed.
Ensure that the modified configurations are in the valid JSON format.
2. Restart Logtail to apply the modified configurations.

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
/etc/init.d/ilogtaild status
```

30.3.1.3. Logtail machine group

30.3.1.3.1. Overview

Log Service uses server groups to manage the servers from which you want to collect logs by using Logtail.

A server group is a virtual group that contains multiple servers. If you want to use the same Logtail configuration to collect logs from multiple servers, you can add the servers to a server group and apply the Logtail configuration to the server group.

To define a server group, you can use either of the following methods:

- **IP address:** Add the IP addresses of servers to a server group. Each server in the server group can be identified by using its unique IP address.
- **Custom ID-based server groups:** Customize an identifier for a server group and use the identifier for the servers in the server group.

 Note Windows and Linux servers cannot be added to the same server group.

IP address-based server groups

You can add multiple servers to a server group by adding their IP addresses to the server group. Then, you can create a Logtail configuration for all the servers at the same time.

- If you use ECS instances and have not associated them with hostnames or changed their network types, you can add the private IP addresses of the instances to the server group.

- In other cases, you must add the server IP addresses obtained by Logtail to a server group. The IP address of each server is recorded in the IP address field of the `app_info.json` file on the server.

 **Note** The `app_info.json` file records the internal information of Logtail. This file includes the server IP addresses obtained by Logtail. If you modify the IP address field of the file, the IP addresses obtained by Logtail remain unchanged.

Logtail obtains a server IP address by using the following methods:

- If the IP address of a server is associated with the hostname in the `/etc/hosts` file of the server, Logtail obtains this IP address.
- If the IP address of a server is not associated with the hostname, Logtail obtains the IP address of the first network interface controller (NIC) on the server.

For more information, see [Create an IP address-based server group](#).

Custom ID-based server groups

You can use custom IDs to dynamically define server groups.

An application system consists of multiple modules. You can scale out each module by adding multiple servers to the module. If you want to collect logs by module, you can create a server group for each module. Therefore, you must specify a custom ID for each server in each module. For example, a website consists of an HTTP request processing module, a caching module, a logic processing module, and a storage module. The custom IDs of these modules can be `http_module`, `cache_module`, `logic_module`, and `store_module`.

For more information, see [Create a machine group based on a custom ID](#).

30.3.1.3.2. Create a machine group based on a server IP address

This topic describes how to create a machine group based on a server IP address. You can create a machine group based on a server IP address that is obtained by using a Logtail configuration file. You can then use the same Logtail configuration file to collect logs from the machine group.

Prerequisites

- A project is created. A Logstore is created in the project.
- One or more servers are available. The IP addresses of the servers are obtained.
- Logtail is installed on the servers. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
- The ID of your Apsara Stack tenant account is configured on the server. For more information, see [Configure an account ID for a server](#).

Procedure

1. [Log on to the Log Service console](#).
2. In the **Projects** section, click the name of a project.
3. In the left-side navigation pane, click the **Machine Groups** icon.
4. In the pane that appears, click the  icon next to **Machine Group** and select **Create Machine Group** from the shortcut menu. You can also create a machine group in the Logtail configuration wizard.
5. Create a machine group.
 - i. Enter a machine group name in the `name` field. The machine group name must be 3 to 128 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.

 **Notice** After the machine group is created, its name cannot be modified.

- ii. Select **IP Addresses** from the **Identifier** drop-down list.

- iii. Enter a topic name in the **Topic** field. For more information about topics, see [Generate a topic](#).
- iv. Enter the IP addresses of the servers in the **IP Addresses** text box.

 Notice

- Separate each IP address with a line break.
- Do not add Windows servers and Linux servers to the same machine group.

6. Click **OK**.

Result

You can view the created machine group in the **Machine Groups** pane.



30.3.1.3.3. Create a machine group based on a custom ID

This topic describes how to create a machine group based on a custom ID.

Context

You can use a custom ID to identify a machine group in the following scenarios:

- Servers reside in multiple custom network environments such as virtual private clouds (VPCs). IP addresses of different servers may be the same. In this scenario, Log Service cannot distinguish between servers based on IP addresses.
- You want to implement automatic server discovery. To do this, you only need to set a custom ID of a new server to the custom ID of an existing machine group. Log Service automatically identifies the server and adds it to the machine group.

Procedure

1. Set a custom ID on a server.

○ **Linux Logtail**

Set a custom ID in the `/etc/ilogtail/user_defined_id` file.

For example, if you need to set a custom ID of a server to `userdefined`, run the following command to open the file:

```
# vim /etc/ilogtail/user_defined_id
```

In the file, enter `userdefined`.

○ **Windows Logtail**

Set a custom ID in the `C:\LogtailData\user_defined_id` file.

For example, if you need to set a custom ID of a server, run the following command:

```
C:\LogtailData>more user_defined_id  
userdefined_windows
```

Notice

- A machine group cannot include both Linux and Windows servers. Therefore, do not set a custom ID of a Linux server and a custom ID of a Windows server to the same value.
- You can set multiple custom IDs for a single server. Separate each custom ID with a line break.
- If the `/etc/ilogtail/` or `C:\LogtailData` directory or the `/etc/ilogtail/user_defined_id` or `C:\LogtailData\user_defined_id` file does not exist, create the directory or the file.

2. Create a machine group.

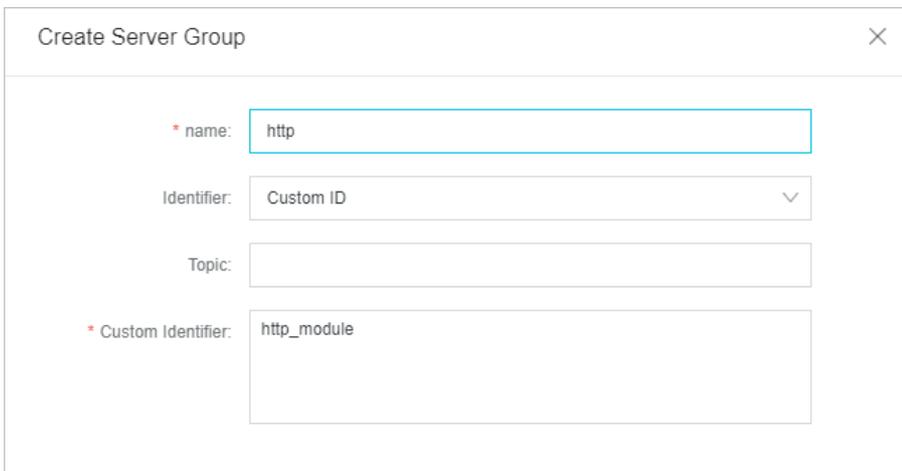
- Log on to the Log Service console.**
- In the **Projects** section, click a project.
- In the left-side navigation pane, click the **Machine Groups** icon.
- Click the  icon next to **Machine Groups**, and select **Create Machine Group** from the shortcut menu.
- Set the parameters of the machine group.

- **name:** Enter a machine group name.

The machine group name must be 3 to 128 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.

 **Note** After the machine group is created, its name cannot be modified.

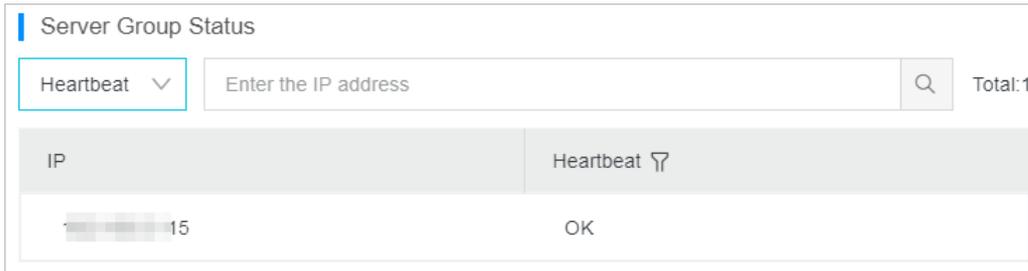
- **Identifier:** Select **Custom ID**.
- **Topic:** Enter a topic name for the machine group. For more information, see [Generate a topic](#).
- **Custom Identifier:** Enter the custom ID that you set in Step 1.



- Click **OK**.

 **Note** If you need to add a server to the machine group, set a custom ID of the server to the custom ID of the machine group. The server is then listed in the **Machine Group Status** section.

- View the status of the machine group.** In the **Machine Groups** pane, click the name of the machine group. On the **Machine Group Settings** page, view the status of the machine group. You can view the IP address list of the servers in the machine group and their heartbeat status.



Note

- The **Machine Group Status** section lists the IP addresses of the servers whose custom ID is the same as the custom ID that you set for the machine group.
For example, the custom ID of a machine group is `userdefined` and the IP addresses in the **Machine Group Status** section are `10.10.10.10`, `10.10.10.11`, and `10.10.10.12`. This means that the custom ID `userdefined` is set for the three servers. If you need to add the server whose IP address is `10.10.10.13` to the machine group, set the custom ID `userdefined` for the server. Then, the IP address of the server is displayed in the **Machine Group Status** section. Log Service collects logs from the server based on the Logtail configuration file of the machine group.
- The heartbeat status indicates whether the connection between a server and Log Service is normal. For information about how to troubleshoot heartbeat errors, see [What can I do if no heartbeat packet is received from a Logtail client?](#)

Delete the custom IDs of a server

If you need to change the ID of a server from a custom ID to the server IP address, delete the `user_defined_id` file. The change takes effect within 1 minute.

- In Linux, run the following command to delete the file:

```
rm -f /etc/ilogtail/user_defined_id
```

- In Windows, run the following command to delete the file:

```
del C:\LogtailData\user_defined_id
```

Time required for a change to take effect

After you create, edit, or delete a `user_defined_id` file, the change takes effect within 1 minute.

If you need the change to take immediate effect, restart Logtail.

- In Linux, run the following commands to restart Logtail:

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
```

- In Windows, perform the following steps to restart Logtail:

Open the Control Panel. In the window that appears, choose **Control Panel > Administrative Tools > Services**. Right-click **LogtailWorker**, and select **Restart** from the shortcut menu.

Example

An application consists of multiple modules. Each module runs on multiple servers. For example, a website consists of an HTTP request processor, a cache, a logic processor, and a storage. You may scale out each module by adding multiple servers. You need to collect logs from both the existing and new servers.

1. Set a custom ID for each server. Install Logtail on the servers and set a custom ID for each server. In this example, you can use four custom IDs: `http_module`, `cache_module`, `logic_module`, and `store_module`. Each custom ID corresponds to a module.

2. Create a machine group for each module. When you create a machine group for a module, enter the custom ID of the module in the Custom Identifier field.

The screenshot shows a 'Create Server Group' dialog box with the following fields:

- * name:** http
- Identifier:** Custom ID
- Topic:** (empty)
- * Custom Identifier:** http_module

3. View the status of the machine group. On the Machine Group Settings page of the machine group, you can view the status of the machine group in the Machine Group Status section. You can view the list of servers in the machine group and their heartbeat status.
4. If you need to add a server whose IP address is 10.1.1.3 to the machine group whose custom ID is http_module, set the custom ID http_module for the server. Then, you can view the server in the Machine Group Status section.

IP	Heartbeat
[blurred]	OK
10.1.1.3	OK

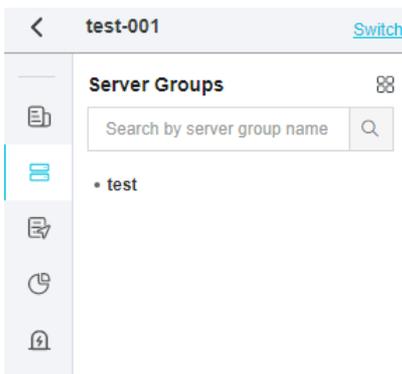
Search bar: IP [dropdown] Enter the IP address [input] [search icon] Total: 2

30.3.1.3.4. View server groups

This topic describes how to view the server groups of a project on the Server Groups page in the Log Service console.

Procedure

1. Log on to the Log Service console.
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the Server Groups icon to display the list of server groups. You can view all server groups of the project.



30.3.1.3.5. Modify a server group

This topic describes how to modify a server group in the Log Service console. After you create a server group, you can modify the parameters of the server group.

Procedure

1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group to be modified. On the **Server Group Settings** page, click **Modify**.

 **Note** The name of the server group cannot be modified.

5. Modify the parameters of the server group, and then click **Save**.

30.3.1.3.6. View the status of a server group

This topic describes how to view the status of a server group in the Log Service console. You can view the heartbeat information of Logtail to check whether Logtail is installed on the servers in a server group.

Procedure

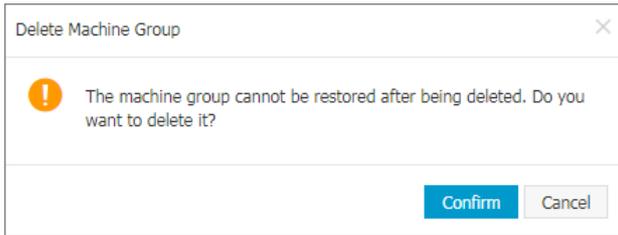
1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group. On the **Server Group Settings** page, check the server group status.
 - If the heartbeat is OK, Logtail is installed on the servers in the server group and Logtail is connected to Log Service.
 - If the heartbeat status is FAIL, Logtail fails to connect to Log Service. If the FAIL state persists, perform troubleshooting based on the instructions provided in [What can I do if no heartbeat packet is received from a Logtail client?](#)

30.3.1.3.7. Delete a server group

This topic describes how to delete a server group in the Log Service console. You can delete a server group if you no longer need to collect logs from the server group.

Procedure

1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Find the server group that you want to delete, click the  icon next to the server group, and then select **Delete**.
5. In the dialog box that appears, click **OK**.



30.3.1.3.8. Manage server group configurations

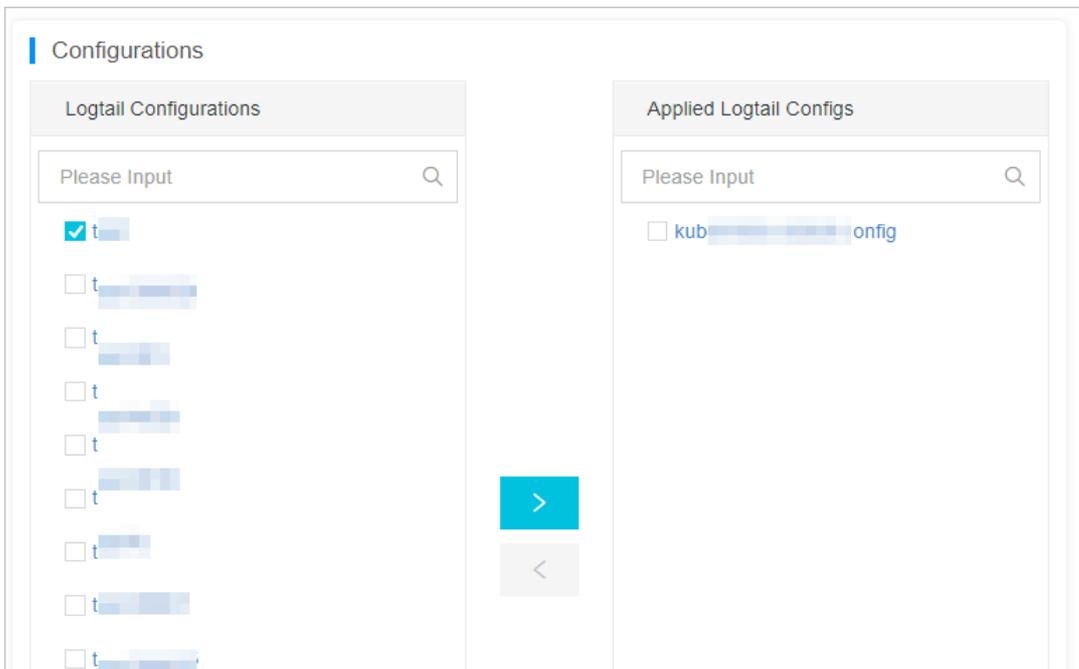
This topic describes how to manage server group configurations in the Log Service console. Log Service uses server groups to manage the servers from which you collect logs by using Logtail. In the Log Service console, you can create, view, modify, and delete server groups. You can also view the status of server groups, manage server group configurations, and apply server group identifiers.

Context

Log Service allows you to manage the Logtail configurations that you create for Logtail installed on the servers in a server group. You can apply Logtail configurations to a server group. The Logtail configurations determine what logs are collected on each server, how the logs are parsed, and which Logstore the logs are written to.

Procedure

1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group whose configurations you want to modify. On the **Server Group Settings** page, click **Modify**.
5. In the **Configurations** section, modify the Logtail configuration that you want to apply to the server group and click **Save**. After a Logtail configuration is added, it is delivered to Logtail on each server in the server group. After a Logtail configuration is removed, it is removed from Logtail.



30.3.1.3.9. Manage a Logtail configuration

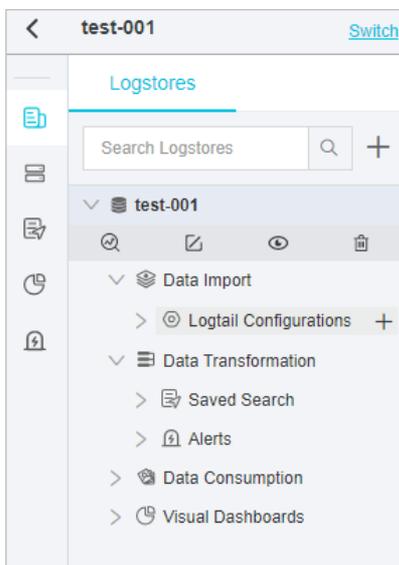
This topic describes how to manage a Logtail configuration in the Log Service console. Before you can collect logs from a server, you must install Logtail on the server. After you install Logtail, you must create a Logtail configuration in the Log Service console and apply the Logtail configuration to the server. You can create and modify Logtail configurations in Logstores.

Create a Logtail configuration

For information about how to create a Logtail configuration in the Log Service console, see [Configure text log collection](#).

View Logtail configurations

1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane, click the closing angle bracket (>) next to the target Logstore and choose **Data Import > Logtail Configurations**. Each item under **Logtail Configurations** indicates a Logtail configuration.



Modify a Logtail configuration

Under **Logtail Configurations**, click the name of the Logtail configuration. On the **Logtail Config** page, click **Modify**.

You can also change the log collection mode of the Logtail configuration, and then apply the Logtail configuration to the server group again. The process of modifying a Logtail configuration is the same as the process of creating a Logtail configuration.

Delete a Logtail configuration

Click the  icon next to the Logtail configuration, and then select **Delete**.

After the Logtail configuration is deleted, it is disassociated from the server group. Logtail no longer collects logs specified by the Logtail configuration.

30.3.1.3.10. Configure an account ID on a server

This topic describes how to configure the ID of an Apsara Stack tenant account on a server.

Prerequisites

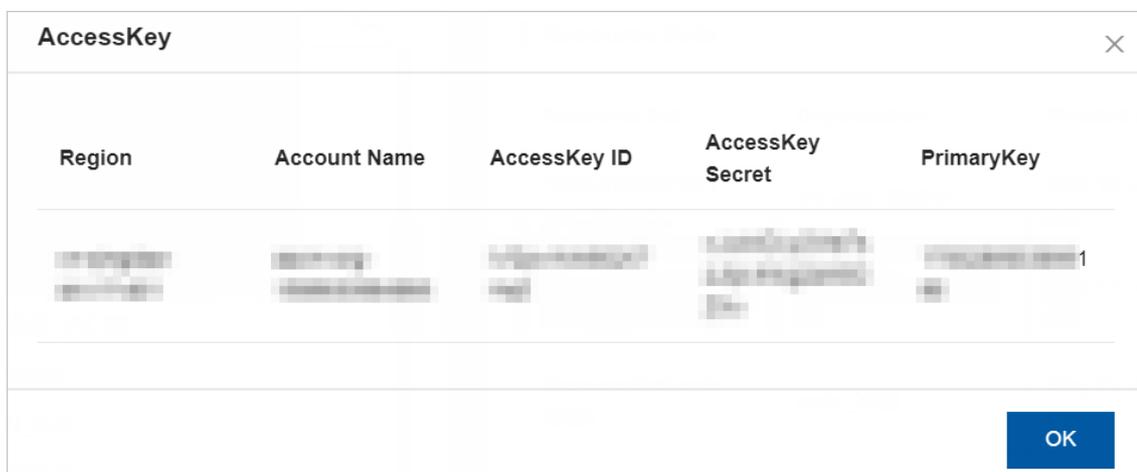
Logtail is installed on the server. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

Context

You must configure the ID of your Apsara Stack tenant account on the server on which Logtail is installed if the server is an ECS instance that belong to another Apsara Stack tenant account, a cloud server that is provided by a third-party vendor, or an on-premises server. If you do not configure the account ID on the server, Log Service cannot receive heartbeats from the server, or use Logtail to collect logs from the server.

Procedure

1. View the ID of your Apsara Stack tenant account.
 - i. Log on to the Apsara Stack Cloud Management (ASCM) console. For more information, see [Log on to the Log Service console](#).
 - ii. In the top navigation bar, click **Enterprise**.
 - iii. In the left-side navigation pane, click **Organizations**.
 - iv. On the page that appears, click the  icon next to the account, and select **AccessKey** from the shortcut menu.
 - v. In the **AccessKey** dialog box, view the account ID.



2. Log on to the server and configure the account ID on the server.

- o Linux server:

In the `/etc/ilogtail/users` directory, create a file. Set the name of the file to the account ID. If the directory does not exist, create the directory first. You can configure multiple account IDs on a server. For example, you can run the following commands to create files for two account IDs:

```
touch /etc/ilogtail/users/1*****
touch /etc/ilogtail/users/1*****
```

If you no longer need to collect logs from the server to a Log Service project of an Apsara Stack tenant account, run the following command to delete the file of the account:

```
rm /etc/ilogtail/users/1*****
```

- o Windows server:

In the `C:\LogtailData\users` directory, create a file. Set the name of the file to the account ID. To delete the account ID, delete the file.

For example, the path to the file of an account ID is `C:\LogtailData\users\1*****`.

Note

- After you configure the ID of an Apsara Stack tenant account on a server, the account is authorized to use Logtail for collecting logs from the server. If an account is no longer used to collect logs from the server, delete the account ID file from the server at the earliest opportunity.
- After you configure or delete an account ID, the change takes effect within 1 minute.

30.3.1.4. Text logs

30.3.1.4.1. Configure text log collection

This topic describes how to configure Logtail in the Log Service console to collect text logs from specified servers.

Prerequisites

Logtail is installed. Logtail can be installed on a Windows or Linux operating system. For more information, see [Install Logtail in Linux](#) and [Install Logtail in Windows](#).

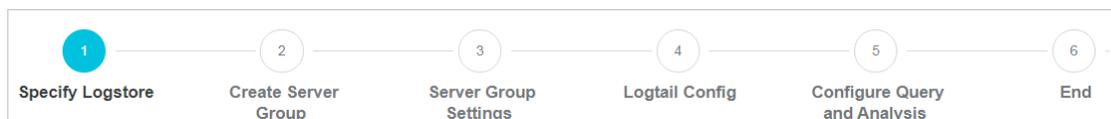
Limits

- Each log file can be collected by using only one Logtail configuration. If you want to collect a log file by using more than one Logtail configuration, we recommend that you use symbolic links. For example, to collect a log file by using two Logtail configurations in the `/home/log/nginx/log` directory, you can use the original log path for one Logtail configuration. Then, run the `ln -s /home/log/nginx/log /home/log/nginx/link_log` command to create a symbolic link for this directory and use the symbolic link as the log path for the other Logtail configuration.
- Logtail supports only Windows or Linux operating systems. For more information, see [Logtail overview](#).

Logtail configuration procedure

You can specify Logtail configurations in the Log Service console. Logtail supports various collection modes, such as simple mode, NGINX configuration mode, Apache configuration mode, IIS configuration mode, delimiter mode, JSON mode, and full regex mode.

Configuration procedure



Collection modes

Logtail supports various collection modes, such as simple mode, NGINX configuration mode, Apache configuration mode, IIS configuration mode, delimiter mode, JSON mode, and full regex mode.

- Simple mode
Logtail can be used to collect logs in the simple mode. For more information, see [Collect logs by line](#).
- Full regex mode
Logtail can be used to collect logs in the full regex mode. For more information, see [Use regular expressions to collect logs](#).
- Delimiter mode
Logtail can be used to collect logs in the delimiter mode. For more information, see [Collect DSV formatted logs](#).
- JSON mode

Logtail can be used to collect logs in the JSON mode. For more information, see [Collect JSON logs](#).

- **NGINX configuration mode**

Logtail can be used to collect logs in the NGINX configuration mode. For more information, see [Collect NGINX logs](#).

- **IIS configuration mode**

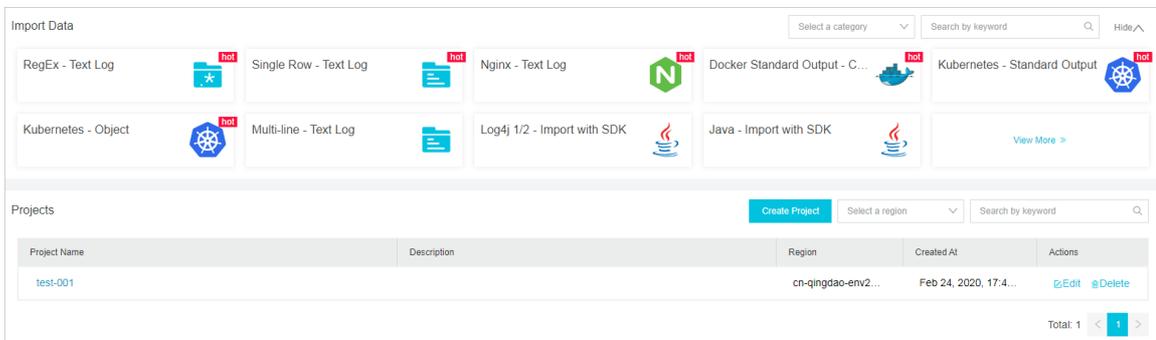
Logtail can be used to collect logs in the IIS configuration mode. For more information, see [Collect IIS logs](#).

- **Apache configuration mode**

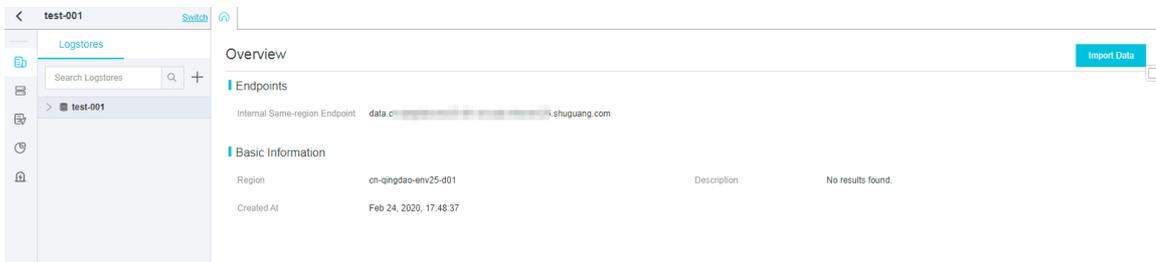
Logtail can be used to collect logs in the Apache configuration mode. For more information, see [Collect Apache logs](#).

Procedure

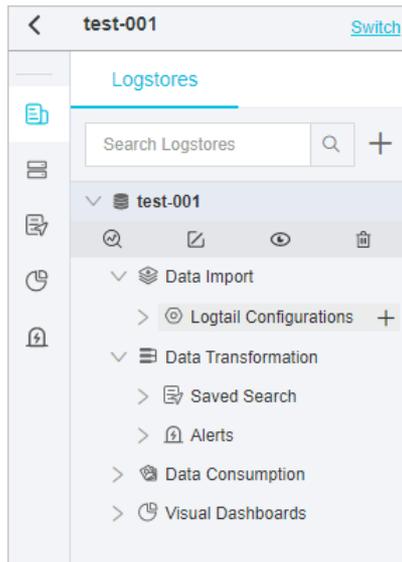
1. [Log on to the Log Service console](#).
2. Select a data source. You can use one of the following three methods to select a data source:
 - On the homepage of the Log Service console, select a data source in the **Import Data** section.



- In the **Projects** section, click a project name. On the **Overview** page, click **Import Data**.



- On the **Logstores** tab in the left-side navigation pane, find a Logstore and click the closing angle bracket (>) in front of the Logstore name. Then, click the plus sign (+) next to **Data Import**.



Select a data source based on your business requirements. Log Service supports the following log sources of text logs: **RegEx-Text Log**, **Single Row-Text Log**, **Multi-Row-Text Log**, **Delimiter Mode-Text Log**, **JSON-Text Log**, **Nginx-Text Log**, **IIS-Text Log**, and **Apache-Text Log**.

3. Select a Logstore, and then click Next. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

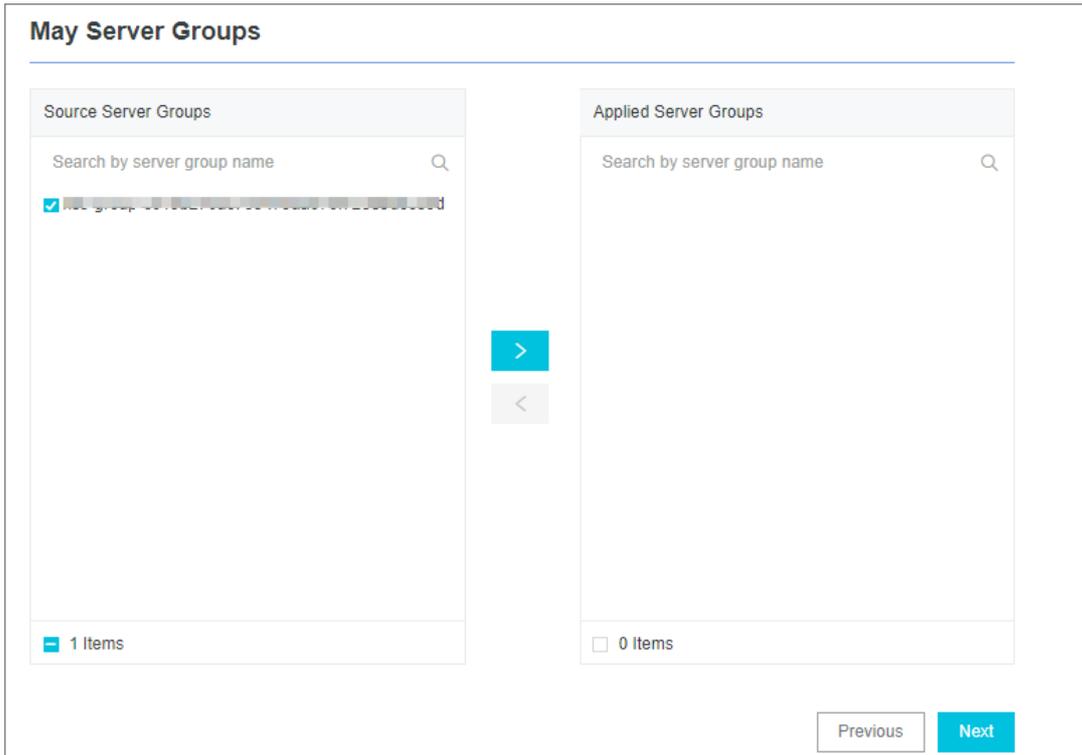
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.

5. Configure the server group, and then click Next. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Specify Logtail parameters. Logtail parameters vary based on collection modes. For more information, see the relevant parameters for specific collection modes.
7. (Optional)Specify **Advanced Options** and click **Next**.Specify **Advanced Options** based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ <code>utf8</code>: indicates UTF-8 encoding. ◦ <code>gbk</code>: indicates GBK encoding.

Parameter	Description
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> Never: All log files are continuously monitored and never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code> . It indicates that only logs with the severity level of WARNING or ERROR are collected. Filter logs that do not meet a condition: <ul style="list-style-type: none"> Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code> . It indicates that logs with the severity level of INFO or DEBUG are not collected. Set the condition to <code>Key:url Regex:.^(?!.*(healthcheck)).*</code> . It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiankong.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect logs.

30.3.1.4.2. Collect logs by line

This topic describes how to collect logs by line and configure indexes. You can specify the required settings in the Log Service console.

Context

To collect logs by line, you must select the simple mode. The simple mode can be divided into two types:

- Singleline mode

In this mode, each line of log data is considered as a log. Two logs in a log file are separated by a line break.

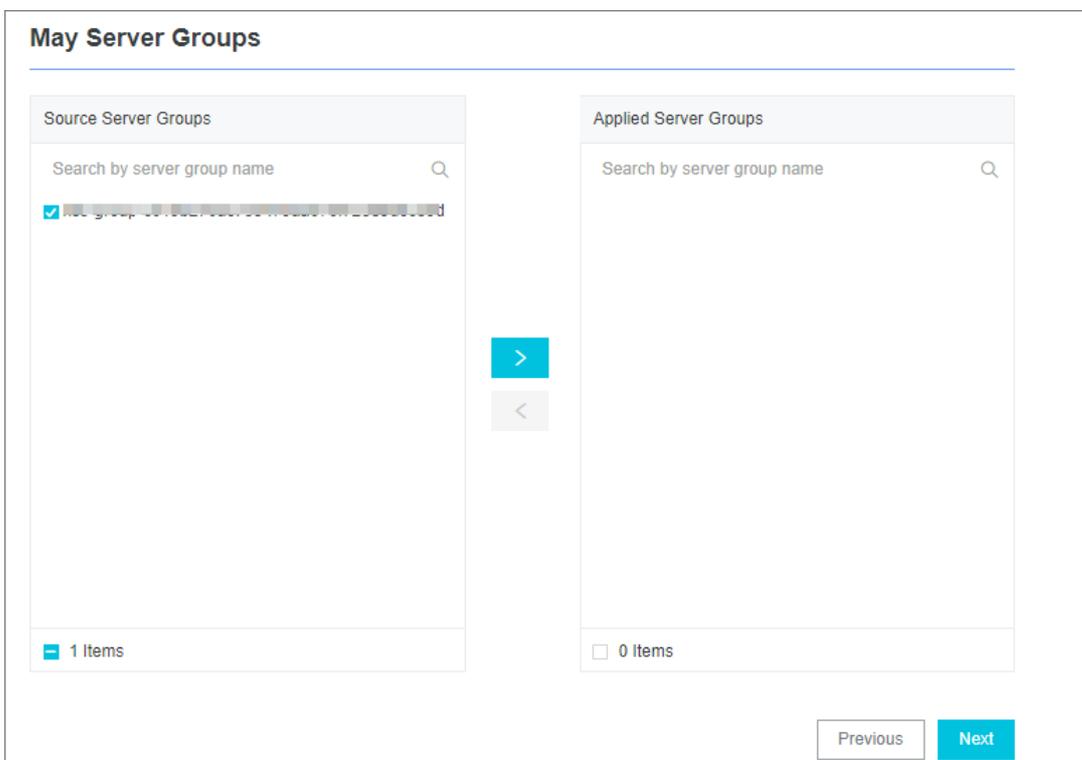
Logtail does not extract log fields in this mode. The default regular expression is `(.*)` . Logtail records the system time of the current server as the timestamp of a log. You can modify or manage advanced Logtail settings after you have completed the configuration procedure. For more information, see [Manage a Logtail configuration](#).

- Multi-line mode

In the multi-line mode, a regular expression is used to match the first line of a log. The settings in the multi-line mode are similar to the settings in the full regex mode. For more information, see [Use regular expressions to collect logs](#).

Procedure

1. Log on to the Log Service console.
2. Select a data source. Select **Single Row-Text Log**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p>Note The configuration name cannot be modified after it is created.</p>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> Each log file can be collected by using only one Logtail configuration. Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified Single Row-Text Log for the data source, the default mode is Simple Mode. You can change the mode.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory that is specified in the log path is monitored.

7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.

Parameter	Description
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> utf8: indicates UTF-8 encoding. gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> Never: All log files are continuously monitored and never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> Collect logs that meet a condition: Set the condition to <code>Key:level RegEx:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. Filter logs that do not meet a condition: <ul style="list-style-type: none"> Set the condition to <code>Key:level RegEx:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. Set the condition to <code>Key:url RegEx:.^(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangong.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect logs by line.

30.3.1.4.3. Use regular expressions to collect logs

This topic describes how to collect logs by using regular expressions and configure indexes. You can specify the required settings in the Log Service console.

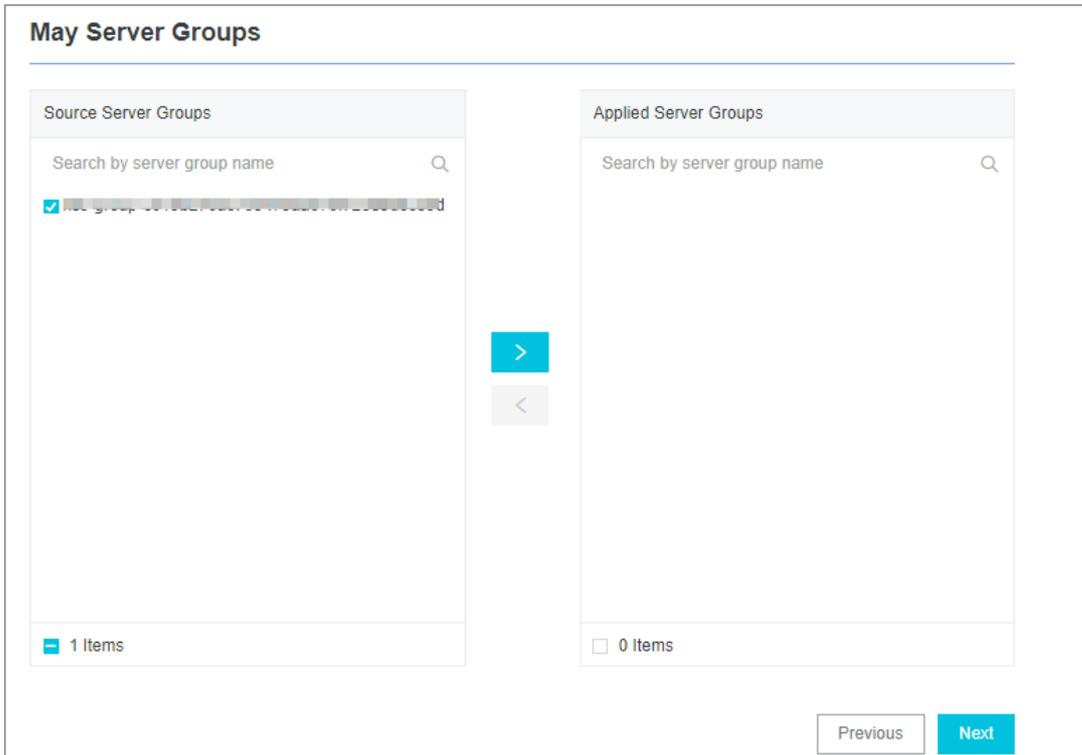
Context

If you need to collect multi-line logs and extract fields from logs, we recommend that you use regular expressions. Log Service can generate a regular expression based on a sample log that you enter in the Import Data wizard. However, you must modify the expression to match fields in the sample log as expected. For more information, see [How do I test a regular expression?](#).

Procedure

1. [Log on to the Log Service console](#).
2. Select a data source. Select **RegEx-Text Log**.

3. Select a Logstore, and then click Next. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click Next. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <p>? Note The configuration name cannot be modified after it is created.</p> </div>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> ▪ Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. ▪ Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ▪ Each log file can be collected by using only one Logtail configuration. ▪ Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	<p>If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.</p>
Mode	<p>If you have specified Regex-Text Log for the data source, the default mode is Full Regex Mode. You can change the mode.</p>
Singleline	<p>The singleline mode is enabled by default. In this mode, logs are separated by line. To collect multi-line logs, such as Java program logs, you must disable the Singleline mode and configure Regex to Match First Line.</p>
Log Sample	<p>Enter a sample log that is retrieved from the data source. Then, Log Service generates a regular expression.</p>
Regex to Match First Line	<p>You can click Auto Generate or Manual. After you enter a sample log entry and click Auto Generate, the system generates a regular expression. If no regular expression is generated, you can switch to the manual mode and enter a regular expression for verification.</p>
Extract Field	<p>To analyze and process specific fields in logs, you can turn on the Extract Field switch. Then, the specified fields are converted to key-value pairs and sent to Log Service. You must specify a regular expression to parse the log content.</p>

Parameter	Description
RegEx	<p>If you turn on the Extract Field switch, you must specify this setting.</p> <ul style="list-style-type: none"> Automatically generate a regular expression You can select the fields to be extracted from the sample log and then click Generate Regular Expression. The system generates a regular expression. Enter a regular expression You can also enter a regular expression. Click Manually to switch to the manual mode. After you enter a regular expression, click Validate to check whether the regular expression can parse the log content. For more information, see How do I test a regular expression?.
Extracted Content	<p>If you turn on the Extract Field switch, you must specify this setting.</p> <p>After a regular expression is automatically generated or manually specified, you must specify the key name for each extracted field.</p>
Use System Time	<p>If you turn on the Extract Field switch, you must specify this setting.</p> <p>If you turn off the Use System Time switch, you must specify a field as the time field and name this field <code>time</code>. After you specify the <code>time</code> field, click Auto Generate in the Time Conversion Format field to automatically parse the time. For more information, see Configure the time format.</p>
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	<p>The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.</p>

7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	<p>Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.</p>
Upload Raw Log	<p>Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.</p>
Topic Generation Mode	<ul style="list-style-type: none"> Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.

Parameter	Description
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> utf8: indicates UTF-8 encoding. gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> Never: All log files are continuously monitored and never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> Collect logs that meet a condition: Set the condition to <code>Key:level RegEx:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. Filter logs that do not meet a condition: <ul style="list-style-type: none"> Set the condition to <code>Key:level RegEx:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. Set the condition to <code>Key:url RegEx:.^(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangkok.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to use regular expressions to collect logs.

30.3.1.4.4. Collect DSV formatted logs

This topic describes how to collect delimiter-separated values (DSV) formatted logs and configure indexes. You can specify the required settings in the Log Service console.

Context

DSV formatted logs use line breaks as boundaries. Each line is a log. The fields of each log are separated with a fixed delimiter. The characters that can be used as delimiters include the tab (\t), space, vertical bar (|), comma (,), and semicolon (;). A field that contains a delimiter must be enclosed in double quotation marks (""), which are used as quotes.

Log formats

Common DSV formatted logs include comma-separated values (CSV) and tab-separated values (TSV) formatted logs.

A delimiter can contain a single character or multiple characters.

Single-character delimiter

You can specify a single-character delimiter and a quote in the console.

- **Delimiter:** The fields of each log are separated with a single-character delimiter, such as the tab (`\t`), vertical bar (`|`), space, comma (`,`), and semicolon (`;`). You can also specify a non-printable character as the delimiter.

 **Note** A double quotation mark (`"`) cannot be used as a delimiter.

If a double quotation mark (`"`) is included in a log but not used as a quote, it must be escaped and processed as double quotation marks (`""`). When Log Service parse logs, it restores double quotation marks (`""`) into a double quotation mark (`"`). You can use a double quotation mark (`"`) on each boundary of a field as a quote. You can also use a double quotation mark (escaped as `""`) in the content of a field. If the use of a double quotation mark (`"`) does not comply with the defined format, you can use the simple mode or full regex mode to parse fields.

If you use commas as delimiters while double quotation marks and commas are included in a field, enclose the field with quotes and escape the double quotation marks into `""` . For example, a processed log is `1999, Chevy,"Venture ""Extended Edition, Very Large""",",5000.00` . The log can be parsed into five fields as follows: `1999` , `Chevy` , `Venture "Extended Edition, Very Large"` , empty field, and `5000.00` .

- **Quote:** If a log field contains delimiters, you must specify a quote to enclose the field. Otherwise, the field cannot be parsed as expected. Log Service parses the content enclosed in quotes as one field. Only delimiters can exist between fields.

You can use one of the following characters as the quote: tab (`\t`), vertical bar (`|`), space, comma (`,`), semicolon (`;`), and non-printable characters.

For example, a log is `1997,Ford,E350,"ac, abs, moon",3000.00` . In this example, the comma (`,`) is used as the delimiter and the double quotation mark (`"`) is used as the quote. The log entry can be parsed into five fields as follows: `1997` , `Ford` , `E350` , `ac, abs, moon` , and `3000.00` . Among the five fields, `ac, abs, moon` enclosed in quotes is regarded as one field.

 **Note** Log Service allows you to use a non-printable character as a delimiter or quote. Non-printable characters are characters whose decimal ASCII codes are in the range of 1 to 31 and 127. If you use a non-printable character as a delimiter or quote, you must find the hexadecimal ASCII code of this character and enter the character in the following format: `0xthe hexadecimal ASCII code of the non-printable character` . For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter `0x01` .

Multi-character delimiter

Each multi-character delimiter can contain two or three characters, such as `||` , `&&&` , and `^_^` . If you specify a multi-character delimiter, Log Service parses logs only based on the delimiter. You do not need to use quotes to enclose log fields.

 **Note** You must ensure that log fields do not contain the delimiter. Otherwise, Log Service cannot parse these fields as expected.

For example, if the delimiter is set to `&&` , the log `1997&&Ford&&E350&&ac&abs&moon&&3000.00` is parsed into the following five fields: `1997` , `Ford` , `E350` , `ac&abs&moon` , and `3000.00` .

Sample log

- **Single-character delimiter**

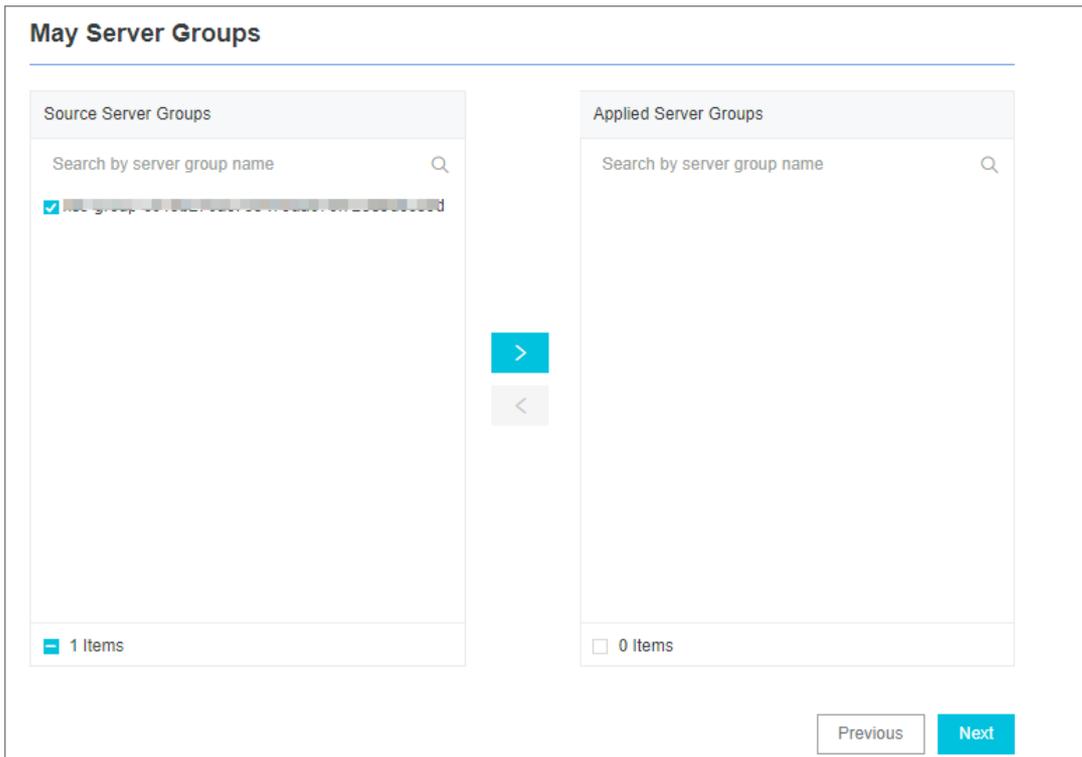
```
05/May/2016:13:30:28,10.10. *. *, "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=*****&Date
=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=***** HTTP/1.
1",200,18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10. *. *, "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=*****&Date
=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=***** HTTP/1.
1",401,23472,aliyun-sdk-java
```

- **Multi-character delimiter**

```
05/May/2016:13:30:28&&10.200. **. **&&POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=*****
&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2BmKd6x7h
AgQ7b1c%3D HTTP/1.1&&200&&18204&&aliyun-sdk-java
05/May/2016:13:31:23&&10.200. **. **&&POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=*****
&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=***** H
TTP/1.1&&401&&23472&&aliyun-sdk-java
```

Procedure

1. [Log on to the Log Service console.](#)
2. Select a data source. Select **Delimiter-Text Log**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p>Note The configuration name cannot be modified after it is created.</p>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p>Note</p> <ul style="list-style-type: none"> Each log file can be collected by using only one Logtail configuration. Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	<p>If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.</p>
Mode	<p>If you have specified Delimiter-Text Log for the data source, the default mode is Delimiter Mode. You can change the mode.</p>
Log Sample	<p>Enter a sample log that is retrieved from the data source. Then, Log Service generates a regular expression.</p>
Delimiter	<p>Select a delimiter.</p> <p>Select a delimiter based on the log format. Otherwise, logs may fail to be parsed.</p> <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p>Note If you use a non-printable character as a delimiter, you must find the hexadecimal ASCII code of this character and enter the character in the following format: <code>0xthe hexadecimal ASCII code of the non-printable character</code> . For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter <code>0x01</code> .</p> </div>

Parameter	Description
Quote	<p>Select a quote.</p> <p>Select a quote based on the log format. Otherwise, logs may fail to be parsed.</p> <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p> Note If you use a non-printable character as a quote, you must find the hexadecimal ASCII code of this character and enter the character in the following format: <code>0xthe hexadecimal ASCII code of the non-printable character</code>. For example, to use the non-printable character whose decimal ASCII code is 1 and hexadecimal ASCII code is 01, you must enter <code>0x01</code>.</p> </div>
Extracted Content	<p>After you enter a sample log and select a delimiter, Log Service extracts log fields based on the delimiter and defines the fields as values. You must specify a key for each value.</p>
Incomplete Entry Upload	<p>Specifies whether to upload a log whose number of parsed fields is less than the number of the specified keys. If you turn on this switch, the log is uploaded. If you turn off this switch, the log is dropped.</p> <p>For example, if you set the delimiter to the vertical bar (), the sample log <code>11 22 33 44 55</code> can be parsed into the following fields: <code>11</code>, <code>22</code>, <code>33</code>, <code>44</code>, and <code>55</code>. You can set the keys to <code>A</code>, <code>B</code>, <code>C</code>, <code>D</code>, and <code>E</code>. If you turn on the Incomplete Entry Upload switch, the <code>55</code> field is uploaded as the value of the <code>D</code> key when Log Service collects the log <code>11 22 33 55</code>. If you turn off the Incomplete Entry Upload switch, Log Service drops the log because the fields and keys do not match.</p>
Use System Time	<p>If you turn on the Extract Field switch, you must specify this setting.</p> <p>If you turn off the Use System Time switch, you must specify a field as the time field and name this field <code>time</code>. After you specify the <code>time</code> field, click Auto Generate in the Time Conversion Format field to automatically parse the time. For more information, see Configure the time format.</p>
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> ◦ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ◦ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	<p>The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.</p>

7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	<p>Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.</p>

Parameter	Description
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ utf8: indicates UTF-8 encoding. ◦ gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ◦ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ◦ Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> ◦ Never: All log files are continuously monitored and never time out. ◦ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> ◦ Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. ◦ Filter logs that do not meet a condition: <ul style="list-style-type: none"> ▪ Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. ▪ Set the condition to <code>Key:url Regex:.*(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangkok.html</code> are not collected.

8. **Configure an index.** Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect DSV formatted logs.

30.3.1.4.5. Collect JSON logs

This topic describes how to use Logtail to collect JSON logs and configure indexes. You can specify the required settings in the Log Service console.

Context

Logtail can parse JSON objects from logs. It extracts the keys and values from the first layer of an object as the names and values of log fields. The valid data types of field values include object, array, and primitive data types such as string or number.

JSON logs can be written in the following two types of structures:

- Object: a collection of key-value pairs.
- Array: an ordered list of values.

Lines of JSON logs are separated with `\n`. Each line is extracted as a single log.

Logtail can parse only JSON logs of the object type. If you want to parse JSON logs of other types, such as JSON arrays, you must use regular expressions to extract the fields or specify the simple mode to collect logs by line.

Sample log

A sample JSON log is as follows:

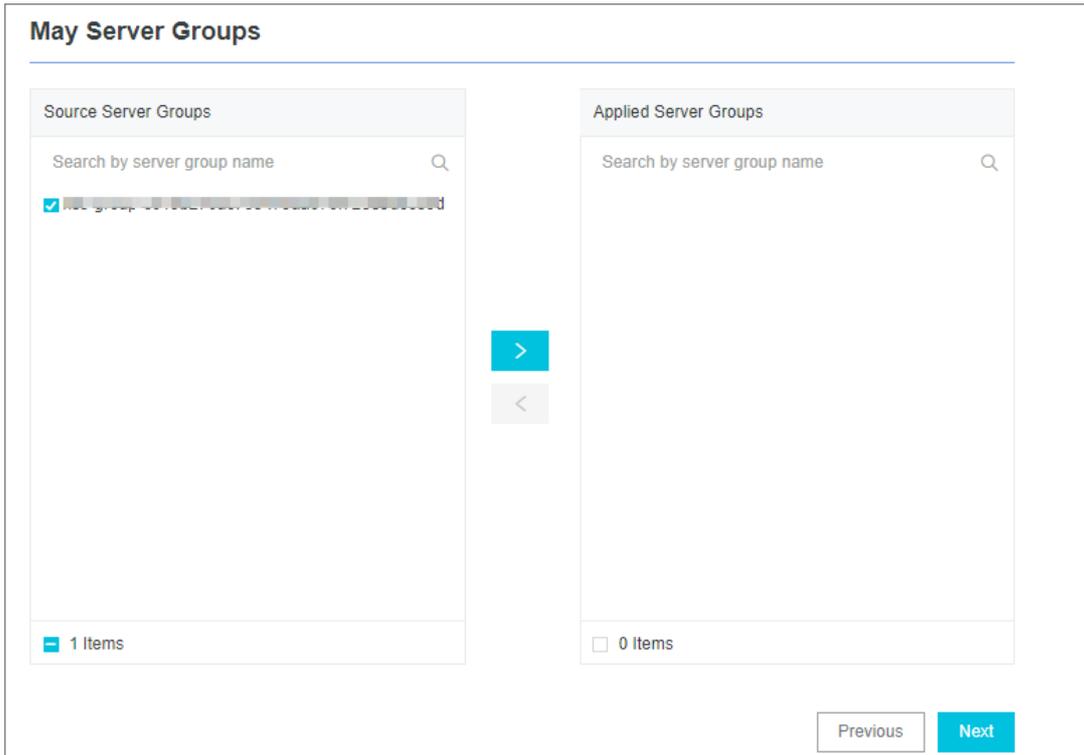
```
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2BmKd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "18204"}, "time": "05/May/2016:13:30:28"}
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2BmKd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.210", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "10204"}, "time": "05/May/2016:13:30:29"}
}
```

Procedure

1. [Log on to the Log Service console](#).
2. Select a data source. Select **JSON-Text Log**.
3. Select a Logstore, and then click Next. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click Next. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p>Note The configuration name cannot be modified after it is created.</p>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> Each log file can be collected by using only one Logtail configuration. Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	<p>If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.</p>
Mode	<p>If you have specified JSON-Text Log for the data source, the default mode is JSON Mode. You can change the mode.</p>
Use System Time	<p>If you turn on the Extract Field switch, you must specify this parameter.</p> <p>If you turn off the Use System Time switch, you must specify a field as the time field and name this field <code>time</code>. After you specify the <code>time</code> field, click Auto Generate in the Time Conversion Format field to automatically parse the time. For more information, see Configure the time format.</p>
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	<p>The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.</p>

7. (Optional)Specify **Advanced Options** and click **Next**.Specify **Advanced Options** based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ <code>utf8</code>: indicates UTF-8 encoding. ◦ <code>gbk</code>: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ◦ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ◦ Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> ◦ Never: All log files are continuously monitored and never time out. ◦ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> ◦ Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. ◦ Filter logs that do not meet a condition: <ul style="list-style-type: none"> ▪ Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. ▪ Set the condition to <code>Key:url Regex:.*(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangkok.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect JSON logs.

30.3.1.4.6. Collect NGINX logs

This topic describes how to collect NGINX logs and configure indexes. You can connect Log Service to NGINX and specify the required settings in the Log Service console.

Context

The NGINX log format and path are specified in the `/etc/nginx/nginx.conf` configuration file.

NGINX log format

In the configuration file, the format of NGINX logs is defined as follows:

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    '$request_time $request_length '
    '$status $body_bytes_sent "$http_referer" '
    '"$http_user_agent";
```

The path of the log file is declared as follows. The "main" portion that follows the path indicates that logs are written in the preceding format.

```
access_log /var/logs/nginx/access.log main
```

Sample log

A sample NGINX log is as follows:

```
192.168.1.2 - - [10/Jul/2015:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1.11.4 Red Hat modified"
```

NGINX log fields

Field	Description
remote_addr	The IP address of the client.
remote_user	The username of the client.
request	The URL and HTTP protocol of the request.
status	The status of the request.
body_bytes_sent	The number of bytes in the response that is returned to the client, excluding the size of the response header.
connection	The serial number of a connection.
connection_requests	The number of requests that are received from a connection.
msec	The time when the log is written. The time is measured in seconds, accurate to milliseconds.
pipe	Indicates whether the request is pipelined. If the request is pipelined, the field value is <code>p</code> . Otherwise, the field value is <code>.</code> .

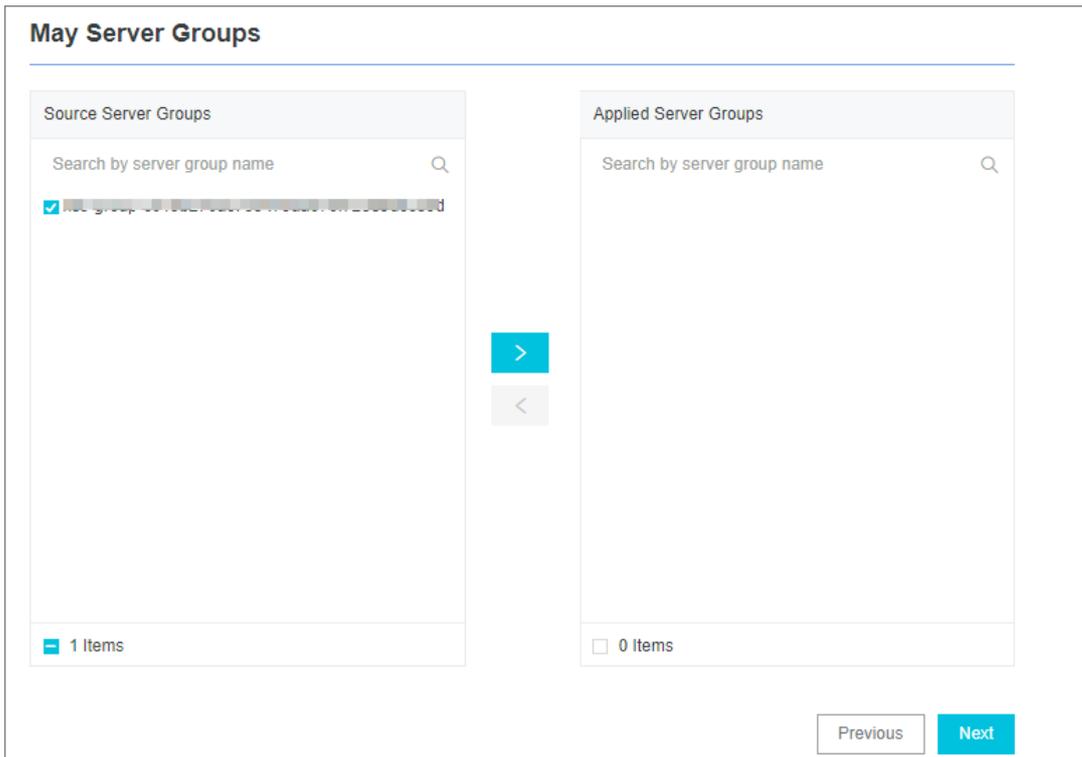
Field	Description
http_referer	The URL of the web page linked to the resource that is being requested.
"http_user_agent"	The browser information of the client. The information must be enclosed by double quotation marks ("").
request_length	The length of the request. The length includes the request line, request header, and request body.
request_time	The time period for which the request is processed. The time period is measured in seconds, accurate to milliseconds. The time period starts when the first byte is read from the client and ends when the log is written after the last byte is sent to the client.
[\$time_local]	The local time in the Common Log Format. The time must be enclosed by brackets [].

Procedure

1. [Log on to the Log Service console.](#)
2. Select a data source. Select **Nginx-Text Log**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p>Note The configuration name cannot be modified after it is created.</p>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is adopted in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> Each log file can be collected by using only one Logtail configuration. Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs.
Mode	If you have specified Nginx-Text Log for the data source, the default mode is NGINX Configuration Mode. You can change the mode.
NGINX Log Configuration	Enter the log configuration section that is specified in a standard NGINX configuration file. The section starts with <code>log_format</code> .
NGINX Key	Log Service reads the keys of NGINX logs.
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.

Parameter	Description
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ utf8: indicates UTF-8 encoding. ◦ gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ◦ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ◦ Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> ◦ Never: All log files are continuously monitored and never time out. ◦ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> ◦ Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. ◦ Filter logs that do not meet a condition: <ul style="list-style-type: none"> ▪ Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. ▪ Set the condition to <code>Key:url Regex:. *^(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiankong.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect NGINX logs.

30.3.1.4.7. Collect IIS logs

This topic describes how to collect Internet Information Services (IIS) logs and configure indexes. You can specify the required settings in the Log Service console.

Context

To meet log analysis requirements, we recommend that you use the W3C Extended Log File Format. To use this format, click **Select Fields** in the IIS Manager, and then select **sc-bytes** and **cs-bytes** in the Standard Fields list.

Log format

The W3C Extended Log File Format is as follows:

```
logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerName, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie, Referer, ProtocolVersion, Host, HttpSubStatus"
```

• Field prefixes

Prefix	Description
s-	The server action.
c-	The client action.
cs-	The client-to-server action.
sc-	The server-to-client action.

• Fields

Field	Description
date	The date on which the client sends the request.
time	The time when the client sends the request.
s-sitename	The Internet service name and instance number of the site visited by the client.
s-computername	The name of the server on which the log is generated.
s-ip	The IP address of the server on which the log is generated.
cs-method	The HTTP request method that is used by the client, for example, GET or POST.
cs-uri-stem	The URI resource requested by the client.
cs-uri-query	The query string that follows the question mark (?) in the HTTP request.
s-port	The port number of the server to which the client is connected.
cs-username	The username used by the client to access the server. Authenticated users are referenced as <code>domain\username</code> . Anonymous users are indicated by a hyphen (-).
c-ip	The IP address of the client that sends the request.
cs-version	The protocol version that is used by the client, for example, HTTP 1.0 or HTTP 1.1.

Field	Description
user-agent	The browser that is used by the client.
Cookie	The content of the sent or received cookie. A hyphen (-) is used if no cookie is sent or received.
referer	The site that the client last visited. This site provides a link to the current site.
cs-host	The header name of the host.
sc-status	The HTTP or FTP status code that is returned by the server.
sc-substatus	The HTTP substatus code that is returned by the server.
sc-win32-status	The Windows status code that is returned by the server.
sc-bytes	The number of bytes that are sent by the server.
cs-bytes	The number of bytes that are received by the server.
time-taken	The processing time of the request. Unit: milliseconds.

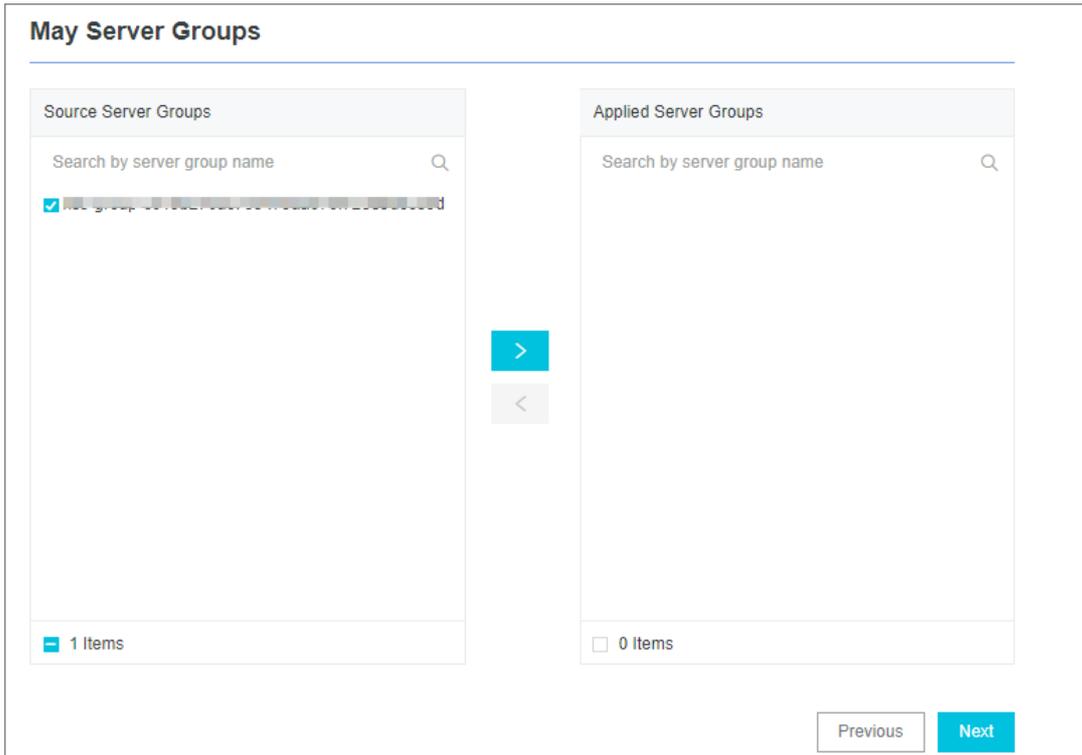
Procedure

1. [Log on to the Log Service console.](#)
2. Select a data source. Select **IIS-Text Log**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p>Note The configuration name cannot be modified after it is created.</p>

Parameter	Description
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <div style="border: 1px solid #add8e6; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> Each log file can be collected by using only one Logtail configuration. Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path. </div>
Docker File	<p>If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs. For more information, see Collect container text logs.</p>
Mode	<p>If you have specified IIS-Text Log for the data source, the default mode is IIS Configuration Mode. You can change the mode.</p>
Log Format	<p>Select the log format of your IIS server logs. Valid values:</p> <ul style="list-style-type: none"> IIS: Microsoft IIS log file format NCSA: NCSA Common log file format W3C: W3C Extended Log File Format
IIS Configuration	<p>Enter the log configuration section that is specified in an IIS configuration file.</p> <ul style="list-style-type: none"> If you select IIS or NCSA, the fields of the IIS log format are preconfigured. If you select W3C, enter the content that is specified for the <code>logFile logExtFileFlags</code> in the configuration file. For more information, see Specify the IIS Configuration field.
IIS Key Name	<p>Log Service reads the keys of IIS logs.</p>
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.

Parameter	Description
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory that is specified in the log path is monitored.

7. Specify the IIS Configuration field.

i. Open the IIS configuration file.

- Default path of the IIS5 configuration file: *C:\WINNT\system32\inetrv\MetaBase.bin*
- Default path of the IIS6 configuration file: *C:\WINDOWS\system32\inetrv\MetaBase.xml*
- Default path of the IIS7 configuration file: *C:\Windows\System32\inetrv\config\applicationHost.config*

ii. Find the `logFile logExtFileFlags` field and copy the text in the quotation marks that follow the field name.

iii. Paste the text into the quotation marks (") in the IIS Configuration field.

8. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ○ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ○ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ○ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ○ <code>utf8</code>: indicates UTF-8 encoding. ○ <code>gbk</code>: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ○ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ○ Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> ○ Never: All log files are continuously monitored and never time out. ○ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.

Parameter	Description
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. Filter logs that do not meet a condition: <ul style="list-style-type: none"> Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. Set the condition to <code>Key:url Regex:.^(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangong.html</code> are not collected.

9. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect IIS logs.

30.3.1.4.8. Collect Apache logs

This topic describes how to collect Apache logs and configure indexes. You can specify the required settings in the Log Service console.

Log formats

The Apache configuration file defines two log formats: combined log format and common log format. You can also customize a log format.

- Syntax of the combined log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

- Syntax of the common log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b"
```

- Syntax of a custom log format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %I %O" customized
```

You must specify the log format, log file directory, and log file name in the Apache configuration file. For example, the following declaration in the configuration file indicates that the combined log format is used. The log file directory is `/var/log/apache2/access_log` and the log file name is `access_log`.

```
CustomLog "/var/log/apache2/access_log" combined
```

Apache log fields

Format string	Key name	Description
%a	client_addr	The IP address of the client in the request.

Format string	Key name	Description
%A	local_addr	The local private IP address.
%b	response_size_bytes	The size of the response. Unit: bytes. If no bytes are sent, the value is "-" .
%B	response_bytes	The size of the response. Unit: bytes. If no bytes are sent, the value is 0.
%D	request_time_msec	The time period for which the request is processed. Unit: milliseconds.
%h	remote_addr	The name of the remote host.
%H	request_protocol_supple	The request protocol.
%l	remote_ident	The identity information that is provided by a remote computer.
%m	request_method_supple	The request method.
%p	remote_port	The port number of the server.
%P	child_process	The ID of the child process.
%q	request_query	The query string. If it does not exist, the value is an empty string.
"%r"	request	The request, which includes the method name, address, and HTTP protocol.
%s	status	The HTTP status code for the response.
%>s	status	The HTTP status code for the final response.
%f	filename	The name of the requested file.
%k	keep_alive	The number of keep-alive requests.
%R	response_handler	The type of the handler that generates the response on the server.
%t	time_local	The local time when the server receives the request.
%T	request_time_sec	The time period for which the request is processed. Unit: seconds.
%u	remote_user	The username that you used to log on to the client.
%U	request_uri_supple	The requested URL, excluding query strings.
%v	server_name	The name of the server.
%V	server_name_canonical	The server name based on the UseCanonicalName setting.
%I	bytes_received	The number of bytes that are received by the server. To use this field, you must enable the mod_logio module.

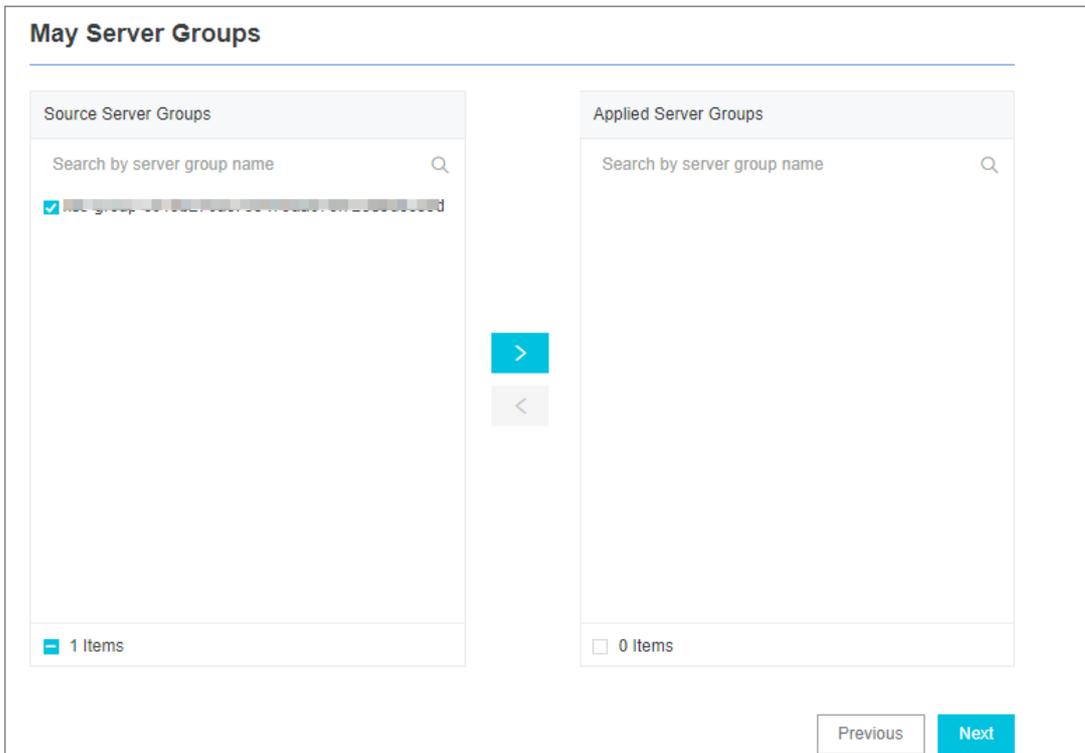
Format string	Key name	Description
%O	bytes_sent	The number of bytes that are sent by the server. To use this field, you must enable the mod_logio module.
"%{User-Agent}i"	http_user_agent	The information about the client.
"%{Referer}i"	http_referer	The URL of the web page linked to the resource that is being requested.

Sample log

```
192.168.1.2 - - [02/Feb/2016:17:44:13 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://localhost/x1.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

Procedure

1. [Log on to the Log Service console.](#)
2. Select a data source. Select **Apache-Text Log**.
3. Select a Logstore, and then click Next. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click Next. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Create a Logtail configuration. The following table lists the Logtail parameters.

Parameter	Description
Config Name	<p>The configuration name must be 3 to 128 characters in length, and can contain lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit.</p> <p> Note The configuration name cannot be modified after it is created.</p>
Log Path	<p>The directory and name of the log file.</p> <ul style="list-style-type: none"> ○ The specified log file name can be a complete file name or a file name that contains wildcards. ○ Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. <ul style="list-style-type: none"> ▪ Example 1: <code>/apsara/nuwa/ ... /*.log</code> indicates the files whose extension is <code>.log</code> in the <code>/apsara/nuwa</code> directory and its sub-directories are monitored. ▪ Example 2: <code>/var/logs/app_* ... /*.log*</code> indicates the files whose file name contains <code>.log</code> in the following directories are monitored: the sub-directories of the <code>/var/logs</code> directory that match the <code>app_*</code> format and the sub-directories of these matching sub-directories. <p> Note</p> <ul style="list-style-type: none"> ▪ Each log file can be collected by using only one Logtail configuration. ▪ Only the asterisk (<code>*</code>) and question mark (<code>?</code>) can be used as wildcards in the log path.
Docker File	<p>If the log file to be collected is in a Docker container, you can configure the internal path and container tag. Logtail monitors the creation and destruction of the container, filters logs of the container based on the tag, and collects the filtered logs. For more information, see Collect container text logs.</p>
Mode	<p>If you have specified Apache-Text Log for the data source, the default mode is Apache Configuration Mode. You can change the mode.</p>
Log Format	<p>Select a log format based on the format declared in your Apache log configuration file. To facilitate the query and analysis of log data, we recommend that you use a custom Apache log format.</p>
APACHE Logformat Configuration	<p>Enter the log configuration section that is specified in the Apache configuration file. The section starts with <code>LogFormat</code>.</p> <p> Note If the specified Log Format is Common or Combined, the system enters a commonly used syntax of the log format. Check whether the log format is the same as that defined in the Apache configuration file.</p>

Parameter	Description
APACHE Key Name	Log Service reads the keys of Apache logs. Confirm the key names on the Logtail configuration page.
Drop Failed to Parse Logs	<p>Specifies whether to upload logs to Log Service if the logs fail to be parsed.</p> <ul style="list-style-type: none"> ◦ If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. ◦ If you turn off this switch, raw logs are uploaded to Log Service when logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the directory specified in the log path is monitored.

7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ utf8: indicates UTF-8 encoding. ◦ gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ◦ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ◦ Custom: Select a time zone.
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> ◦ Never: All log files are continuously monitored and never time out. ◦ 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.

Parameter	Description
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> ○ Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code> . It indicates that only logs with the severity level of WARNING or ERROR are collected. ○ Filter logs that do not meet a condition: <ul style="list-style-type: none"> ▪ Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code> . It indicates that logs with the severity level of INFO or DEBUG are not collected. ▪ Set the condition to <code>Key:url Regex:.*^(?!.*(healthcheck)).*</code> . It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangkok.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After you complete the settings, you can start to collect Apache logs.

30.3.1.4.9. Configure parsing scripts

This topic describes how to configure log contents for log collection.

Specify a method to separate log lines

A complete access log such as an NGINX access log occupies a line. Separate multiple log entries with line breaks. For example, the following shows two access logs:

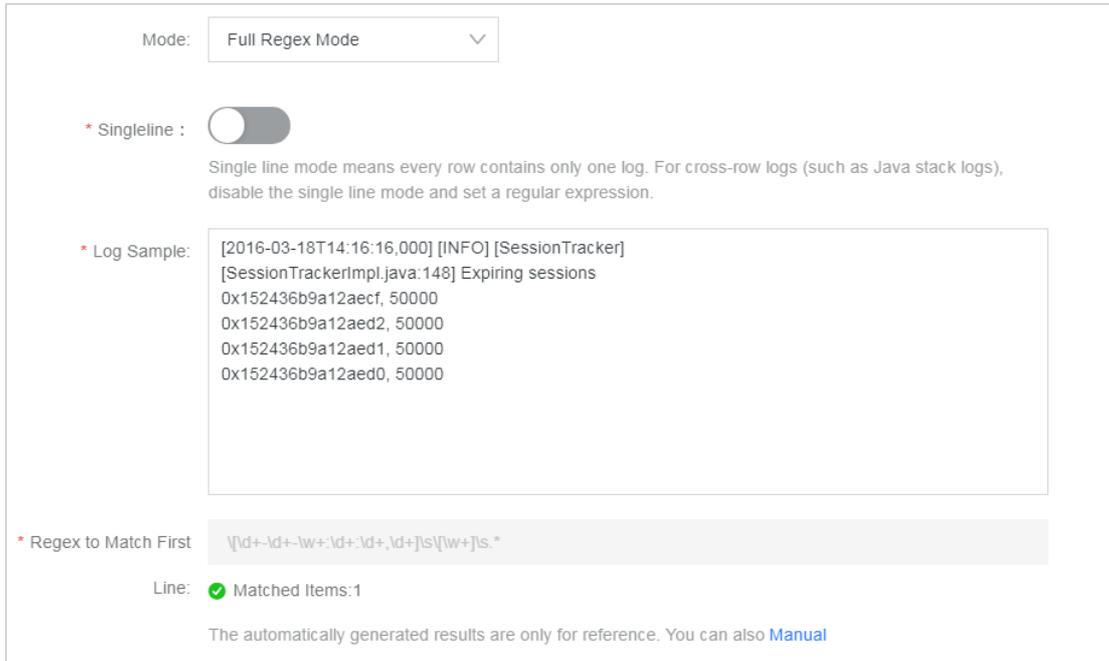
```
10.1.1.1 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
10.1.1.1 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
```

For Java applications, a log entry usually spans several lines. Therefore, log entries are separated based on the identifier at the beginning of each log entry. The following example shows a Java application log.

```
[2016-03-18T14:16:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring sessions
0x152436b9a12aecf, 50000
0x152436b9a12aed2, 50000
0x152436b9a12aed1, 50000
0x152436b9a12aed0, 50000
```

The preceding Java application log entries each start with a time field. The regular expression that matches these time fields is `[\d+-\d+-\w+:\d+:\d+,\d+]\s.*` . You can enter information in the Log Service console as shown in the following figure.

Full regular expression mode



Extract log fields

To conform to the data models of Log Service, a log contains one or more key-value pairs. If you want to extract specific fields for analysis, you must set a regular expression. If you do not want to process the contents of a log, you can treat the log as a key-value pair.

You can determine whether to extract fields from the preceding NGINX access log.

- Extract fields

The regular expression is `(\S+)\s-\s-\s\[([\S+)\s[^\]]+\]\s"(\w+).*`. The extracted fields are `10.1.1.1`, `13/Mar/2016:10:00`, and `GET`.

- Extract all

The regular expression is `(.*)`. The extracted field is `10.1.1.1 -- [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"`.

Specify a log time

To conform to the data models of Log Service, a log must have a time field in the Unix timestamp format. You can use the system time when Logtail collects a log or the time in the log contents as the log time.

For the preceding NGINX access log:

- If you extract the time field from the log contents as the log time, the time is `13/Mar/2016:10:00:10` and the time expression is `%d/%b/%Y:%H:%M:%S`.
- If you use the system time when the log was collected as the log time, the log time is converted into a timestamp.

30.3.1.4.10. Configure the time format

Each log in Log Service must have a timestamp that records the log generation time. When you collect logs from log files, Logtail must extract the timestamp string of each log and parse it into a timestamp. Therefore, you need to specify a timestamp format to facilitate parsing.

In Linux, Logtail supports all time formats provided by the `strptime()` function. If a timestamp string can match one of the time formats that are provided by the `strptime()` function, Logtail can parse and use the timestamp string.

Note

- The log timestamp is accurate to seconds. Therefore, you need to specify seconds in a time format, without the need for other information such as milliseconds or microseconds.
- In addition, you need to configure the time field rather than other information.

Common log time formats supported by Logtail

The timestamp strings of logs have diverse formats. To make configuration easier, the following table lists the common log time formats supported by Logtail:

Format	Description	Example
%a	The abbreviation of a day in a week.	Fri
%A	The day in a week.	Friday
%b	The abbreviation of a month.	Jan
%B	The month name.	January
%d	The numerical day in a month. Valid values: 01 to 31.	07 and 31
%h	The abbreviation of a month. The format is equivalent to %b .	Jan
%H	The hour in the 24-hour format.	22
%I	The hour in the 12-hour format.	11
%m	The numerical month.	08
%M	The numerical minute. Valid values: 00 to 59.	59
%n	A line break.	Line break
%p	The local time in the a.m. or p.m. format.	AM and PM
%r	The time in the 12-hour format. The format is equivalent to %I:%M:%S %p .	11:59:59 AM
%R	The time includes hours and minutes. The format %R is equivalent to %H:%M .	23:59
%S	The numerical second. Valid values: 00 to 59.	59
%t	A tab.	Tab
%y	The two-digit numerical year. Valid values: 00 to 99.	04 and 98
%Y	The four-digit numerical year.	2004 and 1998

Format	Description	Example
%C	The numerical century. Valid values: 00 to 99.	16
%e	The numerical day in a month. Valid values: 1 to 31. A single digit is preceded by a space.	7 and 31
%j	The numerical day in a year. Valid values: 001 to 366.	365
%u	The numerical day in a week. Valid values: 1 to 7, in which 1 represents Monday.	2
%U	The numerical week in a year. Sunday is the first day of a week. Valid values: 00 to 53.	23
%V	The numerical week in a year. Monday is the first day of a week. If a week has four or more days that start from January 1, the week is treated as the first week. Otherwise, the next week is treated as the first week. Valid values: 01 to 53.	24
%w	The numerical day in a week. Valid values: 0 to 6, in which 0 represents Sunday.	5
%W	The numerical week in a year. Monday is the first day of a week. Valid values: 00 to 53.	23
%c	The standard date and time.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact expression.
%x	The standard date.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact time expressions.
%X	The standard time.	To specify more information such as the long date and short date, you can use the preceding formats to provide exact expression.
%s	The Unix timestamp.	1476187251

Example

The following table lists the common log time formats, examples, and corresponding time expressions.

Log time format	Example	Time expression
Custom	2017-12-11 15:05:07	%Y-%m-%d %H:%M:%S
Custom	[2017-12-11 15:05:07.012]	[%Y-%m-%d %H:%M:%S]

Log time format	Example	Time expression
RFC822	02 Jan 06 15:04 MST	%d %b %y %H:%M
RFC822Z	02 Jan 06 15:04 -0700	%d %b %y %H:%M
RFC850	Monday, 02-Jan-06 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC1123	Mon, 02 Jan 2006 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC3339	2006-01-02T15:04:05Z07:00	%Y-%m-%dT%H:%M:%S
RFC3339Nano	2006-01-02T15:04:05.999999999Z07:00	%Y-%m-%dT%H:%M:%S

30.3.1.4.11. Import historical logs

Logtail collects incremental logs by default. If you want to import historical logs, use the historical log importing feature of Logtail.

Prerequisites

To collect logs from Linux servers, use Logtail 0.16.15 or later. To collect logs from Windows servers, use Logtail 1.0.0.1 or later. To ensure successful log collection, update Logtail to the latest version.

Context

Logtail collects log files based on events. The system captures events by detecting or polling files for changes at intervals. Additionally, Logtail can load events from local files to trigger log collection. Logtail implements historical log collection based on these local events.

You can import historical log files from the Logtail installation directory. The location of the directory varies based on the operating system.

- Linux: `/usr/local/ilogtail`
- Windows:
 - 32-bit: `C:\Program Files\Alibaba\Logtail`
 - 64-bit: `C:\Program Files (x86)\Alibaba\Logtail`

Note

- The maximum interval between the time a local event is generated and the time the local event is imported is one minute.
- Loading local configurations is a special action. Therefore, Logtail sends the `LOAD_LOCAL_EVENT_ALARM` alert to your server to notify you of this action.
- If you want to import a large number of log files, we recommend that you modify the Logtail startup configuration to increase the upper limit of CPU to 2.0 GHz or more and the upper limit of the memory size to 512 MB or more. For more information, see [Set Logtail startup parameters](#).

Procedure

1. Configure log collection. If a collection configuration is only used to import historical log files, you can specify a collection directory that does not exist. For more information, see [Configure text log collection](#).
2. Obtain a unique identifier for a collection configuration. Obtain the unique identifier in the `user_log_config.json` file stored in the installation directory of Logtail. In Linux, use the `grep` command in the directory to query the unique identifier. In Windows, use tools such as Notepad to query the unique identifier. To query a unique identifier in a Linux operating system, run the following command:

```
grep "###" /usr/local/ilogtail/user_log_config.json | awk '{print $1}'
###1.0##log-config-test$multi"
###1.0##log-config-test$ecs-test"
###1.0##log-config-test$metric_system_test"
###1.0##log-config-test$redis-status"
```

3. Add local events. Local events are stored in the *local_event.json* file that resides in the installation directory of Logtail. The file is in the JSON format. The syntax is:

```
[
  {
    "config": "${your_config_unique_id}",
    "dir": "${your_log_dir}",
    "name": "${your_log_file_name}"
  },
  {
    ...
  }
  ...
]
```

Parameters

Parameter	Description	Example
config	The unique identifier that is obtained in Step 2.	###1.0##log-config-test\$ecs-test
dir	The directory where logs are stored. Note The directory cannot end with a slash (/).	/data/logs
name	The name of a log file. Wildcards are supported.	For example, access.log.2018-08-08 and access.log*

Note To prevent Logtail from loading invalid JSON files, save local event configurations to a temporary file, edit the configurations in the temporary file, and copy the contents to the *local_event.json* file.

Configuration examples

In a Windows system, you can use tools such as Notepad to add local events to the *local_event.json* file. In a Linux system, add local events as follows:

```
$ cat /usr/local/ilogtail/local_event.json
[
  {
    "config": "##1.0##log-config-test$ecs-test",
    "dir": "/data/log/",
    "name": "access.log."
  },
  {
    "config": "##1.0##log-config-test$ecs-test",
    "dir": "/tmp",
    "name": "access.log.2017-08-09"
  }
]
```

What's next

- Check whether Logtail has loaded configurations

After you save the `local_event.json` file, Logtail loads the configuration file to the memory within one minute and clears the contents of the `local_event.json` file.

To check whether Logtail has read local events, use the following methods:

- If the contents of the `local_event.json` file are cleared, it indicates that Logtail has read the local events.
- Check whether the `ilogtail.LOG` file in the Logtail installation directory contains the `process local event` keywords. If the contents of the `local_event.json` are cleared and these keywords cannot be found, the local configuration file may be screened due to invalid contents.
- Check whether the configuration is loaded but no data is collected

Possible causes are as follows:

 - The configuration is invalid.
 - The local `config` file does not exist.
 - The log file does not exist in the path specified in the collection configuration.
 - The log file has been collected by Logtail.

30.3.1.4.12. Generate a topic

This topic describes how to generate a topic in the Log Service console. After you generate a topic, you can use the topic to group logs. You can specify topics for logs when these logs are written. You can use a topic as a filter when you query logs.

Topic generation modes

You can set a topic when you use Logtail to collect logs or when you use API operations or SDKs to upload logs. The following topic generation modes are available in the Log Service console: **Null - Do not generate topic**, **Server Group Topic Attributes**, and **File Path RegEx**.

- **Null - Do not generate topic**

When you configure Logtail in the Log Service console to collect text logs, the default topic generation mode is **Null - Do not generate topic**. In this mode, no topic is generated and query logs without specifying a topic.

- **Server Group Topic Attributes**

You can use this mode to identify logs that are generated from multiple servers. Logs from multiple servers can be stored in the same file or directory. To identify these logs based on topics during log collection, you can create server groups and add the servers into different groups. When you create server groups, you must specify a unique topic attribute for each server group and set **Topic Generation Mode** to **Server Group Topic Attributes**. After you complete the configuration, apply the Logtail settings to the server groups.

If the **Server Group Topic Attributes** mode is selected, Logtail uploads the topic attribute of each server group as topics to Log Service. When you query logs, you must specify the topic of the target server group as a filter.

● **File Path RegEx**

- You can use this mode to differentiate between logs that are generated by multiple users or instances. If Log Service stores logs in different directories for different users or instances, duplicate sub-directory names or log file names may exist in these directories. As a result, Log Service cannot identify the source of logs. You can select **File Path RegEx** in the **Topic Generation Mode** field. Enter a regular expression that matches an absolute file path, and set an instance name as a topic.
- If you select **File Path RegEx**, Logtail uses an instance name as the topic of the logs that Logtail uploads to Log Service. The topic generated varies based on your directory structure and configuration. You must specify an instance name as a topic when you query logs. For example, the following directory structure includes directories that each store logs generated by different users or instances:

```

/logs
| - /userA/serviceA
| - service.log
| - /userB/serviceA
| - service.log
| - /userC/serviceA
| - service.log
    
```

- If you want to extract multiple separate fields from a file path, use a multi-layer extraction method of `<key> ?P <key>`. The value of the key can contain lowercase letters and digits. For example:

```

/home/admin/serviceA/userB/access.log
\\home\\admin\\(? P<service>[^\]+)/(? P<user>[^\]+)/. *
    
```

The following custom tags are created for logs:

```

"__tag__ : service : serviceA"
"__tag__ : user : userB"
    
```

 **Note** Logtail 0.16.19 and later are supported.

- If you specify the `/logs` file path and the `service.log` file name in a regular expression, Logtail collects logs from the preceding directories that contain the `service.log` file and uploads the logs to Log Service. However, Log Service cannot identify the log source based on log contents. You can select **File Path RegEx** in the **Topic Generation Mode** field, and enter the `\\(.*)\\serviceA\\.*` regular expression to extract instance names. After the configuration is complete, the following topics are generated for logs in different directories: `userA`, `userB`, and `userC`. You can specify a topic as a filter to query logs.

 **Note** You must escape the forward slashes (/) in the file path that the regular expression contains.

● **Static topic generation**

You can select **File Path RegEx** in the **Topic Generation Mode** field, and enter `customized:// + custom topic` in the **Custom RegEx** field.

 **Note** Logtail 0.16.21 and later are supported.

Set a log topic

1. Configure Logtail in the Log Service console. For more information, see [Configure text log collection](#). To set the topic generation mode to **Server Group Topic Attributes**, go to the **Topic** section on the server group creation or modification page.
2. In the Logtail Configuration for Data Import step, click **Advanced Options** and select a **topic generation mode**.

30.3.1.5. Custom plug-ins

Context

Log Service allows you to collect text logs and system logs through Logtail. Logtail supports connections with multiple data sources, such as HTTP or MySQL query results and MySQL binary logs.

You can collect HTTP request data and upload the processing results to Log Service in real time to check service availability check and continuous availability monitoring. You can configure MySQL query results as the data source, and then synchronize incremental data based on custom IDs or time. You can also configure an SQL data source to synchronize MySQL binary logs, subscribe to database changes, and query or analyze logs in real time.

 **Note** This feature is only supported on Linux and must be used together with Logtail. 0.16.0 or later versions. For more information, see [Install Logtail in Linux](#).

Procedure

1. Configure a method that is used to collect logs from the data source.

Different Logtail configurations for different data sources. Select a Logtail configuration according to your data source.

- [Collect MySQL binary logs](#)
- [Collect MySQL query results](#)
- [Collect HTTP request data](#)
- [Collect container standard outputs](#)

2. Configure a processing method.

Logtail provides multiple processing methods for binary logs, MySQL query results, NGINX monitoring data, and HTTP input sources. You can configure multiple processing methods for a single input source. Each input source supports all processing methods. Logtail runs the configured processing methods in sequence.

For more information, see [Process collected data](#).

3. Apply the configurations to the machine group.

After you configure the collection and processing methods, apply them to the specified machine group. Then, Logtail automatically applies the configurations and starts data collection.

30.3.1.5.1. Collect MySQL binary logs

Logtail acts as a MySQL slave that collects binary logs from a MySQL master. This improves log collection performance. Logtail collects binary logs in a similar way to Alibaba Canal.

Features

- Allows you to collect incremental data of databases through binary logs to improve performance. Supports MySQL databases such as ApsaraDB for RDS databases.
- Supports multiple database filters, such as regular expressions.

- Allows you to set binary log file positions.
- Allows you to records synchronization statuses by using the checkpoint mechanism.

Limits

- MySQL binary logs are available only for Logtail 0.16.0 or later that you install on Linux. For more information about how to update Logtail and view Logtail versions, see [Install Logtail in Linux](#).
- Binary logs in the ROW format must be enabled for MySQL databases. Binary logs in the ROW format are enabled for RDS instances by default.

```
# Check whether binary logs are enabled.
mysql> show variables like "log_bin";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin      | ON   |
+-----+-----+
1 row in set (0.02 sec)

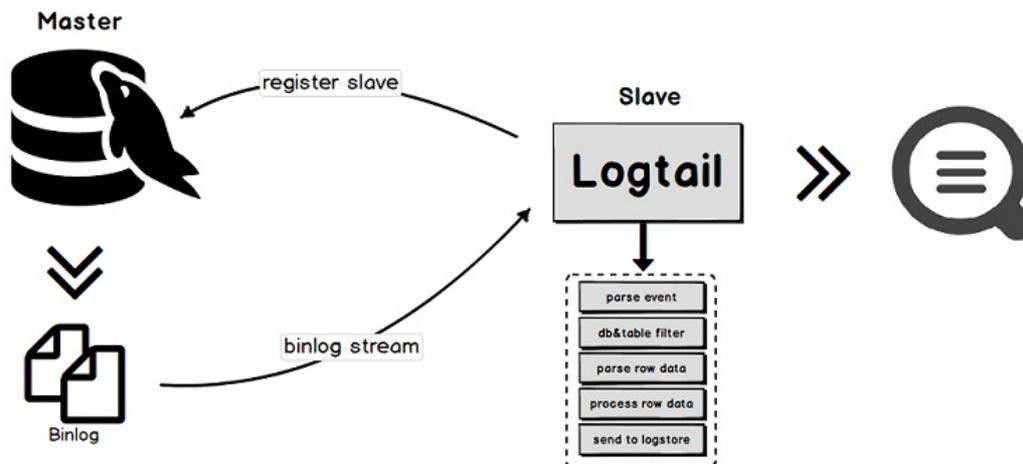
# View the format of binary logs.
mysql> show variables like "binlog_format";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW  |
+-----+-----+
1 row in set (0.03 sec)
```

- Each server ID must be unique. Make sure that the ID of each slave to be synchronized is unique.
- Limits on RDS instances
 - Logtail cannot be installed on an RDS instance. You must install Logtail on an ECS instance that can communicate with the target RDS instance.
 - Secondary RDS databases do not support binary log collection. You must configure a primary RDS database for binary log collection.

Procedure

Logtail enables communication between master and slave MySQL servers. The process is as follows:

1. Logtail acts as a MySQL slave and sends dump requests to the MySQL master.
2. After the dump requests are received, the MySQL master delivers its binary logs to Logtail in real time.
3. Logtail then parses and filters binary logs, and uploads the results to Log Service.



Scenarios

The MySQL binary log feature is suitable for scenarios that require high-performance synchronization of large amounts of data. The following lists some example scenarios:

- Query and analyze incremental data of databases in real time.
- Audit operations performed on databases.
- Use Log Service to perform custom queries and analysis on database updates, visualize the query and analysis results, transform data for stream processing, export log data to MaxCompute for offline computing, and export log data to OSS for long-term storage.

Data reliability

We recommend that you enable the global transaction identifier (GTID) feature of MySQL database servers and upgrade Logtail to version 0.16.15 or later. This avoids repeated data collection in case of a master/slave switchover and ensures data reliability.

- **Incomplete collection of data:** If the network between Logtail and the MySQL server is broken for a long period of time, data may not be completely collected.

A MySQL binary log plug-in acts as a MySQL slave to collect binary logs from the master. Logtail establishes a connection with the master server to obtain data from the server. If the network between Logtail and the master node is broken, the master node still generates new binary logs and deletes expired binary logs. After the network is restored and Logtail is reconnected to the master, Logtail uses the last checkpoint to request binary log data from the master. However, if the network has been disconnected for an extended period of time, the data after the checkpoint may have been deleted. In this case, the recovery mechanism specifies the new point at which Logtail resumes collecting binary logs. The new point at which Logtail resumes collecting binary logs is the most recent binary log file position. If a network is broken for an extended period of time, data collection may be incomplete between the checkpoint and the new data collection point.

- **Repeated data collection:** If the binary log numbers on the master and slave are different and a master/slave switchover occurs, repeated data collection may occur.

When the MySQL master-slave synchronization is configured, the master synchronizes the generated binary log data to the slave, and the slave stores the received binary log data to the local binary log file. If the binary log numbers on the master and slave are different, a master/slave switchover occurs. In this case, the mechanism that uses a binary log file name and an offset as the checkpoint will cause repeated data collection.

For example, a piece of data ranges from (binlog.100, 4) to (binlog.105, 4) on the master while ranges from (binlog.1000, 4) to (binlog.1005, 4) on the slave. Logtail has obtained the data from the master and updated the checkpoint to (binlog.105, 4). In this case, if a master/slave switchover occurs with no exception, Logtail continues to obtain binary logs from the new master based on the local checkpoint (binlog.105, 4). The new master returns the data ranging from (binlog.1000, 4) to (binlog.1005, 4) to Logtail because the numbers of these data on the new master are greater than the numbers of data requested by Logtail, which causes repeated collection of binary log data.

Parameter

The type of input sources is `service_canal`.

Parameter	Type	Required	Description
Host	String	No	The IP address of the host where the database is located. Default value: 127.0.0.1.
Port	Integer	No	The port number you can use to connect to the database. Default value: 3306.
User	String	No	<p>The database username. Default value: root.</p> <p>Make sure that the configured user has the read permissions on the database from which data is collected and the MySQL REPLICATION permission. Example:</p> <pre>CREATE USER canal IDENTIFIED BY 'canal'; GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *. * TO 'canal'@'%'; -- GRANT ALL PRIVILEGES ON *. * TO 'canal'@'%'; FLUSH PRIVILEGES;</pre>
Password	String	No	<p>The database password. This parameter is empty by default.</p> <p>If you require a high level of security, we recommend that you set the password to xxx. After configurations are synchronized to the local server, find the Password parameter in the <code>/usr/local/ilogtail/user_log_config.json</code> file and modify the value.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> After you modify the password, use the <code>sudo /etc/init.d/ilogtailed stop; sudo /etc/init.d/ilogtailed start</code> command to restart Logtail. If you modify the value of the Password parameter on the web and synchronize configurations to the local server, configurations on the local server are overwritten. You can change the configurations on the local server in the future. </div>

Parameter	Type	Required	Description
ServerID	Integer	No	<p>The ID of a MySQL slave whose role Logtail assumes. Default value: 125.</p> <p> Note In a MySQL database, each ID must be unique. Otherwise, synchronization fails.</p>
IncludeTables	String array	Yes	<p>The names of matched tables. Each value contains a database name and a table name, for example, <code>test_db.test_table</code>. You must specify a regular expression for the parameter. Logtail does not collect incremental data from tables whose names do not match the regular expression. To collect incremental data from all tables of a database, set the value of the IncludeTables parameter to <code>.*\..*</code>.</p> <p> Note If exact matching is required, start a regular expression with a caret (<code>^</code>) and end with <code>\$</code>. For example: <code>^test_db\\.test_table\$</code>.</p>
ExcludeTables	String array	No	<p>The names of excluded tables expressed as a regular expression. The name of a table must include the name of the database to which the table belongs, such as <code>test_db.test_table</code>. If a table meets one of the conditions specified in the parameter, the table will not be collected. If you leave this parameter empty, incremental data from all tables is collected.</p> <p> Note If exact matching is required, add <code>^</code> to the beginning of a regular expression and <code>\$</code> to the end. Example: <code>^test_db\\.test_table\$</code>.</p>

Parameter	Type	Required	Description
StartBinName	String	No	<p>The name of the first binary log file Logtail collects. If you leave the parameter empty, Logtail collects binary log files that are generated from the current time.</p> <p>To collect data from a specific location, view the name of the current binary log file and the file offset. Then, set <code>StartBinName</code> and <code>StartBinLogPos</code> to corresponding values.</p> <p>Example:</p> <pre># Set StartBinName to "mysql-bin.000063" and StartBinLogPos to 0. mysql> show binary logs; +-----+-----+ Log_name File_size +-----+-----+ mysql-bin.000063 241 mysql-bin.000064 241 mysql-bin.000065 241 mysql-bin.000066 10778 +-----+-----+ 4 rows in set (0.02 sec)</pre> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note</p> <p>If you set the <code>StartBinName</code> parameter, the first collection will generate a large amount of traffic.</p> </div>
StartBinLogPos	Integer	No	The offset of the first binary log file that is collected. Default value: 0.
EnableGTID	Boolean	No	Specifies whether to add GTID . Default value: true. If the value is false, no GTID is added to uploaded data.
EnableInsert	Boolean	No	Specifies whether to collect log events triggered by INSERT operations. Default value: true. If the value is false, INSERT events are not collected.
EnableUpdate	Boolean	No	Specifies whether to collect UPDATE events. Default value: true. If the value is false, UPDATE events are not collected.
EnableDelete	Boolean	No	Specifies whether to collect DELETE events. Default value: true. If the value is false, DELETE events are not collected.
EnableDDL	Boolean	No	Specifies whether to collect data definition language (DDL) events. Default value: false. If the value is false, DDL events are not collected.
Charset	String	No	The encoding method. Default value: <code>utf-8</code> .

Parameter	Type	Required	Description
TextToString	Boolean	No	Specifies whether to convert data of the text type to a string. Default value: false.
PackValues	Boolean	No	<p>Specifies whether to package event data into the JSON format. Default value: false. If the value is false, event data is not packaged. If this feature is enabled, Logtail packages event data into the data and old_data fields in the JSON format. The old_data field is available only for ROW_UPDATE events.</p> <p>For example, a table has three fields named c1, c2, and c3. If this feature is disabled, the ROW_INSERT event data contains three fields c1, c2, and c3. If this feature is enabled, c1, c2, and c3 are combined into one data field and the value is <code>{"c1": "...", "c2": "...", "c3": "..."} .</code></p> <p> Note This parameter is available only for Logtail 0.16.19 or later.</p>
EnableEvent Meta	Boolean	No	<p>Specifies whether to collect event metadata. Default value: false. If the value is false, event metadata is not collected. The metadata of binary log events includes event_time , event_log_position , event_size , and event_server_id .</p> <p> Note This parameter is available only for Logtail 0.16.21 or later.</p>

Procedure

Synchronize data from tables whose names do not end with `_inner` in the `user_info` RDS database.

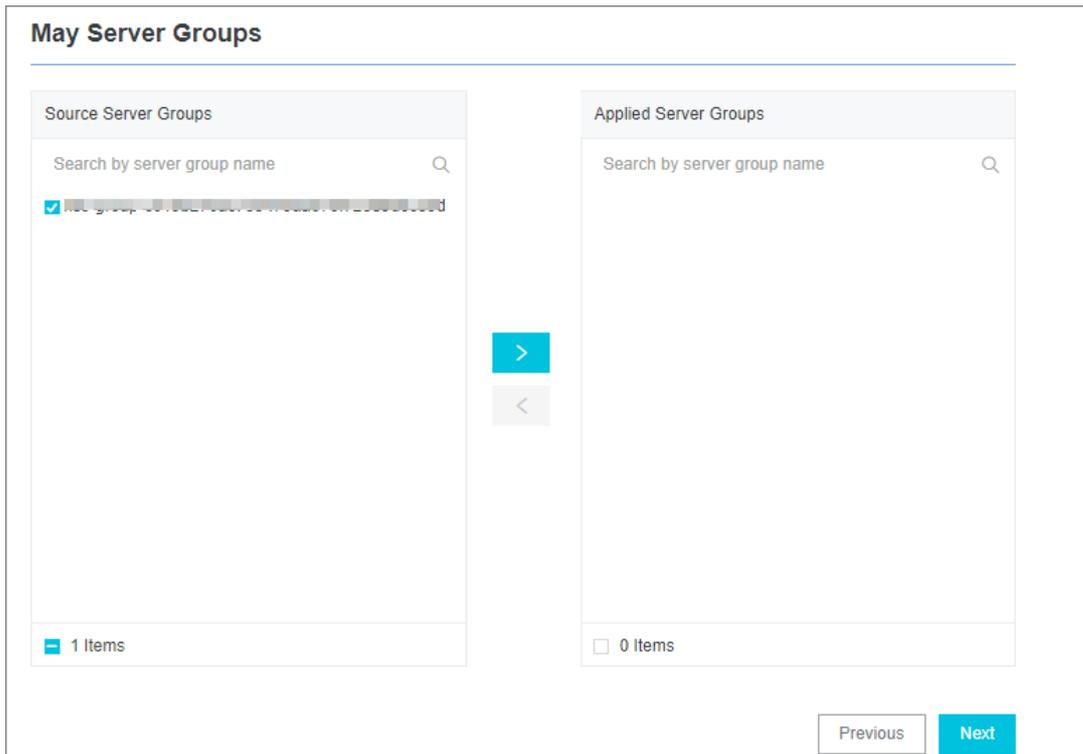
1. [Log on to the Log Service console.](#)
2. Select an input source. Click **Import Data**. On the **Import Data** page that appears, select **MYSQL BinLog**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Configure a data source. Set the Config Name and Plug-in Config fields.

In the **Plug-in Config** field, a default configuration template is specified. Change the parameters in the template based on your business requirements.

```
{
  "inputs": [
    {
      "type": "service_canal",
      "detail": {
        "Host": "*****.mysql.rds.aliyuncs.com",
        "Port": 3306,
        "User": "root",
        "ServerID": 56321,
        "Password": "*****",
        "IncludeTables": [
          "user_info\\..*"
        ],
        "ExcludeTables": [
          ".*\\S+_inner"
        ],
        "TextToString": true,
        "EnableDDL": true
      }
    }
  ]
}
```

- (Required) *inputs* specifies the collection configuration. You must configure statements based on your

data source.

- (Optional) *processors* specifies the processing method. For more information about how to set a processing method, see [Configure data processing methods](#).
7. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).
 8. (Optional) Modify local configuration.

If you do not enter an actual URL, account name, password, and other details on the **Specify Data Source** page, you must complete these details after collection configurations are synchronized to the local server.

 **Note** If you enter actual details, skip this step.

- i. Log on to the server where Logtail is installed, find the `service_canal` keyword in the `/usr/local/ilogtail/user_log_config.json` file, and set related fields. These fields include `Host`, `User`, and `Password`.
- ii. Run the following command to restart Logtail:

```
sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
```

The binary log collection configuration is completed. If changes are made to your database, Logtail will immediately collect the updated data and upload it to Log Service.

 **Note** Logtail collects incremental binary log data. If no data is collected, check whether changes are made to the corresponding table in your database after the configuration is updated.

Metadata fields

During binary log collection, some metadata is also uploaded. The following table lists the fields of uploaded metadata.

Field	Description	Example
<code>_host_</code>	The name of the host where the database resides.	<code>*****.mysql.rds.aliyuncs.com</code>
<code>_db_</code>	The name of the database.	<code>my-database</code>
<code>_table_</code>	The name of the table.	<code>my-table</code>
<code>_event_</code>	The type of the event.	<code>row_update</code> , <code>row_insert</code> , and <code>row_delete</code>
<code>_id_</code>	The ID of the current collection. The value starts from 0 and increments by 1 each time a binary log event is collected.	<code>1</code>
<code>_gtid_</code>	The GTID.	<code>7d2ea78d-b631-11e7-8afb-00163e0eef52:536</code>
<code>_filename_</code>	The name of the binary log file.	<code>binlog.001</code>
<code>_offset_</code>	The offset of the binary log file. The value is updated after each COMMIT operation.	<code>12876</code>

Example

After you complete the preceding steps to set a processing method, perform `INSERT`, `UPDATE`, and `DELETE` operations on the `SpecialAlarm` table in the `user_info` database. The following shows the database scheme, database operations, and Logtail collection.

- Schema

```
CREATE TABLE `SpecialAlarm` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `time` datetime NOT NULL,  
  `alarmtype` varchar(64) NOT NULL,  
  `ip` varchar(16) NOT NULL,  
  `count` int(11) unsigned NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `time` (`time`) USING BTREE,  
  KEY `alarmtype` (`alarmtype`) USING BTREE  
) ENGINE=MyISAM AUTO_INCREMENT=1;
```

- Database operations

Perform the `INSERT`, `DELETE`, and `UPDATE` operations on the database.

```
insert into specialalarm (`time`, `alarmType`, `ip`, `count`) values(now(), "NO_ALARM", "10.10. **.***", 55);  
delete from specialalarm where id = 4829235 ;  
update specialalarm set ip = "10.11. **.***" where id = "4829234";
```

Create an index for `zc.specialalarm`.

```
ALTER TABLE `zc`.`specialalarm`  
ADD INDEX `time_index` (`time` ASC);
```

- Sample logs

On the data preview or search page, you can view a sample log that corresponds to each operation.

- `INSERT` statement

```
__source__: 10.30. **.***  
__tag__:__hostname__: iZbp145dd9fccu*****  
__topic__:  
_db_: zc  
_event_: row_insert  
_gtid_: 7d2ea78d-b631-11e7-8afb-00163e0eef52:536  
_host_: *****.mysql.rds.aliyuncs.com  
_id_: 113  
_table_: specialalarm  
alarmtype: NO_ALARM  
count: 55  
id: 4829235  
ip: 10.10. **.***,  
time: 2017-11-01 12:31:41
```

○ DELETE statement

```
__source__: 10.30. **.**
__tag__:__hostname__: iZbp145dd9fccu****
__topic__:
_db__: zc
_event__: row_delete
_gtid__: 7d2ea78d-b631-11e7-8afb-00163e0eef52:537
_host__: *****.mysql.rds.aliyuncs.com
_id__: 114
_table__: specialalarm
alarmtype: NO_ALARM
count: 55
id: 4829235
ip: 10.10. **.**
time: 2017-11-01 12:31:41
```

○ UPDATE statement

```
__source__: 10.30. **.**
__tag__:__hostname__: iZbp145dd9fccu****
__topic__:
_db__: zc
_event__: row_update
_gtid__: 7d2ea78d-b631-11e7-8afb-00163e0eef52:538
_host__: *****.mysql.rds.aliyuncs.com
_id__: 115
_old_alarmtype: NO_ALARM
_old_count: 55
_old_id: 4829234
_old_ip: 10.10.22.133
_old_time: 2017-10-31 12:04:54
_table__: specialalarm
alarmtype: NO_ALARM
count: 55
id: 4829234
ip: 10.11. **.***",
time: 2017-10-31 12:04:54
```

- DDL statement

```

__source__: 10.30. **. **
__tag__: __hostname__: iZbp145dd9fccu****
__topic__:
__db__: zc
__event__: row_update
__gtid__: 7d2ea78d-b631-11e7-8afb-00163e0eef52:539
__host__: *****.mysql.rds.aliyuncs.com
ErrorCode: 0
ExecutionTime: 0
Query: ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC)
StatusVars:

```

Precautions

We recommend that you increase resource limits on Logtail to accommodate traffic surges and avoid risks to your data. If the limits are exceeded, Logtail may be forced to restart.

You can modify the resource limits in the `/usr/local/ilogtail/ilogtail_config.json` file. After modification, run the `sudo /etc/init.d/ilogtailed stop;sudo /etc/init.d/ilogtailed start` command to restart Logtail.

The following example shows how to set the CPU limit to two and memory limit to 2,048 MB.

```

{
  ...
  "cpu_usage_limit":2,
  "mem_usage_limit":2048,
  ...
}

```

30.3.1.5.2. Collect query results from a MySQL database

This topic describes how to use SQL statements to collect query results from a MySQL database.

Context

If you want to collect data from a MySQL database, you can install Logtail on a server and connect the server with the database. Then, you can create a Logtail configuration with a custom SQL statement in the Log Service console and deliver the configuration to Logtail. Logtail can use the custom SQL statement to collect data from the database at regular intervals.

 **Note** This feature applies only to Logtail 0.16.0 and later versions that run on Linux. For more information, see [Install Logtail in Linux](#).

Benefits

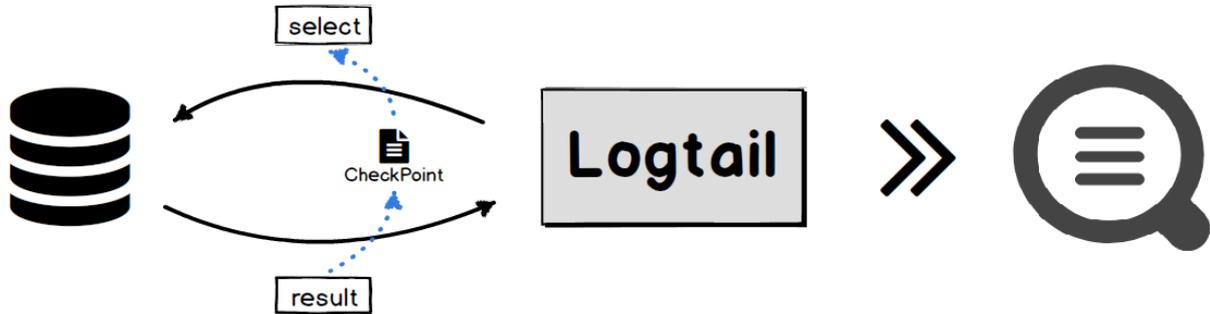
You can collect query results from databases that provide MySQL interfaces, including ApsaraDB for RDS databases. During the collection process, you can perform the following operations:

- Paginate query results.
- Set time zones.

- Set timeout periods.
- Store checkpoints.
- Transmit data over the SSL protocol.
- Set the maximum number of records that are collected for each query.

Implementation

Logtail periodically runs a SELECT statement that you specify and uploads the returned results to Log Service.



When Logtail obtains a result, it locally saves the value of the CheckPoint field specified in the result. The next time Logtail executes the SELECT statement, Logtail adds the last saved value of the CheckPoint field to the SELECT statement. Then, Logtail collects only the incremental data.

Scenarios

- You want to synchronize incremental data based on specific marks such as an auto-increment ID or a point in time.
- You want to customize synchronization based on filtering conditions.

Parameters

The following table describes Logtail parameters. The type of the data source is `service_mysql`.

Parameter	Type	Required	Description
Address	String	No	The address of the MySQL database. Default value: 127.0.0.1:3306.
User	String	No	The username of the MySQL database. Default value: root.
Password	String	No	The password of the MySQL database. This parameter is not specified by default.
DialTimeOutMs	Integer	No	The timeout period for the database connection. Unit: milliseconds. Default value: 5000.
ReadTimeOutMs	Integer	No	The timeout period for database reading. Unit: milliseconds. Default value: 5000.
StateMent	String	Yes	The SQL statement.

Parameter	Type	Required	Description
Limit	Boolean	No	Specifies whether to use limit-based pagination. Default value: false.
PageSize	Integer	No	The number of records to return on each page. This parameter is required when the Limit parameter is set to true.
MaxSyncSize	Integer	No	The maximum number of records to be synchronized at a time. If the value is 0, no limit is imposed. Default value: 0.
CheckPoint	Boolean	No	Specifies whether to use a checkpoint. Default value: false.
CheckPointColumn	String	No	The name of the checkpoint column. This parameter is required when the CheckPoint parameter is set to true.
CheckPointColumnType	String	No	The type of the checkpoint column. Valid values: <code>int</code> and <code>time</code> .
CheckPointStart	String	No	The initial value of the checkpoint.
CheckPointSavePerPage	Boolean	No	If this parameter is set to true, a checkpoint is saved after each pagination. If this parameter is set to false, a checkpoint is saved after each synchronization.
IntervalMs	Integer	Yes	The synchronization interval. Unit: milliseconds.

Limits

- We recommend that you use `limit`-based pagination. When you use `limit`-based pagination, the `LIMIT` clause is added to `StateMent` in the SQL query.
- If `CheckPoint` is set to true, the data selected through `StateMent` must contain the checkpoint column. In addition, the `WHERE` clause in `StateMent` must contain the checkpoint field. The value of the checkpoint field is expressed with a question mark (`?`).
For example, if the checkpoint is "id", `StateMent` is `SELECT * from ... where id > ?`.
- If `CheckPoint` is set to true, `CheckPointColumn`, `CheckPointColumnType`, and `CheckPointStart` must be specified.

- The value of `CheckpointColumnType` can only be set to `int` or `time`. If the value is set to `int`, internal storage uses the `int64` type. If the value is set to `time`, `MySQL DATE`, `DATETIME`, and `TIME` are supported.

Procedure

The following procedure describes how to perform an incremental synchronization from a MySQL database to Log Service. In this procedure, the `logtail.VersionOs` field is synchronized every 10 seconds. The value of the count parameter in this field is greater than 0. The value of the initial checkpoint is 2017-09-25 11:00:00. The pagination method is used and 100 records are returned on each page. The checkpoint is saved after each pagination. The steps are as follows:

1. [Log on to the Log Service console](#).
2. Select a data source. Click **Import Data**. On the **Import Data** page, select **MYSQL Query Result-Plug-in**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

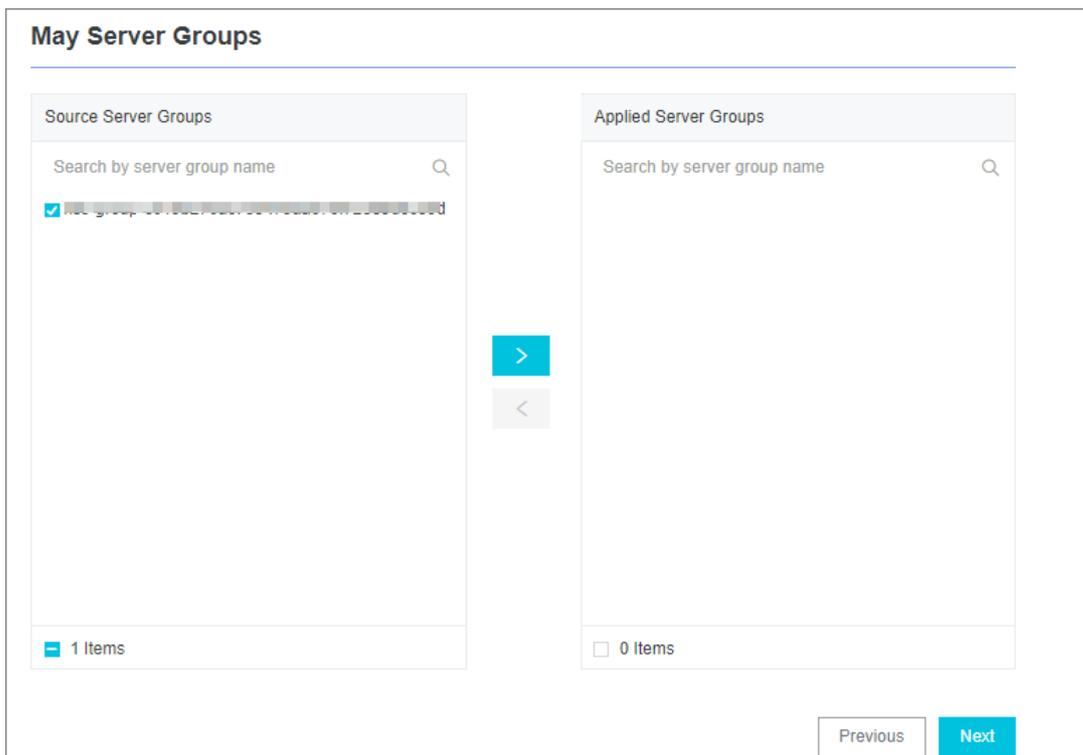
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.

5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Configure the data source.
 - In the **Plug-in Config** field, a configuration template is provided. Modify the parameters.
 - The **inputs** section is required. It specifies the collection configuration. The **processors** section is optional. It specifies the processing configuration. You must specify a collection statement for the collection configuration based on the data source. You can specify one or more processing methods for the processing configuration. For more information, see [Configure data processing methods](#).

Note If you have high security requirements, we recommend that you set the SQL access username and password to xxx. After you synchronize the Logtail configuration to the server, you can modify the username and password in the `/usr/local/ilogtail/user_log_config.json` file.

A configuration example is as follows:

```
{
  "inputs": [
    {
      "type": "service_mysql",
      "detail": {
        "Address": "*****:3306",
        "User": "logtail",
        "Password": "*****",
        "DataBase": "logtail",
        "Limit": true,
        "PageSize": 100,
        "StateMent": "SELECT * from logtail.VersionOs where time > ?",
        "CheckPoint": true,
        "CheckPointColumn": "time",
        "CheckPointStart": "2017-09-25 11:00:00",
        "CheckPointSavePerPage": true,
        "CheckPointColumnType": "time",
        "IntervalMs": 10000
      }
    }
  ]
}
```

7. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).
8. (Optional) Modify the configurations on the server. If you do not enter your actual URL, account, or password on the **Specify Data Source** page, you must modify the information after you deliver the Logtail configuration to the local server.

Note If you have entered the actual information, skip this step.

- i. Log on to the server where Logtail is installed, find the `service_mysql` keyword in the `/usr/local/ilogtail/user_log_config.json` file and modify the following fields: `Address`, `User`, and `Password`.
- ii. Run the following command to restart Logtail:

```
sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
```

Example

After you configure the processing method, you can view the processed data in the Log Service console. The following code shows examples of the schema and data collected by Logtail.

- Schema

```
CREATE TABLE `VersionOs` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT COMMENT 'id',  
  `time` datetime NOT NULL,  
  `version` varchar(10) NOT NULL DEFAULT "",  
  `os` varchar(10) NOT NULL,  
  `count` int(11) unsigned NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `timeindex` (`time`)  
)
```

- Sample output

```
"count": "4"  
"id": "721097"  
"os": "Windows"  
"time": "2017-08-25 13:00:00"  
"version": "1.3.0"
```

30.3.1.5.3. Collect syslogs

This topic describes how to use the syslog plug-in of Logtail to collect syslogs from a server.

Prerequisites

Logtail 0.16.13 or a later version is installed on the server.

Overview

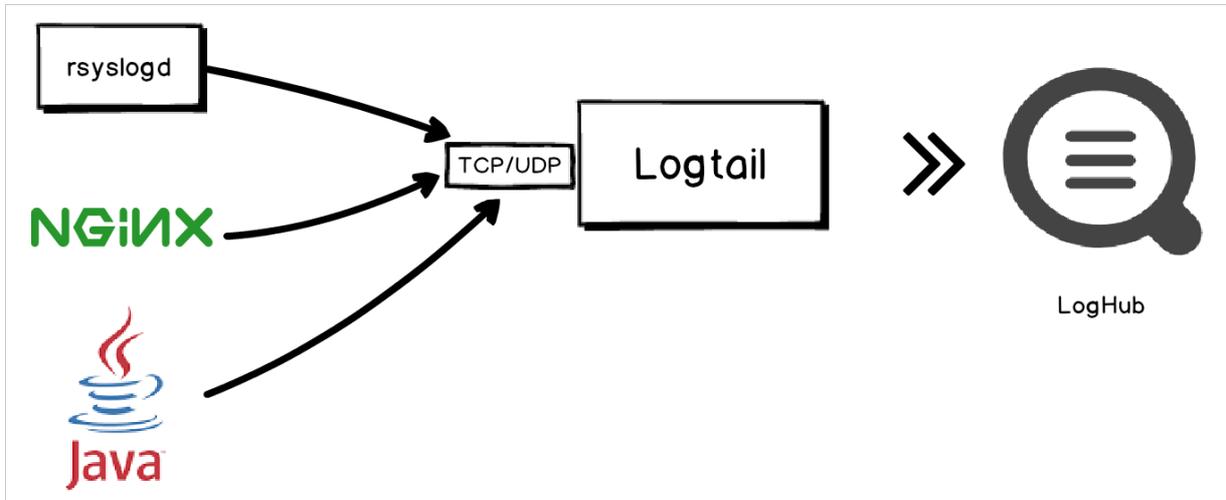
On a Linux server, local syslogs can be forwarded to the IP address and port of a specified server by using syslog agents such as rsyslog. After you create a Logtail configuration for the specified server, the syslog plug-in of Logtail receives the syslogs over TCP or UDP. In addition, the syslog plug-in parses the received syslogs and extract log fields such as facility, tag (program), severity, and content based on the specified syslog protocol. The syslog protocol can be RFC 3164 or RFC 5424.

Note

- Logtail installed on a Windows server does not support the syslog plug-in.
- You can configure multiple syslog plug-ins for Logtail. For example, you can use both TCP and UDP to listen on 127.0.0.1:9999.

Implementation

After the syslog plug-ins start to listen on a specified IP address and port, Logtail can act as a syslog server to collect syslogs from various data sources. These syslogs include system logs collected by rsyslog, access or error logs forwarded by NGINX, and logs forwarded by syslog clients in languages such as Java.



Logtail parameters

The following table describes Logtail parameters. The type of the input is `service_syslog`.

Parameter	Type	Required	Description
Address	String	No	<p>The protocol, address, and port on which the syslog plug-in listens. The syslog plug-in obtains logs based on the value of this parameter. Format: <code>[tcp/udp]://[ip]:[port]</code>. Default value: <code>tcp://127.0.0.1:9999</code>.</p> <div style="background-color: #e0f2f7; padding: 5px;"> <p>Note</p> <ul style="list-style-type: none"> The specified protocol, address, and port must be the same as those specified for the forwarding rule in the rsyslog configuration file. If the server on which Logtail is installed has multiple IP addresses, you can set the IP address to 0.0.0.0. It means that the syslog plug-in listens on all IP addresses of the server. </div>
ParseProtocol	String	No	<p>The protocol that is used to parse logs. This parameter is not specified by default, indicating that logs are not parsed. Valid values:</p> <ul style="list-style-type: none"> <code>rfc3164</code>: The RFC 3164 protocol is used to parse logs. <code>rfc5424</code>: The RFC 5424 protocol is used to parse logs. <code>auto</code>: The syslog plug-in selects a protocol based on the log content.
IgnoreParseFailure	Boolean	No	<p>Specifies whether to ignore a parsing failure. Default value: <code>true</code>. Valid values:</p> <ul style="list-style-type: none"> <code>true</code>: Logs that fail to be parsed are included in the returned content field. <code>false</code>: Logs that fail to be parsed are dropped.

Default fields

Field	Type	Description
hostname	String	The hostname. If a hostname is not provided in the log, the hostname of the current host is obtained.
program	String	The tag field in the protocol.
priority	String	The priority field in the protocol.
facility	String	The facility field in the protocol.
severity	String	The severity field in the protocol.
unixtimestamp	String	The timestamp of the log.
content	String	The log content. If the log fails to be parsed, this field contains the complete content of the log.
ip	String	The IP address of the current host.

Configure the plug-in of Logtail to collect syslogs

1. Add a forwarding rule for rsyslog. Modify the `/etc/rsyslog.conf` rsyslog configuration file on the server from which syslogs are collected. Add a forwarding rule at the end of the configuration file. Then, rsyslog forwards syslogs to the specified IP address and port.
 - If you want to collect syslogs of the server by using Logtail on this server, set the forwarding address to 127.0.0.1 and the port to an idle port.
 - If you want to collect syslogs of the server by using Logtail on a second server (Server B), set the forwarding address to the public IP address of the second server and port to an idle port.

For example, the following forwarding rule indicates that logs are forwarded to 127.0.0.1:9000 over TCP.

```
*.* @127.0.0.1:9000
```

2. Run the following command to restart rsyslog and validate the log forwarding rule:

```
sudo service rsyslog restart
```

3. [Log on to the Log Service console.](#)
4. Select the data source **Custom Data Plug-in**. You can use one of the following three methods to select a data source:
 - On the homepage of the Log Service console, select a data source in the **Import Data** section.
 - In the **Projects** section, click a project name. On the **Overview** page, click **Import Data**, and then select a data source.
 - On the **Logstores** tab in the left-side navigation pane, find a Logstore and click the closing angle bracket (>) in front of the Logstore name. Click the plus sign (+) next to **Data Import**, and then select a data source.

5. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

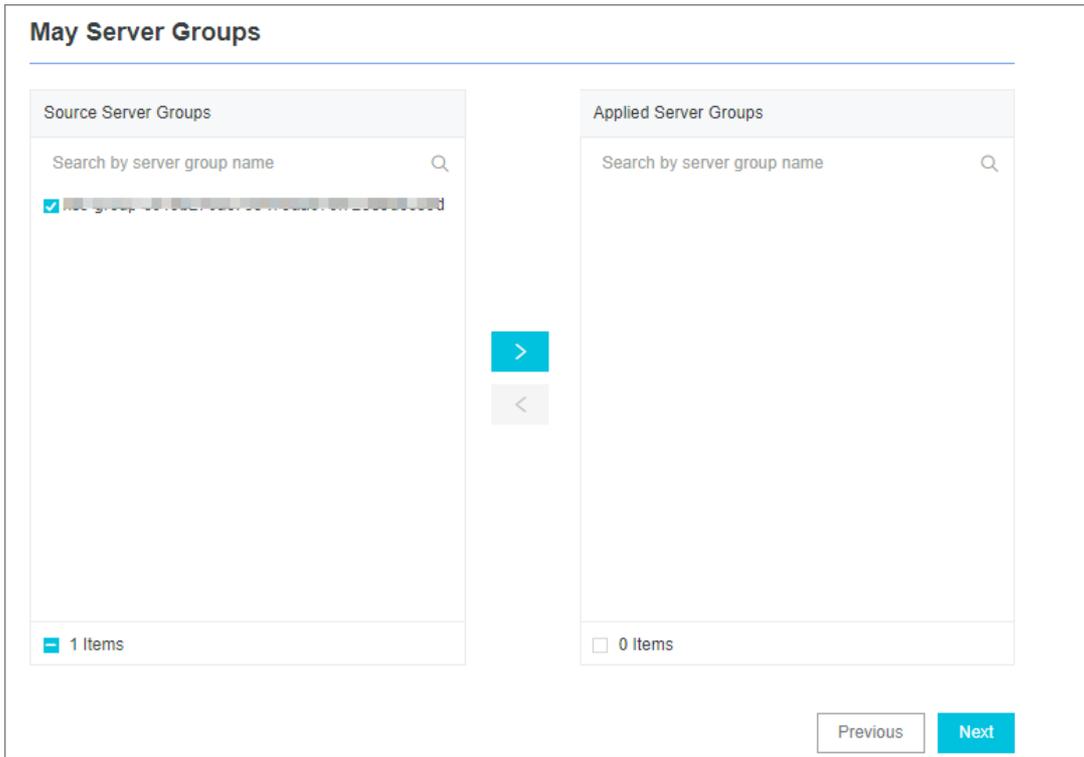
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the **Logstores** tab, the system skips this step.

6. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.

7. Configure the server group, and then click Next. Select a server group and move the group from Source Server Groups to Applied Server Groups.



8. Configure the data source. Set **Config Name** and **Plug-in Config**.

The `inputs` section is required. It specifies the collection configuration. The `processors` section is optional. It specifies the processing configuration. You must specify a collection statement for the collection configuration based on the data source. You can specify one or more processing methods for the processing configuration. For more information, see [Configure data processing methods](#).

The following sample code shows how to use UDP and TCP to listen on 127.0.0.1:9000:

```
{
  "inputs": [
    {
      "type": "service_syslog",
      "detail": {
        "Address": "tcp://127.0.0.1:9000",
        "ParseProtocol": "rfc3164"
      }
    },
    {
      "type": "service_syslog",
      "detail": {
        "Address": "udp://127.0.0.1:9001",
        "ParseProtocol": "rfc3164"
      }
    }
  ]
}
```

9. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Configure the plug-in of Logtail to collect NGINX logs

NGINX access logs can be forwarded to specified addresses and ports over the syslog protocol. To deliver NGINX access logs as syslogs from a server to Log Service, you can create a Logtail configuration and apply it to the server group to which the server belongs.

1. Add a forwarding rule to the `nginx.conf` configuration file on the NGINX server. For example, add the following content to the configuration file.

```
http {
    ...

    # Add this line.
    access_log syslog:server=127.0.0.1:9000,facility=local7,tag=nginx,severity=info combined;

    ...
}
```

2. Run the following command to restart the NGINX service and validate the configuration.

```
sudo service nginx restart
```

3. Create a Logtail configuration and apply it to the server group to which the server belongs. For more information, see [Configure the plug-in of Logtail to collect syslogs](#).
4. Check whether the Logtail configuration takes effect. Run the `curl http://127.0.0.1/test.html` command in shell to generate an access log. If the Logtail configuration takes effect, you can view the log information on the query page of the Log Service console.

30.3.1.5.4. Configure data processing methods

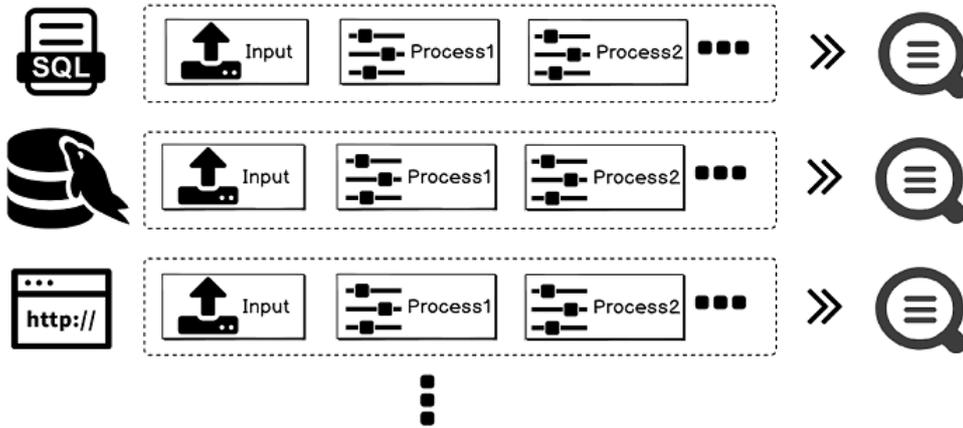
This topic describes how to configure processing methods for the data that you use a Logtail plug-in to collect from data sources. In Log Service, you can configure a plug-in to specify one or more processing methods for a data source when you create a Logtail configuration. After you deliver the Logtail configuration to Logtail installed on the data source, Logtail processes collected data based on the order in which these processing methods are configured.

 **Note** These processing methods are available only for plugin-driven input sources and container standard output.

Implementation

The following figure shows how to process collected data.

Implementation



Plug-in elements

When you configure the data processing methods, you must set the key parameter to `processors` and the value parameter to an array of JSON objects. Each object contains the details of a processing method.

Each object contains the `type` and `detail` fields. The `type` field specifies the type of a processing method and the `detail` field contains configuration details.

```

"processors" : [
  {
    "type": "processor_anchor"
    "detail" : {
      ...
    }
  },
  {
    "type": "processor_regex",
    "detail" : {
      ...
    }
  }
]
    
```

Processing methods

Available processing methods include:

- Regular expression-based extraction
- Calibration extraction
- Single-character delimiter
- Multi-character delimiter
- GeolP conversion
- Regular expression-based filter
- Filed insertion
- Field deletion
- Log time extraction (Go)
- Field expansion (JSON)

- [Field combination \(JSON\)](#)
- [Field renaming](#)
- [Log time extraction \(Strptime\)](#)

You can also create a custom method that includes several preceding methods. For more information, see [Custom methods](#).

Regular expression-based extraction

This method extracts the fields that match a specified regular expression.

The type of the plug-in is `processor_regex`.

Parameters

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the field to extract by using the regular expression.
Regex	String	Yes	The regular expression. Enclose the value of fields to extract with parentheses <code>()</code> .
Keys	String array	Yes	The array of fields to extract, for example, ["key1" , "key2" ...].
NoKeyError	Boolean	No	Specifies whether to report errors when no field matches the regular expression. Default value: false.
NoMatchError	Boolean	No	Specifies whether to report errors when the specified regular expression does not match logs.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: false. The value false specifies that the SourceKey parameter is not returned.
FullMatch	Boolean	Yes	Specifies whether to extract fields that exactly match the <code>Regex</code> parameter. Default value: true. The value false specifies that fields that partially match <code>Regex</code> parameter are extracted.

The following example shows how to extract fields from an access log.

- **Input data**

```
"content" : "10.200. **. ** - - [10/Aug/2017:14:57:51 +0800] \"POST /PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%
3A30%20GMT&Topic=raw&Signature=<yourSignature> HTTP/1.1\" 0.024 18204 200 37 \"-\" \"aliyun-sdk-java"
```

- **Plug-in configurations**

```
{
  "type": "processor_regex",
  "detail": {"SourceKey": "content",
    "regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+ \\] \\(\\w+ ([^\\\"]*)\\) ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \\("[^\\\"]*" \\ "[^\\\"]*"\\) (\\d+)",
    "key": ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "browser"],
    "NoKeyError": true,
    "NoMatchError": true,
    "KeepSource": false
  }
}
```

• Results

```
"ip" : "10.200. **. **",
"time" : "10/Aug/2017:14:57:51"
"method" : "POST",
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

Calibration extraction

This method uses the `start` and `stop` keywords to extract the required fields. You can extract fields from the JSON string between the start and stop keywords. You can also expand the JSON string into other forms.

The type of the plug-in is `processor_anchor`.

Parameters

Parameter	Type	Required	Parameters
SourceKey	String	Yes	The name of the field to extract.
Anchors	Array	Yes	The array that includes field-value pairs. For more information, see the following table.
NoAnchorError	Boolean	No	Specifies whether to report errors when no specified keyword is found. Default value: false.
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: false.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: false. A value of false specifies that the SourceKey parameter is not returned.

Anchor fields

Field	Type	Required	Description
Start	String	Yes	The start keyword. If you do not specify the parameter, Logtail matches the first character of a string.
Stop	String	Yes	The stop keyword. If you do not specify the parameter, Logtail matches the last character of a string.
FieldName	String	Yes	The name of the field to extract.
FieldType	String	Yes	The type of the field. Valid values: <code>string</code> and <code>json</code> .
ExpondJson	Boolean	No	Specifies whether to expand the JSON string. Default value: <code>false</code> . This parameter is available only if you specify <code>json</code> for the <code>FieldType</code> parameter.
ExpondConnector	String	No	The connector that combines separate field names into a string. Default value: <code>_</code> .
MaxExpondDepth	Integer	No	The maximum depth of JSON expansion. Default value: <code>0</code> , indicating no limit.

The following example shows how to use this method to process input data of multiple types.

- Input data

```
"content" : "time:2017.09.12 20:55:36\tjson:{\"key1\" : \"xx\", \"key2\": false, \"key3\":123.456, \"key4\" : { \"inner1\" : 1, \"inner2\" : false}}"
```

- Plug-in configurations

```
{
  "type": "processor_anchor"
  "detail": {"SourceKey": "content",
    "Anchors": [
      {
        "Start": "time",
        "Stop": "\\t",
        "FieldName": "time",
        "FieldType": "string",
        "ExpondJson": false
      },
      {
        "Start": "json:",
        "Stop": ""
        "FieldName": "val",
        "FieldType": "json",
        "ExpondJson": true,
      }
    ]
  }
}
```

- Results

```
"time" : "2017.09.12 20:55:36"
"val_key1" : "xx"
"val_key2" : "false"
"val_key3" : "123.456"
"value_key4_inner1" : "1"
"value_key4_inner2" : "false"
```

Single-character delimiter

This method uses `single-character delimiters` to split logs into several fields. You can enclose delimited fields with characters that you specify in the Quote parameter.

The type of the plug-in is `processor_split_char`.

Parameters

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the field to extract.
SplitSep	String	Yes	The single-character delimiter. You must specify a single character as a delimiter. You can specify an invisible character such as <code>\u0001</code> as a delimiter.
SplitKeys	String array	Yes	The names of the fields into which you split a log, for example, ["key1" , "key2" ...].

Parameter	Type	Required	Description
QuoteFlag	Boolean	No	Specifies whether to use the <code>Quote</code> parameter. Default value: false.
Quote	String	No	You must specify a single character. The parameter is available only if you specify true for the <code>QuoteFlag</code> parameter. You can specify invisible characters, such as <code>\u0001</code> .
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: false.
NoMatchError	Boolean	No	Specifies whether to report errors when the specified delimiter is not found. Default value: false.
KeepSource	Boolean	No	Specifies whether to return the <code>SourceKey</code> parameter. Default value: false. The value false specifies that the <code>SourceKey</code> parameter is not returned.

The following example shows how to use this method to process a log.

- Input data

```
"content" : "10. **. **. **|10/Aug/2017:14:57:51 +0800|POST|PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%
3A30%20GMT&Topic=raw&Signature=<yourSignature>|0.024|18204|200|37|-|
aliyun-sdk-java
```

- Configuration details

```
{
  "type" : "processor_split_char",
  "detail" : {"SourceKey" : "content",
    "SplitSep" : "|",
    "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "browser"]
  }
}
```

- Results

```
"ip" : "10. **. **. **",
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST",
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

Multi-character delimiter

Similar to the single-character delimiter method, this method uses multi-character delimiters to split a log into several fields. The Quote parameter is not available for the method.

The type of the plug-in is `processor_split_string` .

Parameters

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the field to extract.
SplitSep	String	Yes	The multi-character delimiter. You can specify a combination of several whitespace characters, for example, <code>\u0001\u0002</code> .
SplitKeys	String array	Yes	The names of the fields into which you split a log, for example, ["key1" , "key2" ...].
PreserveOthers	Boolean	No	Specifies whether to retain extra fields when the number of split fields exceeds the number fields defined in the <code>SplitKeys</code> parameter. Default value: false.
ExpandOthers	Boolean	No	Specifies whether to parse extra fields. Default value: false.
ExpandKeyPrefix	String	No	The prefix of the names of extra fields. For example, if you specify <code>expand_</code> for the parameter, extra fields are named <code>expand_1</code> and <code>expand_2</code> .
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: false.
NoMatchError	Boolean	No	Specifies whether to report errors when no multi-character delimiter is found. Default value: false.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: false. A value of false specifies that the SourceKey parameter is not returned.

The following example shows how to use this method to extract fields from a log.

- Input data

```
"content" : "10. **. **. **|#|10/Aug/2017:14:57:51 +0800|#|POST|#|PutData?
Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%
3A30%20GMT&Topic=raw&Signature=<yourSignature>|#|0.024|#|18204|#|200|#|37|#|-|#|
aliyun-sdk-java"
```

- Configuration details

```
{
  "type" : "processor_split_string",
  "detail" : {"SourceKey" : "content",
    "SplitSep" : "|#|",
    "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status"],
    "PreserveOthers" : true,
    "ExpandOthers" : true,
    "ExpandKeyPrefix" : "expand_"
  }
}
```

- Results

```
"ip" : "10. **. **. **",
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST",
"url" : "/PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%20
013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status": "red"
"expand_1" : "27"
"expand_2" : "-"
"expand_3" : "aliyun-sdk-java"
```

GeoIP conversion

This method converts IP addresses in a log into geolocation, including countries, provinces, cities, and geographic coordinates.

 **Note** By default, the Logtail installation package does not include GeoIP databases. You must download these databases to your localhost and configure the required parameters. We recommend that you download a version of the GeoIP database that includes the `city` database.

The type of the plug-in is `processor_geoip` .

Parameters

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the field to which you want to apply IP conversion.

Parameter	Type	Required	Description
DBPath	String	Yes	The absolute path of the GeoIP database. The database format is MMDB. For example, /user/data/GeoLite2-City_20180102/GeoLite2-City.mmdb.
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: false.
NoMatchError	Boolean	No	Specifies whether to report errors when the specified IP address is invalid or does not match any IP addresses stored in the library. Default value: false.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: true.
Language	String	No	The language. Default value: zh-CN. Make sure that converted results can be translated into the language.

The following example shows how to use this method to convert an IP address into geolocation.

- Input data

```
"source_ip" : "***.**.**.**"
```

Download a GeoIP database and install the database on the host where Logtail resides. We recommend that you download **MaxMind GeoLite2** that includes the city database.

 **Note** Make sure that the database format is MMDB.

- Configuration details

```
{
  "type": "processor_geoip",
  "detail": {
    "SourceKey": "ip",
    "NoKeyError": true,
    "NoMatchError": true,
    "KeepSource": true,
    "DBPath" : "/user/local/data/GeoLite2-City_20180102/GeoLite2-City.mmdb"
  }
}
```

- Results

```
"source_ip_city_" : "***.**.**.**",
"source_ip_province_" : "Zhejiang"
"source_ip_city_" : "Hangzhou"
"source_ip_province_code_" : "ZJ"
"source_ip_country_code_" : "CN"
"source_ip_longitude_" : "120.*****"
"source_ip_latitude_" : "30.*****"
```

Regular expression-based filter

This method uses regular expressions to filter logs. You can specify conditions in the `Include` and `Exclude` parameters.

The type of the plug-in is `processor_filter_regex`.

Parameters

Parameter	Type	Required	Description
Include	The JSON object that includes key-value pairs.	No	Each key includes a log field. Each value includes a regular expression. If the value of one of the log fields matches a regular expression, the log is collected.
Exclude	The JSON object that includes key-value pairs.	No	Each key includes a log field. Each value includes a regular expression. If the value of one of the log fields matches a regular expression, the log is dropped.

Note If a regular expression that you specified in the `Include` parameter matches the value of a log field and all regular expressions that you specified in the `Exclude` parameter do not match the value of any one of the log fields, the log is collected. Otherwise, the log is dropped.

The following example shows how to use this method to process logs.

- Input data

- Log 1

```
"ip" : "10. **. **.***",
"method" : "POST",
...
"browser" : "aliyun-sdk-java"
```

- Log 2

```
"ip" : "10. **. **.***",
"method" : "POST",
...
"browser" : "chrome"
```

- Log 3

```
"ip" : "192.168. *.*"
"method" : "POST",
...
"browser" : "ali-sls-ilogtail"
```

- Configuration details

```
{
  "type": "processor_filter_regex",
  "detail": {
    "Include": {
      "ip": "10\\.\\.\\.*",
      "method": "POST",
    },
    "Exclude": {
      "browser": "aliyun.*"
    }
  }
}
```

• Results

Log	Matched	Reason
Log 1	No	The match failed because the value of the browser field matches a regular expression you specified in the Exclude parameter.
Log 2	Yes	N/A
Log 3	No	The match failed because the value of the ip field does not match a regular expression you specified in the Include parameter. The regular expression matches IP addresses that start with 10 .

Filed insertion

You can use this method to add multiple fields to a log.

The type of the plug-in is `processor_add_fields` .

 **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
Fields	Map	No	The JSON object that includes key-value pairs. You can specify multiple key-value pairs in the parameter.
IgnoreIfExist	Boolean	No	Specifies whether to retain key-value pairs with the same keys. Default value: false.

The following example shows how to use this method to add fields to a log.

• Input data

```
"aaa1": "value1"
```

• Configuration details

```
{
  "processors": [
    {
      "type": "processor_add_fields",
      "detail": {
        "fields": {
          "aaa2": "value2",
          "aaa3": "value3"
        }
      }
    }
  ]
}
```

- Results

```
"aaa1": "value1"
"aaa2": "value2"
"aaa3": "value3"
```

Field deletion

You can use this method to remove specific fields from logs.

The type of the plug-in is `processor_drop` .

 **Note** The method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
DropKeys	Array	No	The array that includes a set of strings. You can remove multiple fields from a log.

The following example shows how to remove the `aaa1` and `aaa2` fields from a log.

- Input data

```
"aaa1": "value1"
"aaa2": "value2"
"aaa3": "value3"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_drop",
      "detail": {
        "DropKeys": ["aaa1", "aaa2"]
      }
    }
  ]
}
```

• Results

```
"aaa3": "value3"
```

Log time extraction (Go)

You can use this method to extract time from a log field and convert the time format.

The type of the plug-in is `processor_gotime`.

 **Note** This plug-in is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the log field from which you want to extract time.
SourceFormat	String	Yes	The time that parsed in Go.
SourceLocation	Integer	Yes	The time zone.
DestKey	String	Yes	The destination field.
DestFormat	String	Yes	The destination time format in Go.
DestLocation	Integer	No	The destination time zone. An empty value specifies the time zone of localhost.
SetTime	Boolean	No	Specifies whether to overwrite the original time. Default value: true.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: true.
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: true.
AlarmIfFail	Boolean	No	Specified whether to report error when extraction failed. Default value: true.

Use `2006-01-02 15:04:05` (UTC +8) of the `s_key` field as the source time. Convert the time into `2006/01/02 15:04:05` (UTC +9) and save the new time in the `d_key` field. The following example shows how to use this method to process a log.

• Input data

```
"s_key": "2019-07-05 19:28:01"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_gotime",
      "detail": {
        "SourceKey": "s_key",
        "SourceFormat": "2006-01-02 15:04:05",
        "SourceLocation": 8,
        "DestKey": "d_key",
        "DestFormat": "2006/01/02 15:04:05",
        "DestLocation": 9,
        "active": true,
        "KeepSource": true,
        "NoKeyError": true,
        "AlarmIfFail": true
      }
    }
  ]
}
```

- Results

```
"s_key": "2019-07-05 19:28:01"
"d_key": "2019/07/05 20:28:01"
```

Field expansion (JSON)

You can use this method to expand a log field.

The type of the plug-in is `processor_json`.

 **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Name	Data type	Required	Parameters
SourceKey	String	Yes	The name of the field you want to expand.
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: true.
ExpandDepth	Integer	No	The depth to expand. Default value: 0, specifying the maximum depth to expand. A value of n specifies that the number of levels deep to expand is n.
ExpandConnector	String	No	The delimiter that you use to connect multiple levels. Default value: <code>_</code> . You can leave this parameter empty.

Name	Data type	Required	Parameters
Prefix	String	No	The prefix that you want to add to the name of each new field after expansion.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: true.
UseSourceKeyAsPrefix	Boolean	No	Specifies whether to retain the name of the original field as part of the name of each new field after expansion. Default value: false.

The following example shows how to the field expansion (JSON) method to expand the `s_key` field.

- Input data

```
"s_key":>{"k1\":{"k2\":{"k3\":{"k4\":{"k51\":"51\","\k52\":"52\"},\k41\":"41\"}}}}}
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_json",
      "detail": {
        "SourceKey": "s_key",
        "NoKeyError": true,
        "ExpandDepth": 0,
        "ExpandConnector": "-",
        "Prefix": "j",
        "KeepSource": false,
        "UseSourceKeyAsPrefix": true
      }
    }
  ]
}
```

- Results

```
"s_key":>{"k1\":{"k2\":{"k3\":{"k4\":{"k51\":"51\","\k52\":"52\"},\k41\":"41\"}}}}}"
"js_key-k1-k2-k3-k4-k51":"51"
"js_key-k1-k2-k3-k4-k52":"52"
"js_key-k1-k2-k3-k41":"41"
```

Field combination (JSON)

You can use this method to combine multiple log fields into one field.

The type of the plug-in is `processor_packjson`.

 **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Name	Type	Required	Description
SourceKeys	Array	Yes	The array that include the names of fields that you want to combine.
DestKey	String	No	The name of the destination field after combination.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: true.
AlarmIfIncomplete	Boolean	No	Specifies whether to report alerts when no match exists. Default value: true.

The following example shows how to use the method to combine the `a` and `b` fields into the `d_key` field

- Input data

```
"a": "1"
"b": "2"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_packjson",
      "detail": {
        "SourceKeys": ["a", "b"],
        "DestKey": "d_key",
        "KeepSource": false,
        "AlarmIfEmpty": true
      }
    }
  ]
}
```

- Results

```
"a": "1"
"b": "2"
"d_key": "{\"a\": \"1\", \"b\": \"2\"}"
```

Field renaming

You can use this method to rename multiple fields.

The type of the plug-in is `processor_rename` .

 **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Data type	Required	Description
-----------	-----------	----------	-------------

Parameter	Data type	Required	Description
NoKeyError	Boolean	No	Specifies whether to report errors when no match exists. Default value: true.
SourceKeys	Array	Yes	The array that includes the names of fields that you want to rename.
DestKeys	Array	Yes	The array that includes the new names of the fields.

The following example shows how to use this method to rename the `aaa1` and `aaa2` fields to `bbb1` and `bbb2` .

- Input data

```
"aaa1": "value1"
"aaa2": "value2"
"aaa3": "value3"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_rename",
      "detail": {
        "SourceKeys": ["aaa1", "aaa2"],
        "DestKeys": ["bbb1", "bbb2"],
        "NoKeyError": true,
      }
    }
  ]
}
```

- Results

```
"bbb1": "value1"
"bbb2": "value2"
"aaa3": "value3"
```

Log time extraction (Strptime)

You can use this method to extract time from a log field and parse the time by using the Linux `strptime()` function.

The type of the plug-in is `processor_strptime` .

 **Note** This method is available for Logtail 0.16.28 or later.

Parameters

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the field from which you want to extract time.
Format	String	Yes	The time format to parse the time.
AdjustUTCOffset	Boolean	No	Specifies whether to adjust the time zone. Default value: false.
UTCOffset	Integer	No	The offset in seconds. For example, a value of 28800 changes the time zone to UTC +8.
AlarmIfFail	Boolean	No	Specifies whether to report alerts when extraction failed. Default value: true.
KeepSource	Boolean	No	Specifies whether to return the SourceKey parameter. Default value: true.

Parse the value of the `log_time` in the `%Y/%m/%d %H:%M:%S` time format and use the time zone of your localhost. The following examples show how to use this method to process logs.

- Example 1: The time zone is UTC +8.

- Input data

```
"log_time": "2016/01/02 12:59:59"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_strptime",
      "detail": {
        "SourceKey": "log_time",
        "Format": "%Y/%m/%d %H:%M:%S"
      }
    }
  ]
}
```

- Results

```
"log_time": "2016/01/02 12:59:59"
Log.Time = 1451710799
```

- Example 2: The time zone is UTC +7.

- Input data

```
"log_time": "2016/01/02 12:59:59"
```

- Configuration details

```
{
  "processors": [
    {
      "type": "processor_strptime",
      "detail": {
        "SourceKey": "content",
        "Format": "%Y/%m/%d %H:%M:%S",
        "AdjustUTCOffset": true,
        "UTCOffset": 25200
      }
    }
  ]
}
```

- Results

```
"log_time": "2016/01/02 12:59:59"
Log.Time = 1451714399
```

Custom methods

You can use a combination of multiple processing methods to process logs. The following example shows how to use a single-character delimiter to split a log into several fields and then specify calibration points to extract the required contents from the `detail` field.

- Input data

```
"content": {
  "ACCESS|QAS|11. **. **. **|1508729889935|52460dbed4d540b88a973cf5452b1447|1238|appKey=ba,env=pub,requestTime
=1508729889913,latency=22ms,
request={appKey:ba,optional:{\domains\:\:\daily\,\version\:\:\v2\},rawQuery:{\query\:\:\The route to Locati
on A\,\domain\:\:\Navaigation\,\intent\:\:\navigate\,\slots\:\:\to_geo:level3=Location A\,\location\:\:\Lo
cation B\},
requestId:52460dbed4d540b88a973cf5452b1447},
response={answers:[],status:SUCCESS}"
```

- Configuration details

```
"processors" : [
  {
    "type" : "processor_split_char",
    "detail" : {"SourceKey" : "content",
      "SplitSep" : "|",
      "SplitKeys" : ["method", "type", "ip", "time", "req_id", "size", "detail"]}
  },
  {
    "type" : "processor_anchor",
    "detail" : "SourceKey" : "detail",
    "Anchors" : [
```

```
{
  "Start" : "appKey=",
  "Stop" : ",env=",
  "FieldName" : "appKey",
  "FieldType" : "string"
},
{
  "Start" : ",env=",
  "Stop" : ",requestTime=",
  "FieldName" : "env",
  "FieldType" : "string"
},
{
  "Start" : ",requestTime=",
  "Stop" : ",latency",
  "FieldName" : "requestTime",
  "FieldType" : "string"
},
{
  "Start" : ",latency=",
  "Stop" : ",request=",
  "FieldName" : "latency",
  "FieldType" : "string"
},
{
  "Start" : ",request=",
  "Stop" : ",response=",
  "FieldName" : "request",
  "FieldType" : "string"
},
{
  "Start" : ",response=",
  "Stop" : "",
  "FieldName" : "response",
  "FieldType" : "json"
}
]
}
}
]
```

- Results

```
"method" : "ACCESS"
"type" : "QAS"
"ip" : "****. *. *. **"
"time" : "1508729889935"
"req_id" : "52460dbed4d540b88a973cf5452b1447"
"size" : "1238"
"appKey" : "ba"
"env" : "pub"
"requestTime" : "1508729889913"
"latency" : "22ms"
"request" : "{appKey:nui-banma,optional:{\\domains\\:\\daily-faq\\,\\version\\:\\v2\\},rawQuery:{\\query\\:\\345\\216\\273\\344\\271\\220\\345\\261\\261\\347\\232\\204\\350\\267\\257\\347\\272\\277\\,\\domain\\:\\345\\257\\274\\350\\210\\252\\,\\intent\\:\\navigate\\,\\slots\\:\\to_geo:level3=\\344\\271\\220\\345\\261\\261\\,\\location\\:\\345\\214\\227\\344\\272\\254\\},requestId:52460dbed4d540b88a973cf5452b1447}"
"response_answers" : "[]"
"response_status" : "SUCCESS"
```

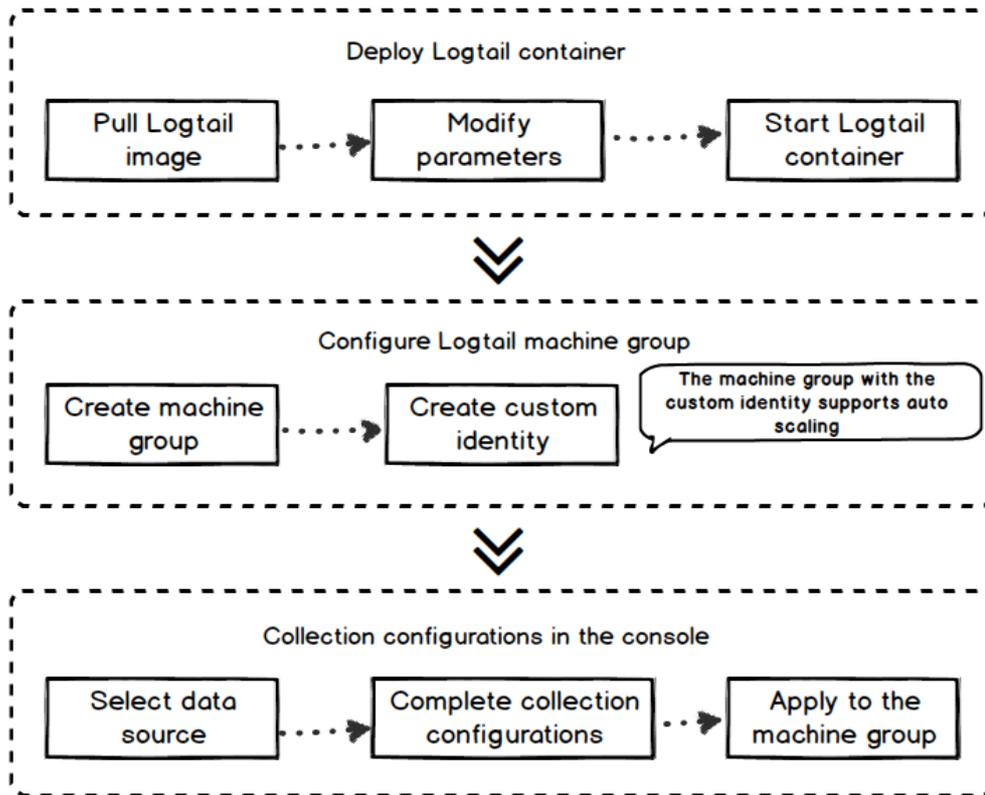
30.3.1.6. Collect container logs

30.3.1.6.1. Collect standard Docker logs

This topic describes how to use Logtail to collect standard Docker logs and upload these logs together with the container metadata to Log Service.

Procedure

Procedure



1. **Deploy a Logtail container.**
2. **Configure a Logtail server group.**

Create a server group with a custom ID in the Log Service console. The container cluster can automatically scale up or down based on data traffic.

3. **Create a Logtail configuration.**

Create a Logtail configuration in the Log Service console. The Logtail configuration process is completed in the Log Service console. No local configuration is needed.

Deploy a Logtail container

1. Run the following command to pull the Logtail image.

```
docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```

2. Start a Logtail container. Set the `your_region_name`, `your_aliyun_user_id`, and `your_machine_group_user_defined_id` parameters in the startup template.

```
docker run -d -v /:/logtail_host:ro -v /var/run:/var/run --env
ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/your_region_name/ilogtail_config.json
--env ALIYUN_LOGTAIL_USER_ID=your_aliyun_user_id --env
ALIYUN_LOGTAIL_USER_DEFINED_ID=your_machine_group_user_defined_id registry.cn-hangzhou.aliyuncs.com/l
og-service/logtail
```

Notice Before you set the parameters, you must complete one of the following configurations. Otherwise, the `container text file busy` error may occur when you delete another container.

- For CentOS 7.4 and later versions, set `fs.may_detach_mounts` to 1. For more information, see [Bug 1468249](#), [Bug 1441737](#), and [Issue 34538](#).
- Grant the `privileged` permission to Logtail by adding the `--privileged` flag to the startup parameters. For more information, see [Docker run reference](#).

Parameter	Description
<code>#{your_region_name}</code>	The region of the project. For more information, see View the information of a project .
<code>#{your_aliyun_user_id}</code>	The user ID. Set this parameter to the ID of your Alibaba Cloud account, which is a string. For information about how to view the ID, see Step 1 in Configure an account ID for a server .
<code>#{your_machine_group_user_defined_id}</code>	The custom ID of your server group. For information about how to set the custom ID, see Step 1 in Create a machine group based on a custom ID .

After you set the parameters, run the following command to start the Logtail container.

```
docker run -d -v /:/logtail_host:ro -v /var/run:/var/run
--env ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/cn_hangzhou/ilogtail_config.json --env
ALIYUN_LOGTAIL_USER_ID=1654218*****--env ALIYUN_LOGTAIL_USER_DEFINED_ID=log-docker-demo registry.cn-ha
ngzhou.aliyuncs.com/log-service/logtail
```

Notice

You can customize the startup parameters of the Logtail container if the following conditions are met:

- The following environment variables exist before you start the Logtail container: `ALIYUN_LOGTAIL_USER_DEFINED_ID`, `ALIYUN_LOGTAIL_USER_ID`, and `ALIYUN_LOGTAIL_CONFIG`.
- The `/var/run` directory is mounted on the `/var/run` directory of the Logtail container.
- To collect container standard output, container logs, or host files, you must mount the root directory on the `/logtail_host` directory of the Logtail container.
- If an error showing *The parameter is invalid : uuid=none* occurs in the `/usr/local/ilogtail/ilogtail.LOG` Logtail log file, create a file named `product_uuid` on the host. Add a valid UUID such as `169E98C9-ABC0-4A92-B1D2-AA6239C0D261` to the file, and mount the file on the `/sys/class/dmi/id/product_uuid` directory of the Logtail container.

Configure a Logtail server group

- Log on to the [Log Service console](#).
- Click a project name.
- In the left-side navigation pane, click the **Server Groups** icon to show the server group list.
- Click the icon next to **Server Groups**, and then select **Create Server Group**. You can also create a server group when you import data to Log Service.
- In the dialog box that appears, select **Custom ID** from the Identifier drop-down list. Enter the value of `ALIYUN_LOGTAIL_USER_DEFINED_ID` set in the previous step in the **Custom Identifier** field.

Click OK. One minute later, click the name of the server group in the **Server Groups** list. On the **Server Group Settings** page that appears, you can view the heartbeat status of the Logtail container. For more information, see [View the status of a server group](#).

Create a Logtail configuration

Create a Logtail configuration in the console.

- For more information about Docker logs, see [Collect container text logs](#).
- For more information about Docker standard output, see [Collect stdout and stderr logs from containers](#).
- [Host text logs](#).

The root directory of a host is mounted on the `/logtail_host` directory of the Logtail container by default. You must add the `/logtail_host` prefix to the log path. For example, if you want to collect data from the `/home/logs/app_log/` directory of the host, you must set the log path as `/logtail_host/home/logs/app_log/`.

What to do next

- View the status of the Logtail container.

You can run the `docker exec ${logtail_container_id} /etc/init.d/ilogtailed status` command to view the status of Logtail.

- View the version number, IP address, and startup time of Logtail.

You can run the `docker exec ${logtail_container_id} cat /usr/local/ilogtail/app_info.json` command to view Logtail information.

- View the operations logs of Logtail.

The operations logs of Logtail are stored in the `ilogtail.LOG` file in the `/usr/local/ilogtail/` directory. If the log file is rotated and compressed, it is stored as a file named `ilogtail.LOG.x.gz`.

For example:

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 tail -n 5 /usr/local/ilogtail/ilogtail.LOG
[2018-02-06 08:13:35.721864] [INFO] [8] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume:start
[2018-02-06 08:13:35.722135] [INFO] [8] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume:success
[2018-02-06 08:13:35.722149] [INFO] [8] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] start add existed check point events, size:0
[2018-02-06 08:13:35.722155] [INFO] [8] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point events, size:0 cache size:0 event size:0 success count:0
[2018-02-06 08:13:39.725417] [INFO] [8] [build/release64/sls/ilogtail/ConfigManager.cpp:3776] check container path update flag:0 size:1
```

Ignore the following standard output:

```

start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82
155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e6
9718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c2
2dbe/merged: must be superuser to unmount
...
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running

```

- Restart Logtail.

To restart Logtail, use the following sample code:

```

[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtailed stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtailed start
ilogtail is running

```

30.3.1.6.2. Collect Kubernetes logs

This topic describes how to install and use Logtail to collect logs from Kubernetes clusters.

Configuration procedure

Perform the following steps to collect logs from Kubernetes clusters:

1. Install the alibaba-log-controller Helm package.
2. Use the Log Service console to manage log collection configurations.

Step 1: Install Logtail

- Install Logtail in an Alibaba Cloud Container Service for Kubernetes cluster.

If Log Service components are not installed in your cluster, you must manually install the components.

- i. Connect to the Kubernetes cluster by using CloudShell.
- ii. Run the following command in CloudShell to obtain the ID of your Apsara Stack tenant account.

```
echo $ALIBABA_CLOUD_ACCOUNT_ID
```

- iii. After you set the `${your_k8s_cluster_id}`, `${your_ali_uid}`, and `${your_k8s_cluster_region_id}` parameters, run the following command:

```
wget https://acs-logging.oss-cn-hangzhou.aliyuncs.com/alibabacloud-k8s-log-installer.sh -O alibabacloud-k8s-log-installer.sh; chmod 744 ./alibabacloud-k8s-log-installer.sh; ./alibabacloud-k8s-log-installer.sh --cluster-id ${your_k8s_cluster_id} --ali-uid ${your_ali_uid} --region-id ${your_k8s_cluster_region_id}
```

- Install Logtail in a user-created Kubernetes cluster.

 Notice

- The version of the Kubernetes cluster must be 1.8 or later.
- Helm 2.6.4 or later must be installed.

- In the Log Service console, create a project whose name starts with `k8s-log-custom-`.
- Replace the parameters in the following command based on your business requirements:

```
wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/kubernetes/alibabacloud-log-k8s-custom-install.sh; chmod 744 ./alibabacloud-log-k8s-custom-install.sh; sh ./alibabacloud-log-k8s-custom-install.sh {your-project-suffix} {region-id} {aliuid} {access-key-id} {access-key-secret}
```

The following table lists the parameters in the preceding command.

Parameter	Description
{your-project-suffix}	The portion of the project name at the end of <code>k8s-log-custom-</code> . For example, if you create a project whose name is <code>k8s-log-custom-xxxx</code> , set this parameter to <code>xxxx</code> .
{regionid}	The ID of the region where the project resides. For more information, see View the information of a project .
{aliuid}	The user ID. Set this parameter to the ID of your Apsara Stack tenant account.  Note The ID of an Apsara Stack tenant account is a string. For more information about how to obtain the ID, see Configure an account ID for a server .
{access-key-id}	The AccessKey ID of your Apsara Stack tenant account.
{access-key-secret}	The AccessKey secret of your Apsara Stack tenant account.

After Logtail is installed in the Kubernetes cluster, Log Service automatically creates a machine group named `k8s-group-${your_k8s_cluster_id}` for the project.

 Note

- A Logstore named `config-operation-log` is automatically created in the project. Do not delete the Logstore.
- When you install Logtail in a user-created Kubernetes cluster, Logtail is granted `privileged` permissions by default. This prevents the `container text file busy` error when you delete a pod. For more information, visit [Bug 1468249](#), [Bug 1441737](#), and [Issue 34538](#).

The following example shows a successful installation:

```
[root@iZbp1dsxxxxqfbiaZ ~]# wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/kubernetes/alicloud-log-k8s-custom-install.sh; chmod 744 ./alicloud-log-k8s-custom-install.sh; sh ./alicloud-log-k8s-custom-install.sh xxxx cn-hangzhou 165xxxxxxxx050 LTAxxxxxxxxx Alxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
....
....
....
NAME: alibaba-log-controller
LAST DEPLOYED: Fri May 18 16:52:38 2018
NAMESPACE: default
STATUS: DEPLOYED
RESOURCES:
==> v1beta1/ClusterRoleBinding
NAME          AGE
alibaba-log-controller 0s
==> v1beta1/DaemonSet
NAME    DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE
logtail-ds 2    2    0  2    0    <none>    0s
==> v1beta1/Deployment
NAME          DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
alibaba-log-controller 1    1    1    0    0s
==> v1/Pod(related)
NAME          READY STATUS    RESTARTS AGE
logtail-ds-7xf2d    0/1 ContainerCreating 0    0s
logtail-ds-9j4bx    0/1 ContainerCreating 0    0s
alibaba-log-controller-796f8496b6-6jxb2 0/1 ContainerCreating 0    0s
==> v1/ServiceAccount
NAME          SECRETS AGE
alibaba-log-controller 1    0s
==> v1beta1/CustomResourceDefinition
NAME          AGE
aliyunlogconfigs.log.alibabacloud.com 0s
==> v1beta1/ClusterRole
alibaba-log-controller 0s
[INFO] your k8s is using project : k8s-log-custom-xxx, region : cn-hangzhou, aliuid : *****, accessKeyld : LTA*
*****
[SUCCESS] install helm package : alibaba-log-controller success.
```

To check the status of each Log Service component in the Kubernetes cluster, run the `helm status alibaba-log-controller` command. If all pods are in the Running state, Logtail is installed.

Log on to the Log Service console to find the project. If you have multiple projects, search for the project by using the `k8s-log` keyword.

Step 2: Configure log collection

Create Logtail configurations for log collection in the console as required.

- For information about how to collect Kubernetes text logs, see [Collect container text logs](#).
- For information about how to collect Kubernetes stdout logs, see [Collect stdout and stderr logs from](#)

containers.

- **Host text logs.**

By default, the root directory of a host is mounted to the `/logtail_host` directory of the Logtail container. You must add the `/logtail_host` prefix to the log path. For example, if you want to collect data from the `/home/logs/app_log/` directory of the host, you must set the log path to `/logtail_host/home/logs/app_log/`.

Other common commands

- **Store logs of multiple clusters in one project**

You can collect logs from multiple Kubernetes clusters. If you want to store these logs in the same project, you can specify the same cluster ID for the `#{your_k8s_cluster_id}` parameter when you install Log Service components on multiple Kubernetes clusters.

For example, if you have three Kubernetes clusters whose IDs are `abc001`, `abc002`, and `abc003`, specify `abc001` for the `#{your_k8s_cluster_id}` parameter when you install Log Service components for each Kubernetes cluster.

 **Notice** This feature is unavailable for Kubernetes clusters that reside in different regions.

- **Logtail container logs**

Logtail log files named `ilogtail.LOG` and `logtail_plugin.LOG` are stored in the `/usr/local/ilogtail/` directory of a Logtail container. Ignore the following standard output:

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82
155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e6
9718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c2
2dbe/merged: must be superuser to unmount
.....
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- **View the status of each Log Service component in a Kubernetes cluster**

```
helm status alibaba-log-controller
```

- **Troubleshoot alibaba-log-controller startup failures**

Make sure that the following conditions are met:

- Log Service components are installed on the master node of the Kubernetes cluster.
- The Kubernetes cluster ID that you specified is valid when you install Log Service components.

If Log Service components fail to be installed because the preceding conditions are not met, run the `helm delete --purge alibaba-log-controller` command to delete the installation package and install Log Service components again.

- **View the status of Logtail DaemonSet in a Kubernetes cluster**

Run the `kubectl get ds -n kube-system` command.

 **Note** The default namespace of Logtail is `kube-system`.

- View the version number, IP address, and startup time of Logtail.

Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl get po -n kube-system | grep logtail
NAME          READY   STATUS    RESTARTS   AGE
logtail-ds-gb92k 1/1     Running   0          2h
logtail-ds-wm7lw 1/1     Running   0          4d
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json
{
  "UUID" : "",
  "hostname" : "logtail-ds-gb92k",
  "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.2_1517810940",
  "ip" : "172.20.4.2",
  "logtail_version" : "0.16.2",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
  "update_time" : "2018-02-05 06:09:01"
}
```

- View the operational logs of Logtail

Logtail operational logs are stored in the `ilogtail.LOG` file in the `/usr/local/ilogtail/` directory. If the log file is rotated and compressed, it is stored as a file named `ilogtail.LOG.x.gz`.

Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system tail /usr/local/ilogtail/ilogtail.LOG
[2018-02-05 06:09:02.168693] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume: start
[2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume: success
[2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] start add existed check point events, size:0
[2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point events, size:0 cache size:0 event size:0 success count:0
```

- Restart Logtail in a pod

Example:

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild start
ilogtail is running
```

30.3.1.6.3. Collect container text logs

Logtail collects text logs generated in containers and uploads these logs together with the container-related metadata information to Log Service.

Features

Compared with basic log file collection, Docker file collection has the following characteristics:

- Allows you to configure the log path of a container, without the need to consider the mapping between the path and the host.
- Allows you to use labels to specify the containers whose logs are to be collected.
- Allows you to use labels to exclude specific containers.
- Allows you to use environment variables to specify the containers whose logs are to be collected.
- Allows you to use environment variables to exclude specific containers.
- Supports multiline logs such as Java Stack logs.
- Supports automatic labeling for container data.
- Supports automatic labeling for Kubernetes containers.

Limits

- Collection stop policy: When a container is stopped and Logtail detects the `die` event on the container, Logtail stops collecting logs of the container (with a latency of no more than 3 seconds). In this case, if a collection latency occurs, some logs generated before the stop action may be lost.
- Docker storage driver: Only overlay and overlay2 are supported. For other storage drivers, you must mount the log directory to the local host.
- Logtail running mode: Logtail must run in a container and must be deployed based on Logtail deployment solutions.
- Label: refers to the label information in `docker inspect`. It is not synonymous with labels in Kubernetes.
- Environment: refers to the environment information configured during container startup.

Procedure

1. Deploy and configure the Logtail container.
2. Configure log collection in Log Service.

Logtail deployment and configuration

- Kubernetes

For more information about Kubernetes log collection, see [Logtail deployment solution for collecting Kubernetes logs](#).

- Management methods for other containers

For more information about management methods for other containers, such as Swarm and Mesos, see [Common deployment solution for collecting Docker logs](#).

Collection configuration

1. [Log on to the Log Service console](#).
2. Click the **Import Data** button. On the **Import Data** page that appears, select **Docker File**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).

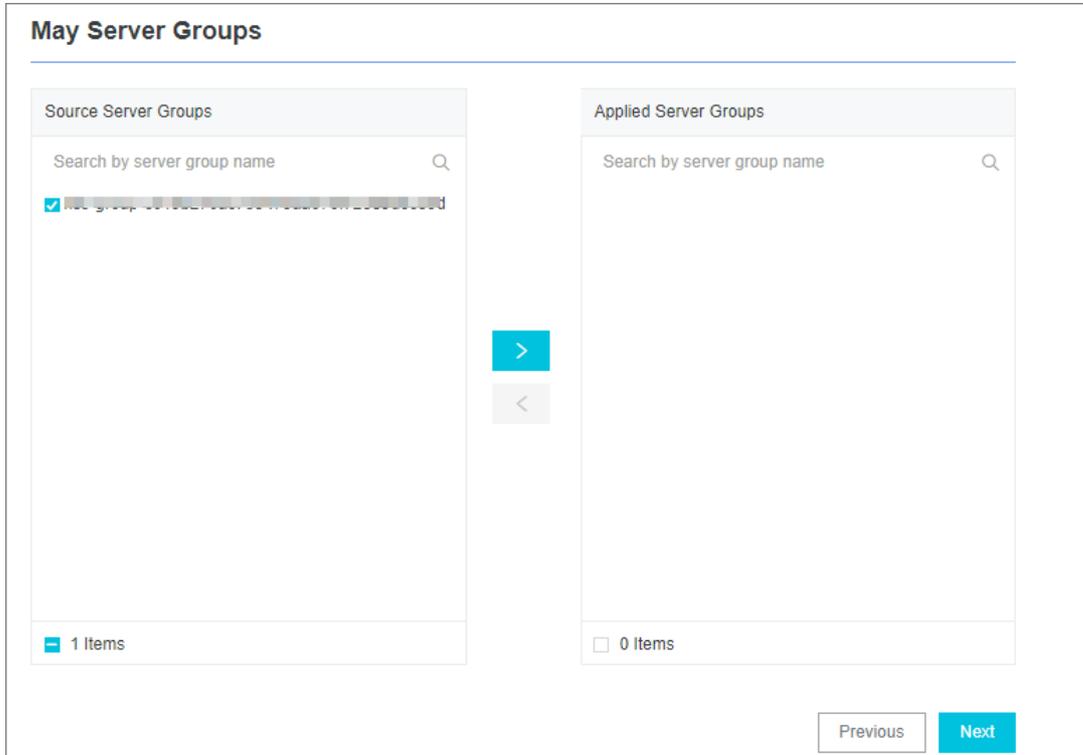
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a server group. Before you create a server group, ensure that Logtail is installed.

Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.

5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Configure Logtail. The following table lists data source-specific parameters. For more information about common parameters, see [Configure text log collection](#).

Parameter	Description
Docker File	This parameter is used to check whether the collected target file is a Docker file.
Label Whitelist	<p>LabelKey is required. If LabelValue is not empty, only containers whose labels contain LabelKey = LabelValue are collected. If LabelValue is empty, all the containers whose labels contain LabelKey are collected.</p> <div style="background-color: #e0f2f7; padding: 10px; border: 1px solid #ccc;"> <p>? Note</p> <ul style="list-style-type: none"> ◦ Key-value pairs are disjunctive with each other. If the label of a container contains one of the key-value pairs you specify, logs of the container are collected. ◦ Labels refer to Docker labels. </div>

Parameter	Description
Label Blacklist	<p>LabelKey is required. If LabelValue is not empty, only containers whose labels contain LabelKey = LabelValue are excluded. If LabelValue is empty, all containers whose labels contain LabelKey are excluded.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the label of a container contains one of the key-value pairs you specify, the container is excluded. Labels described in this topic refer to the label information in docker inspect.
Environment Variable Whitelist	<p>EnvKey is required. If EnvValue is not empty, only containers whose environment variables contain EnvKey = EnvValue are collected. If EnvValue is empty, all containers whose environment variables contain EnvKey are collected.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the environment variable of a container contains one of the key-value pairs you specify, logs of the container are collected. The environment variable refers to the environment information configured in container startup.
Environment Variable Blacklist	<p>EnvKey is required. If EnvValue is not empty, only containers whose environment variables contain EnvKey = EnvValue are excluded. If EnvValue is empty, all containers whose environment variables contain EnvKey are excluded.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the environment variable of a container contains one of the key-value pairs you specify, the container is excluded. The environment variable refers to the environment information configured in container startup.

Note

- Labels in whitelist and blacklist are different from those defined in Kubernetes. Labels in this topic refer to the label information in docker inspect.
- A namespace and a container name in Kubernetes can be mapped to Docker labels. LabelKey corresponding to a namespace is `io.kubernetes.pod.namespace`. LabelKey corresponding to a container name is `io.kubernetes.container.name`. For example, the namespace of the pod you created is `backend-prod` and the container name is `worker-server`. In this case, you can configure a whitelist label: `io.kubernetes.pod.namespace : backend-prod` or `io.kubernetes.container.name : worker-server`, so that only logs of the container are collected.
- In Kubernetes, we recommend that you only use the `io.kubernetes.pod.namespace` and `io.kubernetes.container.name` labels. In other cases, use an environment variable whitelist or blacklist.

7. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Default fields

Each uploaded log of a common Docker container contains the following fields.

Field	Description:
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	The IP address of the container.

If a Kubernetes cluster is used, each uploaded log contains the following fields.

Field	Description
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace to which the pod belongs.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_ip_</code>	The IP address of the pod.

30.3.1.6.4. Collect stdout and stderr logs from containers

This topic describes how to use Logtail to collect standard output (stdout) and standard error (stderr) logs from containers. After you collect stdout and stderr logs, you can upload the logs together with the container-related metadata to Log Service.

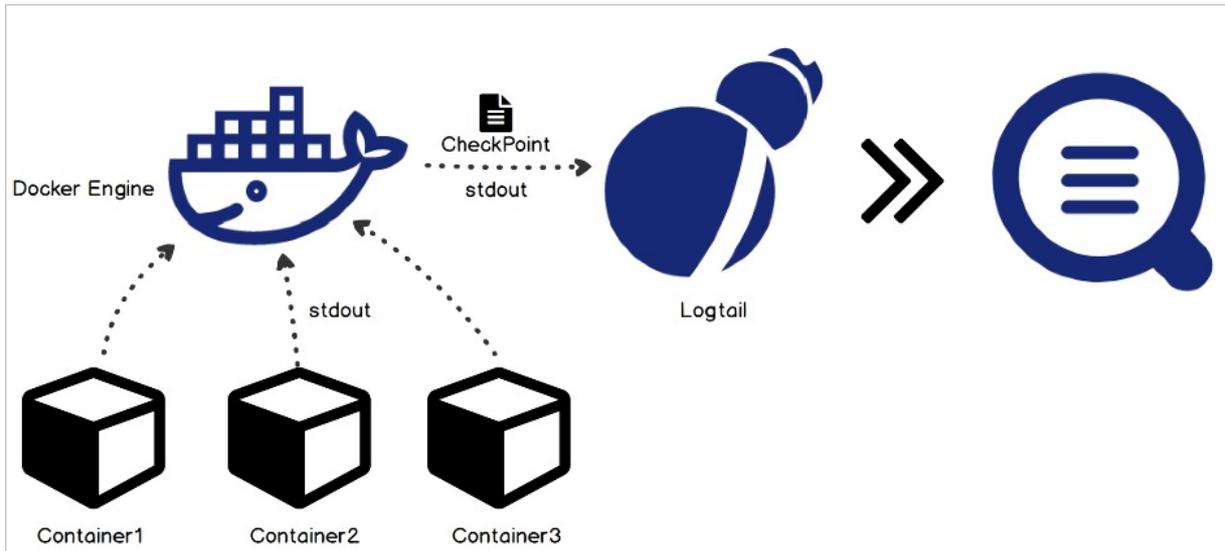
Features

- Collects stdout and xxx logs.
- Labels the containers from which you want to collect stdout and stderr logs.
- Uses tags to exclude specific containers.
- Allows you to use environment variables to specify the containers whose logs are to be collected.
- Allows you to use environment variables to exclude specific containers.
- Supports multiline logs such as Java Stack logs.
- Supports automatic labeling for container data.
- Supports automatic labeling for Kubernetes containers.

Implementation

As shown in the following figure, Logtail communicates with the domain socket of the Docker engine to query containers that run on the Docker engine. Logtail also locates containers from which you want to collect logs based on the specified labels and environment variables. Logtail then uses the docker logs command to collect logs from the specified containers.

When Logtail collects the stdout logs of a container, Logtail records information about log file positions to the checkpoint file at regular intervals. If Logtail is restarted, Logtail collects logs from the last log file position.



Limits

- This feature is available only for Logtail 0.16.0 or later that runs on Linux. For more information about Logtail versions and version updates, see [Install Logtail in Linux](#).
- The domain socket must exist and can access the Docker engine. Otherwise, Logtail cannot access the Docker engine by running the `/var/run/docker.sock` file.
- The last multiline log that you collect must be cached for at least 1,000 seconds. By default, the retention period for a multiline log is 3 seconds. You can specify the period by configuring the `BeginLineTimeoutMs` parameter. The value of the parameter must be a minimum of 1,000 ms. Otherwise, a false positive error may occur.
- Collection stop policy: When a container is stopped and Logtail detects the `die` event on the container, Logtail stops collecting stdout logs of the container. In this case, if a collection latency occurs, some stdout logs that are generated before the stop action may be lost.
- Docker log driver: Only log drivers of the JSON type are supported in the collection of stdout logs.
- Context: By default, a collection configuration is in the same context. If you need to configure different types of containers in different contexts, configure each type separately.
- Data processing: By default, collected logs start with the `content` field. You can apply standard processing methods to these logs. For more information about how to configure one or more processing methods, see [Configure data processing methods](#).
- Label: refers to the label information in docker inspect. It is not related to labels in Kubernetes configurations.
- Environment: refers to environment information that you specified during container startup.

Procedure

1. Deploy and configure Logtail on one or more containers.
2. Create a Logtail configuration and deliver it to Logtail.

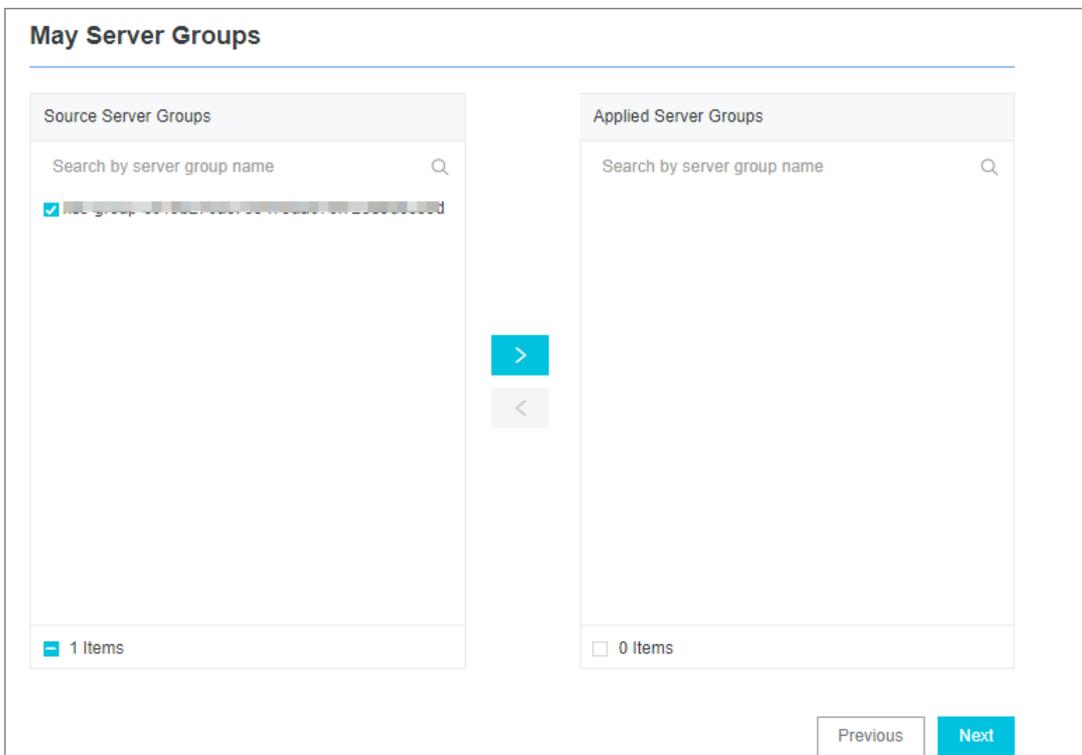
Logtail deployment and configuration

- Kubernetes
 - For more information about how to collect Kubernetes logs, see [Logtail deployment solution for collecting Kubernetes logs](#).
- Configure Logtail on other containers

For more information about how to configure Logtail on other containers, such as Swarm and Mesos, see [Common Logtail deployment solution for collecting Docker logs](#).

Configure a data source

1. [Log on to the Log Service console](#).
2. Click **Import Data**. On the **Import Data** page that appears, select **Docker Standard Output**.
3. Select a Logstore, and then click **Next**. Select an existing project and Logstore. You can also click **Create Now** to create a project and Logstore. For more information, see [Manage a Logstore](#).
If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.
4. Create a server group. Before you create a server group, ensure that Logtail is installed.
Install Logtail as prompted. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).
After you install Logtail, click **Complete Installation** to create a server group. For more information, see [Overview](#). If you have created a server group, click **Use Existing Server Groups** to select the server group.
5. Configure the server group, and then click **Next**. Select a server group and move the group from **Source Server Groups** to **Applied Server Groups**.



6. Configure a data source.
In the **Plug-in Config** section, set the required parameters. The following example shows how to set these parameters. For more information, see [Parameters](#).

```
{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeLabel": {
          "io.kubernetes.container.name": "nginx"
        },
        "ExcludeLabel": {
          "io.kubernetes.container.name": "nginx-ingress-controller"
        },
        "IncludeEnv": {
          "NGINX_SERVICE_PORT": "80"
        },
        "ExcludeEnv": {
          "POD_NAMESPACE": "kube-system"
        }
      }
    }
  ]
}
```

7. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Parameters

The type of the input source is `service_docker_stdout`.

 **Note** Before Logtail uploads data to Log Service, Logtail can process collected data. For more information about processing methods, see [Configure data processing methods](#).

Parameter	Type	Required	Description
IncludeLabel	JSON text. Key: JSON string. Value: JSON string.	Yes	<p>By default, this parameter is not specified. If the parameter is not specified, Logtail collects logs from all containers. If you set the key field and leave the value field empty, Logtail collects logs from containers whose labels include this key.</p> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the label of a container matches one of the key-value pairs, logs of the container are collected. Labels refer to docker labels. </div>

Parameter	Type	Required	Description
ExcludeLabel	JSON text. Key: JSON string. Value: JSON string.	No	<p>This parameter is not specified by default. If the parameter is empty, no containers are excluded. If the key is not empty but the value is empty, the containers whose labels include this key are excluded.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the label of a container matches one of the key-value pairs, the container is excluded. Labels described in this topic refer to Docker labels.
IncludeEnv	Map. Key: string. Value: string	No	<p>This parameter is empty by default. If the parameter is empty, logs of all containers are collected. If the key is not empty but the value is empty, logs of the containers whose environment variables include this key are collected.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. If the environment variable of a container includes one of the key-value pairs, the container is excluded. The environment variable refers to the environment information configured in container startup.
ExcludeEnv	Map. Key: string. Value: string	No	<p>This parameter is empty by default. If you leave the parameter empty, no container is excluded. If you set the key and leave the value empty, the containers whose environment variables include this key are excluded.</p> <p>Note</p> <ul style="list-style-type: none"> Key-value pairs are disjunctive with each other. When the environment variable of a container includes one of the key-value pairs, the container is excluded. The environment variable refers to the environment information configured in container startup.
Stdout	Boolean	No	If the value of the parameter is false, stdout data is not collected. Default value: true.
Stderr	Boolean	No	Default value: true. If the value of the parameter is false, stderr data is not collected.

Parameter	Type	Required	Description
BeginLineRegex	String	No	This parameter is not specified by default. If the parameter is not empty, the regular expression is used to match the first line of each log. If a line matches this regular expression, this line is assumed as the start of a new log. Otherwise, this line is assumed as part of the previous log.
BeginLineTimeoutMs	Integer	No	The timeout period for the regular expression to match a line. Default value: 3000. Unit: ms. If no new log appears within 3 seconds, the most recent log is uploaded.
BeginLineCheckLength	Integer	No	The length of data for the regular expression to match. Default value: 10×1024. Unit: bytes. You can set this parameter to check whether the beginning part of a line can match the regular expression. This improves matching efficiency.
MaxLogSize	Integer	No	The maximum length of a log. Default value: 512×1024. Unit: bytes. If the length exceeds this value, the log data is uploaded directly without finding the first line of logs.

 Note

- Labels defined in IncludeLabel and ExcludeLabel are different from those defined in Kubernetes. Labels in this topic refer to Docker labels.
- A namespace and a container name in Kubernetes can be mapped to Docker labels. The LabelKey parameter corresponding to a namespace is `io.kubernetes.pod.namespace`. The LabelKey parameter corresponding to a container name is `io.kubernetes.container.name`. For example, the namespace of the pod you created is `backend-prod` and the container name is `worker-server`. In this case, you can configure a whitelist label: `io.kubernetes.pod.namespace : backend-prod` or `io.kubernetes.container.name : worker-server`, so that only logs of the container are collected.
- In Kubernetes, we recommend that you use the `io.kubernetes.pod.namespace` and `io.kubernetes.container.name` labels. In other cases, use IncludeEnv or ExcludeEnv.

Default fields

- Common Docker containers

Each uploaded log contains the following fields.

Field	Description:
<code>_time_</code>	The data upload time. Example: <code>2018-02-02T02:18:41.979147844Z</code> .
<code>_source_</code>	The type of the input source. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	The IP address of the container.

- Kubernetes containers

Each uploaded log contains the following fields.

Field	Description
<code>_time_</code>	The data upload time. Example: <code>2018-02-02T02:18:41.979147844Z</code> .
<code>_source_</code>	The type of input sources. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace where the pod is located.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_id_</code>	The IP address of the pod.

Common configuration examples

- Environment configuration

Collect the logs of the container whose environment variable is `NGINX_PORT_80_TCP_PORT=80` but not `POD_NAMESPACE=kube-system` .

 **Note** The environment variable refers to the environment information configured in container startup.

Environment configuration example

```

openshift: false,
"StdinOnce": false,
"Env": [
  "HTTP_SVC_SERVICE_PORT_HTTP=80",
  "LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT=:8080",
  "LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PORT=8080",
  "HTTP_SVC_PORT_80_TCP_ADDR=",
  "NGINX_PORT_80_TCP=tcp://",
  "NGINX_PORT_80_TCP_PROTO=tcp",
  "LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_SERVICE_PORT=8080",
  "KUBERNETES_SERVICE_HOST=",
  "HTTP_SVC_SERVICE_HOST=",
  "HTTP_SVC_PORT_80_TCP_PROTO=tcp",
  "NGINX_PORT_80_TCP_ADDR=",
  "LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PROTO=tcp",
  "KUBERNETES_SERVICE_PORT_HTTPS=443",
  "KUBERNETES_PORT=tcp://:443",
  "NGINX_PORT=tcp://:80",
  "HTTP_SVC_PORT=tcp://:80",
  "HTTP_SVC_PORT_80_TCP_PORT=80",
  "NGINX_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP=tcp://:443",
  "KUBERNETES_PORT_443_TCP_PROTO=tcp",
  "HTTP_SVC_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP_ADDR=17.1.1",
  "HTTP_SVC_PORT_80_TCP=tcp://:80",

```

Collection configuration

```

{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeEnv": {
          "NGINX_PORT_80_TCP_PORT": "80"
        },
        "ExcludeEnv": {
          "POD_NAMESPACE": "kube-system"
        }
      }
    }
  ]
}

```

- Label configuration

Collect the stdout and stderr logs of the container whose label is `io.kubernetes.container.name=nginx` but not `type=pre`.

 **Note** Labels refer to Docker labels.

Label configuration example

```

"onBuild": null,
"Labels": {
  "annotation.io.kubernetes.container.hash": "53073f5a",
  "annotation.io.kubernetes.container.restartCount": "0",
  "annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
  "annotation.io.kubernetes.container.terminationMessagePolicy": "File",
  "annotation.io.kubernetes.pod.terminationGracePeriod": "30",
  "io.kubernetes.container.logpath": "/var/log/pods/ad00a078-85/nginx_0.log",
  "io.kubernetes.container.name": "nginx",
  "io.kubernetes.docker.type": "container",
  "io.kubernetes.pod.name": "example-foo-86ccd54874-r4mfh",
  "io.kubernetes.pod.namespace": "default",
  "io.kubernetes.pod.uid": "ad00a07",
  "io.kubernetes.sandbox.id": "5216-a8d0b6891dfa6da112969",
  "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
},
"StopSignal": "SIGTERM"

```

```

{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeLabel": {
          "io.kubernetes.container.name": "nginx"
        },
        "ExcludeLabel": {
          "type": "pre"
        }
      }
    }
  ]
}

```

Example of configuring multiline log collection

Configuring multiline log collection is important for the collection of Java exception stack logs. The following section introduces a standard collection configuration for Java stdout logs.

- Sample log

```

2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service
start
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : java.la
ng.NullPointerException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
...
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service
start done

```

- Collection configuration

Collect logs of the container whose label is `app=monitor` . The first line of each log to be collected is of the date type. To improve matching efficiency, only the first 10 bytes of each line are checked.

```
{
  "inputs": [
    {
      "detail": {
        "BeginLineCheckLength": 10,
        "BeginLineRegex": "\\d+-\\d+-\\d+.*",
        "IncludeLabel": {
          "app": "monitor"
        }
      },
      "type": "service_docker_stdout"
    }
  ]
}
```

Process collected data

Use a regular expression to extract the time, module, thread, class, and info fields.

- Collection configuration

Collect logs from a container whose label is `app=monitor` . The first line of each log to be collected is of the date type. To improve matching efficiency, only the first 10 bytes of each line are checked.


```

__tag__:__hostname__:logtail-dfgef
_container_name_:monitor
_image_name_:registry.cn-hangzhou.aliyuncs.xxxxxxxxxxxxxxx
_namespace_:default
_pod_name_:monitor-6f54bd5d74-rtzc7
_pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369
_source_:stdout
_time_:2018-02-02T14:18:41.979147844Z
time:2018-02-02 02:18:41.968
level:INFO
module:spring-cloud-monitor
thread:nio-8080-exec-4
class:c.g.s.web.controller.DemoController
message:service start done
    
```

30.3.1.7. Limits

This topic describes the limits on Logtail. These limits are related to log collection, resources, error handling, and other features.

Limits on log collection

Feature	Description
File encoding	Logs encoded in UTF-8 or GBK are supported. We recommend that you use UTF-8 to improve the performance of log processing. Log files encoded in other formats may result in unexpected errors, such as garbled characters and data loss.
Log file size	No limit.
Log rotation	The <code>.log*</code> and <code>.log</code> files are supported.
Log collection behavior upon log parsing block	If log parsing is disabled, Logtail keeps the file descriptor (FD) of the log open. If log rotation occurs multiple times during the block, Logtail attempts to parse each rotated log file in sequence. If the number of rotated logs to be parsed exceeds 20, Logtail does not process subsequent log files.
Soft link	Monitored directories can be soft links.
Size of a single log	The maximum size of a single log is 512 KB. If a multi-line log is divided by using a regular expression to match the first line, the maximum size of each log after division is still 512 KB. If the size of a log exceeds 512 KB, the log is forced to be separated into multiple parts for collection. For example, if a log is 1025 KB, it will be split into three parts: 512 KB, 512 KB, and 1 KB. These log parts are collected in sequence.
Regular expression	Regular expressions can be Perl-based regular expressions.
Applying multiple collection configurations to the same file	Not supported. We recommend that you collect log files and save them to one Logstore and configure multiple subscriptions. If this feature is required, configure soft links for log files to bypass the limit.
File opening behavior	Logtail keeps a log file to be collected open. Logtail automatically closes the log file if the file is not modified for more than 5 minutes (in case that log rotation does not occur).

Feature	Description
First log collection behavior	Logtail collects only incremental log files. If modifications are found in a file for the first time and the file size exceeds 1 MB, Logtail collects the logs from the last 1 MB. Otherwise, Logtail collects the logs from the beginning. If a log file is not modified after the configuration is issued, Logtail does not collect this file.
Non-standard text log	For a log that contains the characters \0, the log is truncated to the position where the characters \0 first appear.

Limits on checkpoints

Limit	Description
Checkpoint timeout interval	If a file remains unchanged for more than 30 days, the checkpoint is deleted.
Checkpoint reservation policy	Checkpoints are saved every 15 minutes and are automatically saved when the Logtail exits.
Checkpoint storage path	The default storage path is <code>/tmp/logtail_checkpoint</code> . You can modify the parameters according to Set Logtail startup parameters .

Limits on configurations

Limit	Description
Configuration updates	A custom configuration update requires about 30 seconds to take effect.
Dynamic loading for configurations	Supported. Updates on a collection configuration do not affect other collection configurations.
Number of collection configurations	No limit in theory. However, we recommend that the number of collection configurations for a server does not exceed 100.
Multi-tenant isolation	Collection configurations for different tenants are isolated.

Limits on resources and performance

Limit	Description
Throughput for log processing	The default throughput of processing raw logs is a maximum of 2 MB/s. After data is encoded and compressed, the data is uploaded. The compression ratio ranges from 5:1 to 10:1. If the actual throughput exceeds the limit, data loss may occur. You can modify the parameters according to Set Logtail startup parameters .
Maximum performance	Single-core capability: The maximum processing capability is 100 MB/s for logs in the simple mode, 20 MB/s for logs in the regular expression format, 40 MB/s for logs in the delimiter-separated format, and 30 MB/s for logs in the JSON format. The capability varies based on the complexity of regular expressions. After multiple processing threads are enabled, the performance can be improved by 1.5 to 3 times.
Number of monitored directories	Logtail sets a limit on the depth of monitored directories to avoid unnecessary consumption of user resources. If the limit is exceeded, Logtail stops monitoring extra directories and log files. Logtail can monitor a maximum of 3,000 directories (including subdirectories).

Limit	Description
Number of monitored files	<p>The maximum Logtail memory capacity can be increased to 2 GB. This means that the maximum number of files for a Logtail collection configuration on each server increases to 100,000 and the maximum number of files that each Logtail client can monitor increases to 1,000,000. Extra files are not monitored.</p> <p>When the limit is reached, you can:</p> <ul style="list-style-type: none"> • Increase the depth of the monitored directory in each Logtail configuration. • Modify the value of the <code>mem_usage_limit</code> parameter to increase the threshold for Logtail memory usage. For more information, see Set Logtail startup parameters. <p>The maximum capacity of memory that Logtail can use can be increased to 2 GB. It specifies that the maximum number of files that a Logtail collection configuration on each server increases to 100,000 and the maximum number of files that each Logtail client can monitor increases to 1,000,000.</p>
Default resources	<p>By default, Logtail consumes up to 40% of CPU usage and 256 MB of memory. If logs are generated at a high speed, you can modify relevant parameters. For more information, see Set Logtail startup parameters.</p>
Solutions to resource quota exceeded issues	<p>If the number of resources consumed by Logtail exceeds the limit and the issue lasts for more than 3 minutes, Logtail is forced to restart. Data loss or duplication issues may occur due to the restart.</p>

Limits on error handling

Limit	Description
Network error handling	<p>If a network error occurs, Logtail sends data again and automatically modify the retry interval.</p>
Resource quota exceeded	<p>If the data transmission rate exceeds the limit of a Logstore, Logtail stops log collection and automatically retries.</p>
Maximum retry period	<p>If Logtail fails to send data for more than six hours, Logtail discards the data.</p>
Status self-check	<p>Logtail automatically restarts each time an exception occurs, for example, application crashes and resource usage limits exceeded.</p>

More limits

Limit	Description
Log collection latency	<p>In most cases, after a log is collected, less than 1 second is required to write the log to a disk. This does not apply when log collection is disabled.</p>
Policies for uploading logs	<p>Logtail automatically uploads a log file when the number of logs in a log file exceeds 2,000, the size of the log file exceeds 2 MB, or Logtail collects logs for more than 3 seconds.</p>

30.3.2. Other collection methods

30.3.2.1. WebTracking

This topic describes how to use WebTracking to collect logs from websites that are written in HTML or HTML5, iOS, and Android.

Context

Log Service uses WebTracking to collect logs from websites written in HTML or HTML5, iOS, and Android. You can customize dimensions and metrics.



In the preceding figure, WebTracking allows you to collect user information from various browsers, iOS apps, and Android apps (except for SDK for iOS or Android). For example:

- Browsers, operating systems, and resolutions used by users.
- Browsing history such as preferences on different websites.
- The length of time that users stay on an app and whether the users are active in the app.

Precautions

- Dirty data may occur due to the use of WebTracking. This occurs because WebTracking allows unauthorized write access from Internet anonymous users to a Logstore.
- GET and POST requests are supported. The size of a request body cannot exceed 16 KB.
- The same limits apply to POST requests and the Putlogs API operation. The maximum number of logs that you can use a POST request to collect at a time is 4,096. The total size of these logs cannot exceed 3 MB.

Step 1: Enable WebTracking

Log on to the Log Service console and enable WebTracking.

1. [Log on to the Log Service console.](#)
2. Select the target Logstore, go to the **Logstore Attributes** page.
 - Method 1: Click the **Modify** icon under the name of the Logstore.
 - Method 2: Click the  icon next to the name of the Logstore, and select **Modify**.


```

import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.LogStore;
import com.aliyun.openservices.log.exception.LogException;
public class WebTracking {
    static private String accessId = "your accesskey id";
    static private String accessKey = "your accesskey";
    static private String project = "your project";
    static private String host = "log service data address";
    static private String logStore = "your logstore";
    static private Client client = new Client(host, accessId, accessKey);
    public static void main(String[] args) {
        try {
            //Enable WebTracking on an existing Logstore.
            LogStore logSt = client.GetLogStore(project, logStore).GetLogStore();
            client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), true));
            //Disable WebTracking.
            //client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), false));
            //Create a Logstore on which you want to enable WebTracking.
            //client.UpdateLogStore(project, new LogStore(logStore, 1, 1, true));
        }
        catch(LogException e){
            e.printStackTrace();
        }
    }
}

```

Step 2: Collect logs

After you enable WebTracking for a Logstore, you can use the following methods to upload data to the Logstore.

- Use the JavaScript SDK
 - i. Copy the *loghub-tracking.js* file to the *web* directory and add the following script to the file. Click [here](#) to copy the script.

```
<script type="text/javascript" src="loghub-tracking.js" async></script>
```

Note To ensure that a page loads, the script asynchronously sends HTTP requests. If data must be sent several times during the page loading, the newest request overwrites the previous HTTP request. A message showing WebTracking is about to exit appears in the browser. To eliminate the issue, send requests in a synchronous manner. To implement the method, perform the following step.

Original statement:

```
this.httpRequest_.open("GET", url, true)
```

Replace the original statement with the following statement:

```
this.httpRequest_.open("GET", url, false)
```

ii. Create a tracker.

```
var logger = new window.Tracker('${host}','${project}','${logstore}');
logger.push('customer', 'zhangsan');
logger.push('product', 'iphone 6s');
logger.push('price', 5500);
logger.logger();
logger.push('customer', 'lisi');
logger.push('product', 'ipod');
logger.push('price', 3000);
logger.logger();
```

The following table lists the parameters:

Parameter	Description
<code>\${host}</code>	The endpoint of the region where Log Service resides. For more information, see the <i>Obtain an endpoint topic in the Log Service Developer Guide</i> .
<code>\${project}</code>	The name of the project that you create in Log Service.
<code>\${logstore}</code>	The name of the Logstore in the <code>\${project}</code> .

After you run the following code, the following logs appear in Log Service.

```
customer:zhangsan
product:iphone 6s
price:5500
```

```
customer:lisi
product:ipod
price:3000
```

- Use HTTP GET requests

```
curl --request GET 'http://${project}.${host}/logstores/${logstore}/track? APIVersion=0.6.0&key1=val1&key2=val2'
```

The following table lists the parameters.

Parameter	Description
<code>\${project}</code>	The name of the project that you create in Log Service.
<code>\${host}</code>	The endpoint of the region where Log Service resides.
<code>\${logstore}</code>	The name of a Logstore that has WebTracking enabled in the <code>\${project}</code> .
<code>APIVersion=0.6.0</code>	(Required) A reserved parameter.
<code>__topic__=yourtopic</code>	(Optional) A reserved parameter that specifies the topic of the log.

Parameter	Description
<i>key1=val1, key2=val2</i>	The key-value pairs that you want to upload to Log Service. You can specify multiple pairs. Make sure that the length of each request URL is less than 16 KB.

- Use HTML img tags

```
<img src='http://${project}.${host}/logstores/${logstore}/track.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>  
<img src='http://${project}.${host}/logstores/${logstore}/track_ua.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>
```

The parameters that you need to specify are the same as the preceding parameters. In addition to custom parameters that are uploaded by track_ua.gif, Log Service also uses UserAgent and referer fields in the HTTP header as the fields of logs.

30.3.2.3. Use SDKs to collect logs

30.3.2.3.1. Producer Library

The Aliyun LOG Java Producer supports Java applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable.

For more information about the related GitHub project, visit [Aliyun LOG Java Producer](#).

30.3.2.3.2. Log4j Appender

Log4j is an open-source logging framework of Apache. You can use Log4j to write logs to the Log Service console, files, graphical user interface (GUI) components, socket servers, NT kernel loggers, or Unix syslog daemons. You can specify the output format of each log. You can also specify the severity level of each log to implement a fine-grained control on log generation.

Log4j consists of three components: loggers, appenders, and layouts.

- Loggers allow you to specify the severity level of each log.
Severity levels are sorted into ERROR, WARN, INFO, and DEBUG in descending order of severity.
- Appenders allow you to specify the destination of each log.
A destination can be the Log Service console or a file.
- Layouts allow you to specify the output format of each log.
The output format defines how logs are displayed.

To write logs to Log Service, use the Alibaba Cloud Log Log4j Appender. For information about where to download the library and how to use it, see [Log4j Appender](#).

30.3.2.3.3. Logback Appender

Logback was created by the same developer of Log4j. Logback allows you to write logs to multiple destinations. These destinations include the Log Service console, files, graphical user interface (GUI) components, socket servers, NT kernel loggers, and Unix syslog daemons. You can define the output format of each log. If you define the severity level of each log, you can implement a fine-grained control on the log generation process.

You can set the destination of logs to Log Service by using the Aliyun Log Logback Appender. The following example shows the format of logs that are uploaded to Log Service.

```
level: ERROR
location: com.aliyun.openservices.log.logback.example.LogbackAppenderExample.main(LogbackAppenderExample.java
:18)
message: error log
throwable: java.lang.RuntimeException: xxx
thread: main
time: 2018-01-02T03:15+0000
log: 2018-01-02 11:15:29,682 ERROR [main] com.aliyun.openservices.log.logback.example.LogbackAppenderExample: err
or log
__source__: xxx
__topic__: yyy
```

For information about where to download the library and how to use it, see [Logback Appender](#).

30.3.2.3.4. Golang Producer Library

The Aliyun LOG Go Producer Library supports Go applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable. You can use the library to create producers that allow you to resend failed logs. Before Go applications send log data to Log Service, you can use these producers to compress the log data. This improves write performance.

For more information about the related GitHub project, visit [Aliyun Log Go Producer](#).

30.3.2.3.5. Python logging

Configurations

For more information about the configurations that are related to the Python logging module, see [Logging configuration](#).

The Python logging module allows you to configure logging by using code or a configuration file. The following example shows how to configure logging by using the `logging.conf` configuration file.

```
[loggers]
keys=root,sls

[handlers]
keys=consoleHandler,slsHandler

[formatters]
keys=simpleFormatter,rawFormatter

[logger_root]
level=DEBUG
handlers=consoleHandler

[logger_sls]
level=INFO
handlers=consoleHandler,slsHandler
qualname=sls
propagate=0

[handler_consoleHandler]
class=StreamHandler
level=DEBUG
formatter=simpleFormatter
args=(sys.stdout,)

[handler_slsHandler]
class=aliyun.log.QueuedLogHandler
level=INFO
formatter=rawFormatter
args=(os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_PROJECT', ''), "logstore")

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(message)s

[formatter_rawFormatter]
format=%(message)s
```

Two handlers named `root` and `sls` are created. The `sls` handler is an object of the `aliyun.log.QueuedLogHandler` class. The following shows the parameters that are specified for the `sls` handler. For more information, see [Parameters](#).

```
args=(os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_PROJECT', ''), "logstore")
```

 **Note** In this case, the `os.environ` function is used to retrieve configurations from environment variables. You can also specify values for these parameters based on your business requirements.

Upload logs

You can use the configuration file to upload logs to Log Service.

```
import logging
import logging.config

# Configurations
logging.config.fileConfig('logging.conf')
logger = logging.getLogger('sls')

# Use the logger
logger.info("test1")

try:
    1/0
except ZeroDivisionError as ex:
    logger.exception(ex)
```

Then, logs are automatically uploaded to Log Service. To use the LogSearch/Analytics feature, you must enable the index feature on the corresponding Logstore.

Configure an index for a Logstore

Enable the index feature on the Logstore that receives logs and configure an index for specific fields. We recommend that you use CLI (Command Line Interface) to configure the index as follows:

```
aliyunlog log update_index --project_name="project1" --logstore_name="logstore1" --index_detail="file:///Users/user1/loghandler_index.json"
```

For more information, see the [python_logging_handler_index.json](#) configuration file.

Specify log fields to be collected

The following table lists supported log fields that you can collect. By default, all of the fields are collected.

Field	Description
message	The contents of a log.
record_name	The name of a handler. In the preceding example, the name is <code>sls</code> .
level	The output level of a logger, such as INFO and ERROR.
file_path	The full path of a configuration file.

Field	Description
func_name	The name of a function.
line_no	The number of a log line.
module	The name of a module where the function resides.
thread_id	The ID of the thread that runs the function.
thread_name	The name of the thread that runs the function.
process_id	The ID of the process that runs the function.
process_name	The name of the process that runs the function.

You can specify log fields to be collected based on the `fields` parameter of the `QueuedLogHandler` class. For more information, see [aliyun.log.LogFields](#).

The following example shows how to modify the preceding configuration file and collect several fields, such as `module` and `func_name`.

```
[handler_slsHandler]
class=aliyun.log.QueuedLogHandler
level=INFO
formatter=rawFormatter
args=('cn-beijing.log.aliyuncs.com', 'ak_id', 'ak_key', 'project1', "logstore1", 'mytopic', ['level', 'func_name', 'module', 'line_no'] )
```

Note

- The message field is collected regardless of your configurations.
- To add a prefix and suffix to the names of these fields, use the `buildin_fields_prefix` and `buildin_fields_suffix` parameters. For example, `__level__`.

Configure logging by using a JSON text

You can use a JSON text to create more flexible logging configurations than code does.

```

#encoding: utf8
import logging, logging.config, os

# Configurations
conf = {'version': 1,
       'formatters': {'rawformatter': {'class': 'logging.Formatter',
                                       'format': '%(message)s'}
                      },
       'handlers': {'sls_handler': {'():
                                   'aliyun.log.QueuedLogHandler',
                                   'level': 'INFO',
                                   'formatter': 'rawformatter',

                                   # custom args:
                                   'end_point': os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ""),
                                   'access_key_id': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ""),
                                   'access_key': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ""),
                                   'project': 'project1',
                                   'log_store': "logstore1"
                                   }
                    },
       'loggers': {'sls': {'handlers': ['sls_handler'],
                           'level': 'INFO',
                           'propagate': False}
                   }
               }
logging.config.dictConfig(conf)

# Use the logger
logger = logging.getLogger('sls')
logger.info("Hello world")

```

 **Note** To instantiate an object of the `aliyun.log.QueuedLogHandler` class, pass named parameters to the constructor. For more information, see [Parameters](#).

30.3.2.4. Common log formats

30.3.2.4.1. Log4j logs

Log Service allows you to collect Log4j logs.

Collect Log4j logs by using LogHub Log4j Appender

For more information, see [Log4j Appender](#).

Configure Logtail to collect Log4j logs

This topic describes how to configure regular expressions based on the default configuration of Log4j 1 logs. If Log4j 2 is used, you must modify the default configuration to record the complete date information. Log4j logs are sorted into Log4j 1 logs and Log4j 2 logs.

```
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss:SSS zzz} [%t] %-5level %logger{36} - %msg%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="com.foo.Bar" level="trace">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

For more information about how to configure Logtail to collect Log4j logs, see [Python logs](#). Configure the required parameters based on your network environment and business requirements.

The automatically generated regular expression is based on the sample log and may not be suitable for other logs. Therefore, you must make minor changes to the regular expression after it is automatically generated

The following shows a sample log of the default Log4j format.

```
2013-12-25 19:57:06,954 [10.207.37.161] WARN impl.PermanentTairDaoImpl - Fail to Read Permanent Tair,key:e:47021731
9319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1,msg=connection error or timeout,value=,flag=0]
```

Regular expression that matches IP addresses that each indicate the start of a line:

```
\d+-\d+-\d+\s.*
```

Regular expression used to extract log information:

```
(\d+-\d+-\d+\s\d+:\d+:\d+,\d+)\s{1,}([^\s]*)\s{1,}(\S+)\s+(\S+)\s-\s{1,}(.*)
```

Time conversion format:

```
%Y-%m-%d %H:%M:%S
```

The following table lists the extraction results of the sample log.

Key	Value
time	2013-12-25 19:57:06,954
ip	10.207.37.161
level	WARN

Key	Value
class	impl.PermanentTairDaoImpl
message	Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]

30.3.2.4.2. Python logs

The Python logging module provides a general logging system, which can be used by third-party modules or applications.

The Python logging module provides different log levels and logging methods, such as file-based, HTTP GET, HTTP POST, SMTP, and Socket logging. You can also create a custom logging method. The Python logging module works in the same way as the Log4j logging module except for some implementation details. The Python logging module includes the logger, handler, filter, and formatter objects.

To collect Python logs, we recommend that you use log handlers.

- Use log handlers to automatically upload Python logs
- Use log handlers to automatically parse logs that consist of key-value pairs
- Use log handlers to automatically parse logs in the JSON format

Log format

A formatter specifies the output format of logs. To instantiate a formatter, pass two parameters to the constructor. One parameter includes a message format string and the other parameter includes a date format string. The parameters are optional.

Log format:

```
import logging
import logging.handlers
LOG_FILE = 'tst.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes = 1024*1024, backupCount = 5) # Instantiate the handler
formatter = logging.Formatter('%(asctime)s - %(filename)s:%(lineno)s - %(name)s - %(message)s') # Instantiate the formatter
handler.setFormatter(formatter) # Add the formatter to the handler
logger = logging.getLogger('tst') # Obtain a logger named tst
logger.addHandler(handler) # Add the handler to the logger
logger.setLevel(logging.DEBUG)
logger.info('first info message')
logger.debug('first debug message')
```

Attributes

Formatter attributes are specified in the `%(key)s` format. The following table lists the attributes.

Format	Description
%(name)s	The name of a logger that generates logs.
%(levelno)s	The log output level in the numeric format. Valid values: DEBUG, INFO, WARNING, ERROR, and CRITICAL

Format	Description
%(levelname)s	The log output level in the text format. Valid values: 'DEBUG', 'INFO', 'WARNING', 'ERROR', and 'CRITICAL'.
%(pathname)s	The full path of a source file that contains the logging module.
%(filename)s	The name of the source file.
%(module)s	The name of a module where the statement that you use to generate logs resides.
%(funcName)s	The name of the function that is used to call the log output function.
%(lineno)d	The number of a code line that contains the statement used to call the log output function.
%(created)f	The time when the log was created. The value is a UNIX timestamp representing the number of seconds that have elapsed since January 1, 1970, 00:00:00 (UTC).
%(relativeCreated)d	The interval between the time when a log was created and the time when the logging module was loaded. Unit: milliseconds.
%(asctime)s	The time when the log was created. The value of 2003-07-08 16:49:45,896 is an example of the default format. The number after the comma (,) indicates the number of milliseconds.
%(msecs)d	The time when the log was created. The value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970, 00:00:00 (UTC).
%(thread)d	The thread ID.
%(threadName)s	The thread name.
%(process)d	The process ID.
%(message)s	The contents of a log.

Sample logs

Sample log:

```
2015-03-04 23:21:59,682 - log_test.py:16 - tst - first info message
2015-03-04 23:21:59,682 - log_test.py:17 - tst - first debug message
```

Common Python logs and the corresponding regular expressions:

- Sample log:

```
2016-02-19 11:03:13,410 - test.py:19 - tst - first debug message
```

Regular expression:

```
(\d+-\d+-\d+\s\S+)\s+-\s+([\^:]+):(\d+)\s+-\s+(\w+)\s+-\s+(. *)
```

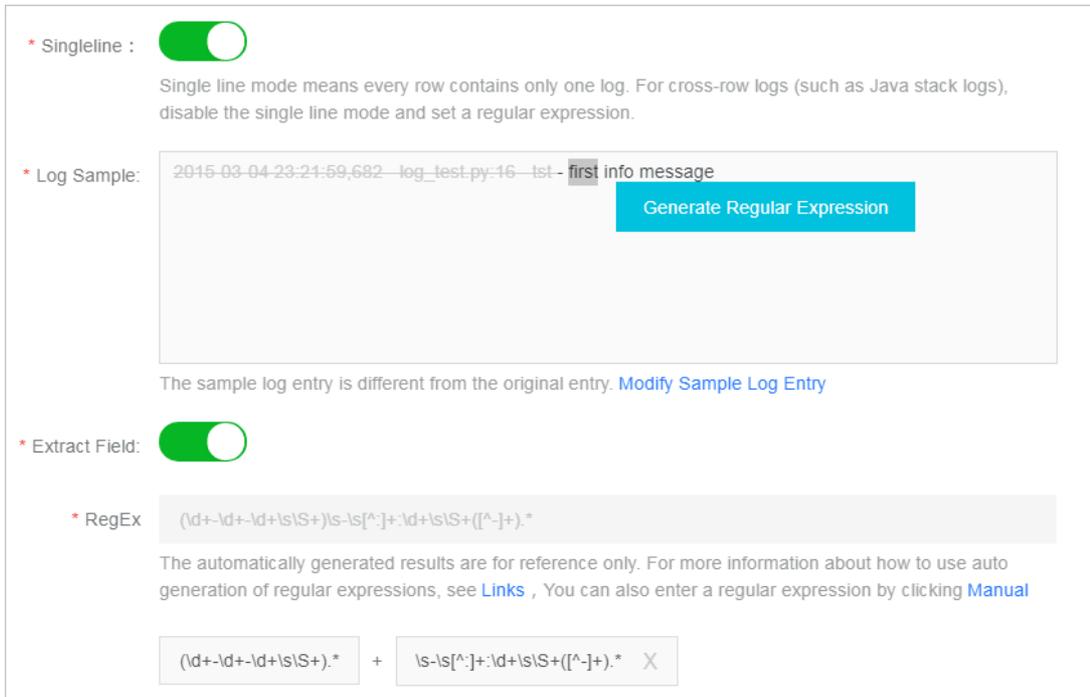
- Log format

```
%(asctime)s - %(filename)s:%(lineno)s - %(levelname)s %(message)s
%(asctime)s - %(filename)s:%(lineno)s - %(levelname)s %(pathname)s %(module)s %(funcName)s %(created)f %(thread)d %(threadName)s %(process)d %(name)s - %(message)s
```


v. Set a regular expression in the RegEx field.

a. Select fields to generate a regular expression

If the regular expression that is automatically generated does not match your sample log, you can select fields in the sample log to generate a regular expression. Log Service can automatically parse the highlighted fields of the sample log to generate a regular expression. In the **Log Sample** field, select the required fields, and click **Generate Regular Expression**. The regular expression of the selected field is displayed in the **RegEx** field. To obtain a full regular expression for the sample log, generate regular expressions for each log field.



* Singleline :

Single line mode means every row contains only one log. For cross-row logs (such as Java stack logs), disable the single line mode and set a regular expression.

* Log Sample: 2015-03-04 23:21:59,682 log_test.py:16 tst - first info message

Generate Regular Expression

The sample log entry is different from the original entry. [Modify Sample Log Entry](#)

* Extract Field:

* RegEx (\\d+\\d+\\d+\\s\\S+)\\s-\\s[^:]+:\\d+\\s\\S+([^-]+).*

The automatically generated results are for reference only. For more information about how to use auto generation of regular expressions, see [Links](#) , You can also enter a regular expression by clicking [Manual](#)

(\\d+\\d+\\d+\\s\\S+).* + \\s-\\s[^:]+:\\d+\\s\\S+([^-]+).* X

b. Modify the regular expression

Actual data formats may vary. In this case, click **Manual** under the RegEx field to adjust the regular expression that is automatically generated based on your business requirements. This ensures that the regular expression is suitable for all formats of the collected logs.

c. Verify the regular expression

After you modify the regular expression, click **Validate** next to the RegEx field. If the regular expression is valid, the extraction results are displayed. If the regular expression is invalid, modify the regular expression again.

- vi. Confirm the extraction results of log fields. View the extraction results of log fields and specify keys for the extracted fields.

Specify an informative name for each log field in the extraction results. For example, time as the name for a time field. If you do not use the system time, you must specify the name of a time field in the Value fields and time in the Key field.

* Extracted Content:

Key	Value
asctime	2015-03-04 23:21:59
filename	682 - log_test.py
lineno	16
name	tst
message	first info message

When you use a regular expression to generate key/value pairs, you can specify the key name in each pair. If you do not specify system time, you must specify a pair that uses "time" as the key name.

- 7. (Optional)Specify Advanced Options and click Next.Specify Advanced Options based on your business requirements. We recommend that you do not modify the default settings unless otherwise required.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
Upload Raw Log	Specifies whether to upload raw logs. If you turn on this switch, raw logs are written to the <code>__raw__</code> field and uploaded with the parsed logs.
Topic Generation Mode	<ul style="list-style-type: none"> ◦ Null - Do not generate topic: This mode is selected by default. In this mode, the topic is set to an empty string and you can query logs without the need to enter a topic. ◦ Server Group Topic Attributes: This mode is used to differentiate log data that is generated by different frontend servers. ◦ File Path RegEx: If you select this mode, you must enter a value in the Custom RegEx field to extract part of the path as the topic. This mode is used to differentiate log data that is generated by users or instances.
Custom RegEx	Specifies a custom regular expression. If you select File Path RegEx for Topic Generation Mode , you must enter a custom regular expression.
Log File Encoding	<ul style="list-style-type: none"> ◦ utf8: indicates UTF-8 encoding. ◦ gbk: indicates GBK encoding.
Timezone	<p>Specifies the time zone where logs are collected.</p> <ul style="list-style-type: none"> ◦ System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. ◦ Custom: Select a time zone.

Parameter	Description
Timeout	<p>If a log file is not updated within a specific period of time, Logtail considers the file to be timed out.</p> <ul style="list-style-type: none"> Never: All log files are continuously monitored and never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file.
Filter Configuration	<p>Only logs that meet all filter conditions are collected.</p> <p>Examples:</p> <ul style="list-style-type: none"> Collect logs that meet a condition: Set the condition to <code>Key:level Regex:WARNIN G ERROR</code>. It indicates that only logs with the severity level of WARNING or ERROR are collected. Filter logs that do not meet a condition: <ul style="list-style-type: none"> Set the condition to <code>Key:level Regex:^(?!.*(INFO DEBUG)).*</code>. It indicates that logs with the severity level of INFO or DEBUG are not collected. Set the condition to <code>Key:url Regex:.^(?!.*(healthcheck)).*</code>. It indicates that logs whose URL contains the keyword healthcheck are not collected. For example, logs in which the key is url and the value is <code>/inner/healthcheck/jiangong.html</code> are not collected.

8. Configure an index. Configure an index based on your business requirements. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

After the configuration is complete, apply the settings to the server group to collect Python logs.

30.3.2.4.3. Node.js logs

Node.js logs are displayed in the Log Service console by default. This impedes data collection and troubleshooting. You can use the log4js function to write logs into files and customize the log format. This facilitates data collection and consolidation.

Example:

```
var log4js = require('log4js');
log4js.configure({
  appenders: [
    {
      type: 'file', //Output to a file
      filename: 'logs/access.log',
      maxLogSize: 1024,
      backups: 3,
      category: 'normal'
    }
  ]
});
var logger = log4js.getLogger('normal');
logger.setLevel('INFO');
logger.info("this is a info msg");
logger.error("this is a err msg");
```

Log format

After logs are written to text files by using the log4js function, these logs are displayed in the following format:

```
[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
[2016-02-24 17:42:38.951] [ERROR] normal - this is a err msg
```

The log4js function defines six log severity levels. They are TRACE, DEBUG, INFO, WARN, ERROR, and FATAL in ascending order of severity.

Use Logtail to collect Node.js logs

For more information about how to configure Logtail to collect Python logs, see [Python logs](#). Use configurations based on your network environment and business requirements.

The automatically generated regular expression is based on the sample log and may not apply to other logs. Therefore, you must make minor changes to the regular expression after it is generated. You can use the following sample Node.js logs to configure appropriate regular expressions for your logs.

Sample Node.js logs:

- Example 1

- Sample log

```
[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
```

- Regular expression:

```
\\([\\^]+)\\s\\([\\^\\]+)\\s(\\w+)\\s-(. *)
```

- Extracted fields:

```
time , level , loggerName , and message
```

- Example 2

- Sample log

```
[2016-01-31 12:02:25.844] [INFO] access - 42.120.73.203 - - "GET /user/projects/ali_sls_log? ignoreError=true HTTP/1.1" 304 - "http://aliyun.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

- Regular expression:

```
\\([\\^]+)\\s\\([\\w+])\\s(\\w+)\\s-\\s(\\S+)\\s-\\s-\\s"([\\^"]+)"\\s(\\d+)[\\^"]+("[\\^"]+)"\\s"([\\^"]+). *
```

- Extracted fields:

```
time , level , loggerName , ip , request , status , referer , and user_agent
```

30.3.2.4.4. WordPress logs

This topic describes the format of WordPress logs and extraction results of a sample log.

Log format

Sample log:

```
172.64.0.2 - - [07/Jan/2016:21:06:39 +0800] "GET /wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0" 200
776 "http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mo
zilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.
36"
```

Configure Logtail to collect WordPress logs

Configurations required to collect WordPress logs:

- Regular expression that matches IP addresses that each indicate the start of a line

```
\d+\.\d+\.\d+\.\d+\s-\s.*
```

- Regular expression used to extract information from the log:

```
(\S+) - - \[([^\]]*)] "(\S+) ([^"]+)" (\S+) (\S+) "([^\"]+)" "([^\"]+)"
```

- Time conversion format:

```
%d/%b/%Y:%H:%M:%S
```

- Results after Logtail extracts information from the sample log

Key	Value
ip	10.10.10.1
time	07/Jan/2016:21:06:39 +0800
method	GET
url	/wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0
status	200
length	776
ref	http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36

30.3.2.4.5. Unity3D logs

Log Service can use the WebTracking feature to collect Unity3D logs. The following example shows how to collect *Unity logs* of the debug type.

Context

Unity3D is a cross-platform game engine developed by Unity Technologies. The engine allows you to create 3D video games, VR buildings, real-time 3D animation, and other interactive content.

Procedure

1. Enable the WebTracking feature. For more information about how to enable this feature, see [WebTracking](#).
2. Create a Unity3D LogHandler. In a Unity editor, create a C# file named *LogOutputHandler.cs*, add the following code, and modify the following variables.
 - `project`: specifies the name of the project.

- `logstore`: specifies the name of the Logstore.
- `serviceAddr`: specifies the endpoint of the project.

For more information about `serviceAddr`, see [View the information of a project](#).

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour
{
    //Register the HandleLog function on scene start to fire on debug.log events
    public void OnEnable()
    {
        Application.logMessageReceived += HandleLog;
    }
    //Remove callback when object goes out of scope
    public void OnDisable()
    {
        Application.logMessageReceived -= HandleLog;
    }
    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "http address of your log service project";
    //Capture debug.log output, send logs to Loggly
    public void HandleLog(string logString, string stackTrace, LogType type)
    {
        string parameters = "";
        parameters += "Level=" + WWW.EscapeURL(type.ToString());
        parameters += "&";
        parameters += "Message=" + WWW.EscapeURL(logString);
        parameters += "&";
        parameters += "Stack_Trace=" + WWW.EscapeURL(stackTrace);
        parameters += "&";
        //Add any User, Game, or Device MetaData that would be useful to finding issues later
        parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/logstores/" + logstore + "/track? APIVersion=0.6.0&" + pa
rameters;
        StartCoroutine(SendData(url));
    }
    public IEnumerator SendData(string url)
    {
        WWW sendLog = new WWW(url);
        yield return sendLog;
    }
}
```

The preceding code allows you to send logs to Log Service in an asynchronous manner. In the code, you can specify more fields you want to collect.

3. Generate Unity logs. In the project, create a C# file named *LogglyTest.cs* and add the following code.

```
using UnityEngine;
using System.Collections.Generic;
public class LogglyTest : MonoBehaviour {
    void Start () {
        Debug.Log ("Hello world");
    }
}
```

4. View logs in the console.

After you complete the preceding steps, run the Unity application. In the Log Service console, view logs that are sent to Log Service.

The preceding code shows how to use *Debug.Log*, *Debug.LogError*, and *Debug.LogException* methods to collect logs. Unity provides Component Object Model (COM)-based exception handling and log handling APIs. These APIs allow you to easily collect device details of clients.

30.4. Query and analysis

30.4.1. Overview

Log Service provides the LogSearch/Analytics feature that you can use to query and analyze a large number of logs. If you do not enable indexes, raw data is consumed in sequence based on shards. The procedure is similar to the sequential consumption of Kafka messages. If you enable indexes, you can query logs and perform statistical analysis on query results in addition to consuming logs in sequence.

Benefits

- **Real-time:** Logs can be analyzed immediately after they are written.
- **Fast:**
 - **Query:** Billions of data records can be processed and queried within one second. Each search statement has a maximum of five conditions specified.
 - **Analysis:** Hundreds of millions of data records can be aggregated and analyzed within one second. Each query has a maximum of five aggregate functions and a GROUP BY clause specified.
- **Flexible:** Query and analysis conditions can be changed as required and the results are returned in real time.
- **All-in-one:** Reports and dashboards are available in the console for quick analysis. In addition to these features, Log Service can work together with Grafana, DataV, Jaeger, and other services. It also supports RESTful APIs, Java Database Connectivity (JDBC) APIs, and other APIs.

Indexing

Indexes refer to a data structure that you can use to sort the values of one or more columns of logs. Indexes allow you to obtain the required information in a timely manner from logs that Log Service collects. Before you use the LogSearch/Analytics feature, you must collect logs and [Enable the index feature and configure indexes for a Logstore](#) on the collected logs.

In Log Service, indexes are sorted into **full-text indexes** and **field-specific indexes**.

- **Full-text index:** Indexing is enabled for the full contents of a log. The values of all fields in a log are queried by default. The log can be queried if one of the fields matches the search term.
- **Field-specific index:** You can configure a field-specific index for a key. Then, you can query logs based on specific keys to narrow the query scope.

To use **field-specific indexes**, you must specify the data type for a field. Available data types for fields in Log Service include **Text**, **JSON**, **Long**, and **Double**. For more information about [Overview](#).

Query methods

- Console

In the Log Service console, you can query logs by specifying time ranges and search statements. For more information about the procedure and search statements, see [Query logs](#) and [Query syntax](#).

- API

To query logs, you can call the `GetLogs` and `GetHistograms` API operations of the Log Service API.

 **Note** Before you query logs, make sure that you collect logs and [Enable the index feature and configure indexes for a Logstore](#).

Search and analysis statements

To apply real-time LogSearch/Analytics to collected logs, you must specify query statements. Each query statement includes the search section and the analytics section. Separate the sections with a vertical bar (`|`).

```
$Search|$Analytics
```

Statement	Required	Description
Search	No	A search statement contains search conditions. These conditions include keywords, fuzzy keywords, values, ranges, and combined conditions. If you leave the statement empty or specify an asterisk (*) for the statement, it indicates that no condition is specified and all data is returned. For more information, see Query syntax .
Analytics	No	You can use an analytics statement to aggregate or analyze data based on query results. If you leave the statement empty, it indicates that no analytics is required and all query results are returned. For more information, see Real-time analysis .

Precautions

You may query a large number of logs. For example, if the number of logs to be queried is more than 1,000,000,000, Log Service may fail to return all results. Log Service returns a partial result set and notifies you that the returned data set includes partial results.

Query results are cached every 15 minutes. If a partial result set is matched in the cache, Log Service continues to scan logs that are not cached. Log Service combines query results of the current query with results of cached results.

Therefore, Log Service enables you to obtain results by calling the API operation multiple times with the same parameters.

30.4.2. Real-time analysis

Log Service supports SQL-like aggregate calculation. This feature integrates search statements with SQL aggregate functions to calculate query results.

Sample statement:

```
status>200|select avg(latency),max(latency) ,count(1) as c GROUP BY method ORDER BY c DESC LIMIT 20
```

Basic syntax:

```
[search query] | [sql query]
```

Separate a search statement and a calculation statement with a vertical bar (|). You can use the search statement to query logs and obtain the required results. Then, use the calculation statement for further aggregation. The search query syntax is specific to Log Service. For more information, see [Query syntax](#).

Prerequisites

To use the statistical analysis feature, click **Index Attributes**. Turn on the **Enable Analytics** switch for the required field. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

- If you turn off the switch, Log Service calculates up to 10,000 rows of data in each shard along with high latency of calculation.
- If you turn on the switch, Log Service analyzes data in seconds.
- The update applies only to new data.
- No extra fee is incurred for the update.

Supported SQL syntax

Log Service supports the following SQL syntax. For more information about a specific topic, click the corresponding link.

- Aggregate functions that are available for SELECT statements include:
 - [General aggregate functions](#)
 - [Security check functions](#)
 - [Map functions](#)
 - [Approximate functions](#)
 - [Mathematical statistics functions](#)
 - [Mathematical calculation functions](#)
 - [String functions](#)
 - [Date and time functions](#)
 - [URL functions](#)
 - [Regular expression functions](#)
 - [JSON functions](#)
 - [Type conversion functions](#)
 - [IP functions](#)
 - [Array functions](#)
 - [Binary string functions](#)
 - [Bitwise operations](#)
 - [Interval-valued comparison and periodicity-valued comparison functions](#)
 - [Comparison functions and operators](#)
 - [Lambda functions](#)
 - [Logical functions](#)
 - [Geospatial functions](#)
 - [Geography functions](#)
 - [Machine learning functions](#)
- [GROUP BY syntax](#)
- [Window functions](#)
- [HAVING syntax](#)
- [ORDER BY syntax](#)
- [LIMIT syntax](#)
- [Syntax for CASE statements and if\(\) functions](#)
- [UNNEST function](#)

- [Field aliases](#)
- [Nested subqueries](#)

Precautions

Before you use SQL statements, note the following items:

- You do not need to specify FROM and WHERE clauses for SQL statements. By default, Log Service queries logs from the current Logstore, and each WHERE clause is specified in the [search query] section.
- The supported clauses include SELECT, GROUP BY, ORDER BY [ASC,DESC], LIMIT, and HAVING.

 **Note** By default, only the first 10 results are returned. If you want to return more results, add a LIMIT clause to the statement. For example, `* | select count(1) as c, ip group by ip order by c desc limit 100`.

Built-in fields

Log Service has multiple built-in fields for statistical analysis. A built-in field is automatically added to a valid column that you create.

Field	Type	Description
<code>__time__</code>	Bigint	The time when a log was created.
<code>__source__</code>	Varchar	The source IP address of a log. When you query logs, the name of the field is source. If you specify the field for an SQL statement, you must add two underscores (__) at both the start and end of source.
<code>__topic__</code>	Varchar	The topic of a log.

Limits

- Maximum number of Logstores from which you can query logs at the same time is 15.
- Maximum length for a field value of the varchar type is 2,048. Extra data will be truncated.
- By default, up to 100 lines of a log file are returned and pagination is not supported. If you want to return more lines, use [LIMIT syntax](#).

Example

Calculate the hourly PV and UV, and the user request of the highest latency.

```
*|select date_trunc('hour',from_unixtime(__time__)) as time,
count(1) as pv,
approx_distinct(userid) as uv,
max_by(url,latency) as top_latency_url,
max(latency,10) as top_10_latency
group by 1
order by time
```

30.4.3. Enable the indexing feature and configure indexes for a Logstore

This topic describes how to enable the indexing feature and configure indexes for a Logstore.

Context

Before you can query logs that are stored in a Logstore, you must enable the indexing feature and configure indexes for the Logstore. We recommend that you configure indexes for your Logstores based on your business requirements.

 **Note** After you enable the indexing feature, the indexes occupy extra storage space and transferring the indexes occupies extra bandwidth.

When you collect a log entry, Log Service adds the relevant information (such as the source and time fields) to the log entry as key-value pairs. These fields are reserved in Log Service. If you enable the indexing feature for a Logstore and configure indexes for fields in the Logstore, the indexing and analytics features are automatically enabled for these fields.

Reserved fields in Log Service

Field	Description
<code>__topic__</code>	The topic of a log entry. If you specify a topic for a log entry, Log Service adds the topic field to the log entry. The key of the field is <code>__topic__</code> and the value of the field is the log topic. For more information, see Specify a log topic .
<code>__source__</code>	The source of a log entry. The source device that generates the log entry.
<code>__time__</code>	The time when a log entry is written to the Logstore by using an SDK.

 **Note** If the values of the `__topic__` and `__source__` fields are null, the keywords that you use to query the two fields must exactly match the field values.

Procedure

1. [Log on to the Log Service console](#).
2. In the **Projects** section, click the target project.
3. Find the target Logstore, and choose  **Search & Analysis**.
4. On the page that appears, click **Enable** in the upper-right corner.

 **Note** If you have created indexes for the Logstore, choose **Index Attributes > Modify** to modify the indexes.

5. Configure the indexes.

 **Note** If you enable a full-text index and a field-specific index at the same time, the field-specific index takes precedence over the full-text index.

Index types

Index type	Description
Full text index	An index is created in the text format for all fields. You can search for key-value pairs that are included in these fields. For fields of the LONG type, you must specify the key name of a field when you query a value of the field. For fields of the other types, you do not need to specify a key name in queries.

Index type	Description
Field-specific index	<p>After you configure a field-specific index, you must specify the name of a key when you query logs. If you configure the field-specific index on a field, the field-specific index takes effect when you query logs. The full-text index does not take effect.</p> <p>Available data types that you can specify for fields include:</p> <ul style="list-style-type: none"> ◦ Query text data ◦ JSON indexes ◦ Numeric (LONG and DOUBLE)

◦ Configure a full-text index.

After you configure a full-text index for a Logstore, the values of all fields in the Logstore are queried by default.

Parameter	Description	Example value
Full Text Index	If you turn on the switch, Log Service traverses the values of all fields in a log entry. If the value of one of the fields matches the keyword, the log entry is returned.	-
Case Sensitive	<p>Specifies whether queries are case-sensitive.</p> <ul style="list-style-type: none"> ▪ If you turn off the switch, queries are not case-sensitive. For example, if you search for <code>internalError</code>, you can use either <code>INTERNALERROR</code> or <code>internalerror</code> as the keyword. ▪ If you turn on the switch, queries are case-sensitive. For example, if you search for <code>internalError</code>, you can use only <code>internalError</code> as the keyword. 	-
Include Chinese	<p>Specifies whether to differentiate the Chinese content and English content.</p> <ul style="list-style-type: none"> ▪ If you turn on the switch, Log Service separates the Chinese content based on the Chinese semantics and English content based on the specified delimiters. ▪ If you turn off the switch, the content of a log entry is separated by the specified delimiters. 	-
Delimiter	<p>The delimiters that you use to separate the content of a log entry into multiple keywords.</p> <p>For example, the content of a log entry is <code>a,b;c;D-F</code>. You can specify commas (<code>,</code>), semicolons (<code>;</code>), and hyphens (<code>-</code>) as delimiters to delimit the log content. Then you can use the five letters <code>a</code>, <code>b</code>, <code>c</code>, <code>D</code>, and <code>F</code> as keywords to match the log entry.</p>	<code>, " ; = () [] { } ? @ & < > / : \ n \ t</code>

◦ Configure a field-specific index.

You can specify fields to be indexed. Field-specific indexes allow you to query log data based on the values of specific fields. This narrows down the query scope.

Parameter	Description	Example value
Key Name	<p>The name of a log field.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin: 10px 0;"> <p> Note</p> <ul style="list-style-type: none"> ▪ If you want to configure an index for fields of the tag type, such as fields that include public IP addresses or UNIX timestamps, you must set the value of the Key Name parameter in the <code>__tag__:key</code> format, for example, <code>__tag__:__receive_time__</code>. ▪ Indexes of the numeric types are unavailable for tag fields. You must select text in the Type field for all tag fields. </div>	<code>_address_</code>
Type	<p>The type of a field. Valid values:</p> <ul style="list-style-type: none"> ▪ text: The data type of the field is TEXT. ▪ long: The data type of the field is LONG. You must specify a numeric range to query log data. ▪ double: The data type of the field is DOUBLE. You must specify a numeric range to query log data. ▪ json: The data type of the field is JSON. <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin: 10px 0;"> <p> Note The Case Sensitive, Include Chinese, and Delimiter parameters are unavailable for fields of numeric types (LONG and DOUBLE).</p> </div>	-
Alias	<p>The alias of a column.</p> <p>Aliases are applied only to SQL statistics. Original names are applied when you store and query log data in Log Service. For more information, see Field aliases.</p>	<code>address</code>
Case Sensitive	<p>Specifies whether queries are case-sensitive.</p> <ul style="list-style-type: none"> ▪ If you turn off the switch, queries are not case-sensitive. For example, if you search for <code>internalError</code>, you can use either <code>INTERNALERROR</code> or <code>internalerror</code> as the keyword. ▪ If you turn on the switch, queries are case-sensitive. For example, if you search for <code>internalError</code>, you can use only <code>internalError</code> as the keyword. 	-

Parameter	Description	Example value
Delimiter	<p>The delimiters that you use to separate the content of a log entry into multiple keywords.</p> <p>For example, the content of a log entry is <code>a,b;c;D-F</code>. You can specify commas (,), semicolons (;), and hyphens (-) as delimiters to delimit the log content. Then you can use the five letters a, b, c, D, and F as keywords to match the log entry.</p>	<code>, "" ; = () [] { } ? @ & < > / : \ \n \t</code>
Include Chinese	<p>Specifies whether to differentiate the Chinese content and English content.</p> <ul style="list-style-type: none"> If you turn on the switch, Log Service separates the Chinese content based on the Chinese semantics and English content based on the specified delimiters. If you turn off the switch, the content of a log entry is separated by the specified delimiters. 	-
Enable Analytics	<p>Specifies whether to enable the analytics feature. The switch is turned on by default.</p> <p>After you turn on the switch, you can use search and analytic statements to obtain statistical results.</p>	-

6. Click **OK**.

Note

- The index configurations take effect within one minute.
- After an index is enabled or modified, the updates on the index apply only to new data that is written to Log Service.

30.4.4. Query logs

This topic describes how to query logs in a Logstore. After you enable and configure the index of the Logstore, you can query and analyze logs in a Logstore in real time.

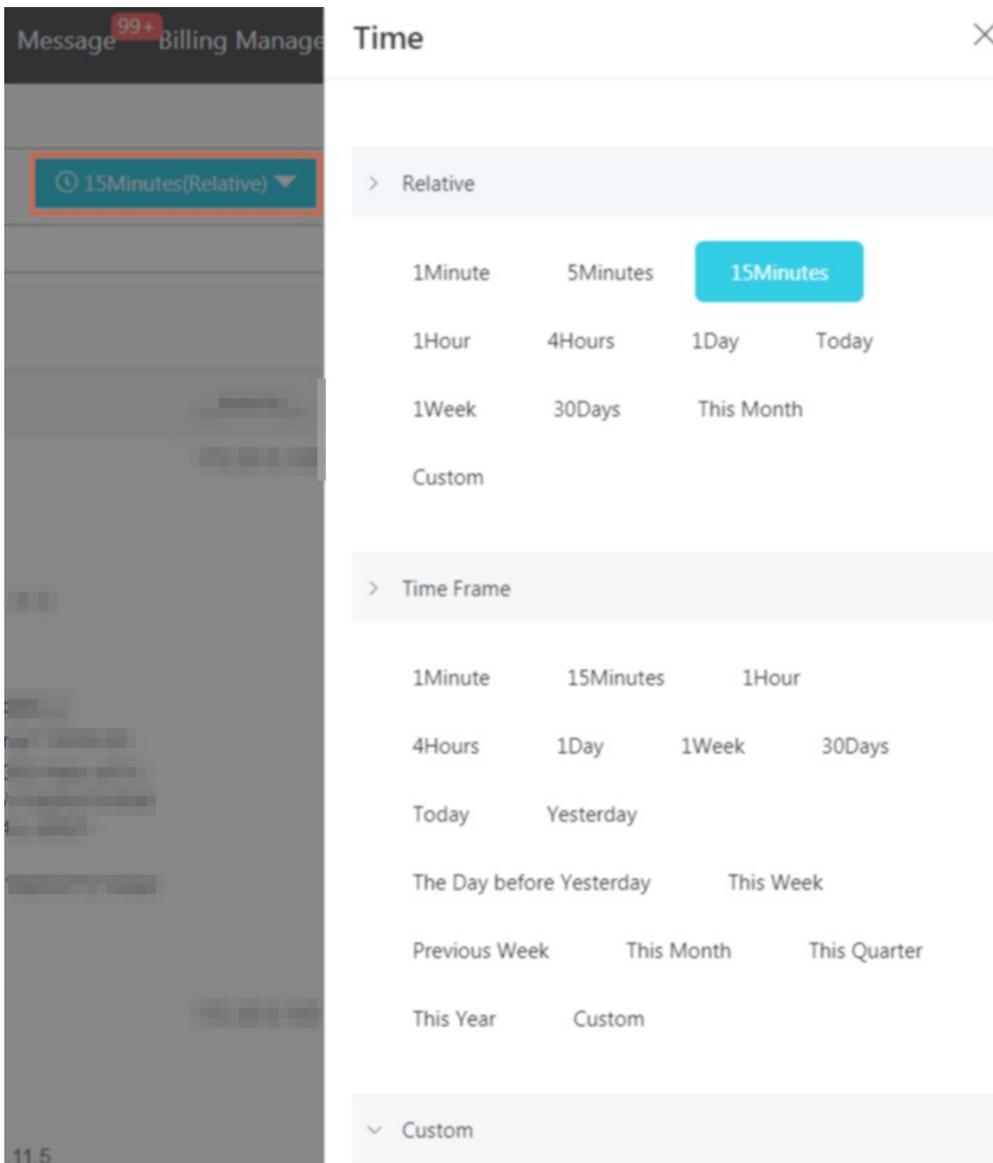
Prerequisites

- Logs are collected and stored in a Logstore.
- Indexes are enabled and configured. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Procedure

- Log on to the [Log Service console](#).
- Click the  icon next to the name of the Logstore, and then select **Search & Analysis**.
- Enter a query statement in the search box. A query statement consists of a search statement and an analytic statement. The syntax is `search statement|analytic statement`. For more information, see [Search and analysis statements](#).
- On the Search & Analysis page, select **15 Minutes** in the Relative section to set the time range for the query. You can select a relative time, time frame, or a custom time range.

Note The query results may contain logs that are generated 1 minute earlier or later than the specified time period.



5. Click **Query & Analyze** to view the query results. Log Service illustrates query results on log distribution histograms, Raw Logs tab, or statistical graphs.

Note 100 results are returned by default. For information about how to retrieve more results, see [LIMIT syntax](#).

- **Log distribution histogram**

The log distribution histogram shows the distribution of query results across different time ranges.

- Move the pointer over a green block to view a time range and the number of logs obtained within the time range.
- Click a data block to view finer-grained log distribution. You can also view the query results on the Raw Logs tab.



○ Raw Logs tab

On the Raw Logs tab, view logs that match your search conditions.

- Quick analysis: Use this feature to analyze the distribution of values for a specific field within a period of time. For more information, see [Quick analysis](#).
- Log download: Click the download icon in the upper-right corner of the tab, select a time range, and then click OK.
- Column settings: Click Column Settings in the upper-right corner of the tab, select the required fields and click Add to add the fields. Then, the columns that correspond to the fields appear on the tab. The field names are also column names. The columns list the field values.

? **Note** To view the log contents on the tab, select Content.

- Content column settings: If the content of a field exceeds 3,000 characters, extra characters will be hidden. In this case, the message "The character string is too long and has been truncated" will be displayed before the Key field. Click Display Content Column. In the dialog box that appears, set the Key-Value Arrangement and Truncate Character String parameters.

? **Note** If the content limit is set to 10,000 characters, no delimiter will be specified for extra characters.

Parameter		Description
Key-Value Pair Arrangement		You can set this parameter to New Line or Full Line.
Truncate Character String	Key	If a field value contains more than 3,000 characters, the field value is truncated. However, this parameter remains unspecified if no field value exceeds 3,000 characters. The value of this parameter is the key of the truncated value.
	Status	This parameter determines whether to enable the value truncation feature. By default, the feature is enabled. <ul style="list-style-type: none"> ■ Enable: If the value in a key-value pair exceeds the specified Truncate Step, extra characters will be truncated. You can click the Show button at the end of the value to show the truncated characters. The increment per click is the specified truncate step. ■ Disable: If the value in the key-value pair exceeds the specified Truncate Step, extra characters will not be truncated.
	Truncate Step	This parameter specifies the maximum number of characters that a field value shows by default. The parameter also specifies the number of extra characters that you displayed each time you click the Show button. Valid values: 500 to 10000. Default value: 3000.

○ Graph

If you enable the Analytics feature on the Search & Analysis page and use search and analytic statements to query logs, you can view the analytical results on the Graph tab.

- Graphs of multiple types are provided in Log Service, including tables, line charts, and bar charts. You can select a graph to show the required analytical results. For more information, see [Graphs](#).
- Log Service allows you to create dashboards for real-time data analysis. For more information, see [Create and delete a dashboard](#). Click **Add to New Dashboard** to save a common chart as query statements to a dashboard.
- Drill-down analysis allows you to move to deeper data layers, which reveals more detailed information. You can set the drill-down parameters and add the chart to the dashboard. Then, you can click the values in the chart to view the analysis results from more dimensions. For more information, see [Drill-down analysis](#).

You can also click **Save Search** or **Save as Alarm** on the Search & Analysis page to use the saved search and alarm features. For more information, see [Save a query statement as a search](#) and [Configure an alert](#).

30.4.5. Export logs

You can export logs on the current page to a CSV file and save the file to your localhost.

Procedure

1. [Log on to the Log Service console](#).
2. Click a project name.
3. Click the  icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Click the  icon next to the **Raw Logs** tab.
5. In the **Download Log** dialog box, select **Download Log in Current Page**.
6. Click **OK** to export logs of the current page to a .CSV file and save the file to the localhost.

30.4.6. Index data type

30.4.6.1. Overview

Log Service allows you to use full-text indexes or field-specific indexes to query collected logs. If you set a full-text index for a log, the value is the entire log. If you set a field-specific index for a log, you can specify a data type for each key.

Date types

The following table lists the supported data types.

Query type	Data type (index)	Description	Example
Basic query	Text	The text type. You can use keywords and fuzzy matches to query logs.	<code>uri:"login*" method:"post"</code>
	Long	The numeric type. You can specify numeric ranges to query logs.	<code>status>200 and status in [200, 500]</code>
	Double	The floating-point type.	<code>price>28.95 and t in [20.0, 37]</code>

Query type	Data type (index)	Description	Example
Combined query	JSON	Indicates that the index is a JSON field that supports nested queries. By default, the data type of the field is text. You can set indexes of the Text, Long, and Double types for the b elements at layer a in the a.b path format. The fields adopt the configured types.	<pre>level0.key>29.95 level0.key2:"action"</pre>
	Text	Indicates that the full contents of the log are queried as text.	<pre>error and "login fail"</pre>

Query examples

The following table lists the keys included in the sample log.

No.	Key	Type
0	time	N/A
1	class	text
2	status	long
3	latency	double
4	message	json

```

0. time:2018-01-01 12:00:00
1. class:central-log
2. status:200
3. latency:68.75
4. message:
{
  "methodName": "getProjectInfo",
  "success": true,
  "remoteAddress": "1.1.1.1:11111",
  "usedTime": 48,
  "param": {
    "projectId": "ali-log-test-project",
    "requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"
  },
  "result": {
    "message": "successful",
    "code": "200",
    "data": {
      "clusterRegion": "ap-southeast-1",
      "ProjectName": "ali-log-test-project",
      "CreateTime": "2017-06-08 20:22:41"
    },
    "success": true
  }
}

```

You can set an index as follows.

Set an index

Key Name	Type	Alias	Case Sensitive	Delimiter: ?	Include Chinese	Enable Analytics	Delete
class	text		<input type="checkbox"/>	, ";=000?@&<>/\n\t	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
message	1 json		<input type="checkbox"/>	, ";=000?@&<>/\n\t	<input type="checkbox"/>	<input type="checkbox"/>	×
methodName	text		<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
param.requestId	text		<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
result.data.clusterRegion	text		<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
usedTime	2 long		<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	×

In the preceding figure,

- ① specifies that Log Service can query data of the string and Boolean types in JSON fields.
- ② specifies that Log Service can query data of the long type.
- ③ enables SQL analysis for specified fields.

Example

1. Query data of the string and Boolean types

- You do not need to configure JSON fields.
- JSON maps and arrays are automatically expanded and can contain nested fields. Separate multiple levels with periods (.).

```
class : cental*
message.traceInfo.requestId : 92.137_1518139699935_5599
message.param.projectName : ali-log-test-project
message.success : true
```

2. Query data of the double and long types

Each JSON field must be specified separately and cannot be contained in a JSON array.

```
latency>40
message.usedTime > 40
```

3. Combined query

```
class : cental* and message.usedTime > 40 not message.param.projectName:ali-log-test-project
```

30.4.6.2. Query text data

This topic describes how to query text data.

Similar to search engines, Log Service queries text data based on terms. Therefore, you must set the Delimiter, Case Sensitive fields.

Configurations

- Case sensitivity

You can specify whether log queries are case-sensitive. For example, you want to query logs by using a search term named `internalError`.

- *false* specifies a case-insensitive query. Both `INTERNALERROR` and `internalerror` can be the keywords.
- *true* specifies a case-sensitive query. Only the `internalError` can be the keyword.

- Delimiter

You can use delimiters to split a search term into multiple keywords.

For example, you want to query logs by using the following search term.

```
/url/pic/abc.gif
```

- If no delimiter is set, the entire `/url/pic/abc.gif` string is treated as a keyword. You must use the entire string as a keyword or a fuzzy string named `/url/pic/*` to query logs.
- If the delimiter is set to `/`, the search term is split into three words: `url`, `pic`, and `abc.gif`. You can use one of these words or a fuzzy word to query logs. For example, `url`, `abc.gif`, or `pi*`. You can also use the `/url/pic/abc.gif` string as a search term to query logs. However, the search term is split into three keywords named `url`, `pic`, and `abc.gif`.
- If the delimiter is set to `/.`, the search term is split into four keywords named `url`, `pic`, `abc`, and `gif`.

 **Note** You can extend query ranges by setting appropriate delimiters.

- Full-text index

By default, full-text indexes treat each log except for the time field as text data. You do not need to specify any keys for a full-text index. For example, the following log includes the time field, status field, level field, and message field.

```
[20180102 12:00:00] 200,error,some thing is error in this field
```

- `time:2018-01-02 12:00:00`
- `level:"error"`
- `status:200`
- `message:" some thing is error in this field"`

Note

- Prefixes are not required for full-text indexes. If you set the search term to `error`, the level and message fields that include error match the search term.
- You must set delimiters for full-text indexes. For example, if you set a space () as a delimiter, the `status:200` string is a search term. If you set a colon (:) as a delimiter, the search term is split into two keywords named `status` and `200` .
- Numbers are treated as text data. For example, you can use `200` to query logs. Values in the time field are not treated as text data.
- You can query logs by using keys such as `status` .

30.4.6.3. Numeric type

When you configure indexes, you can set the data type of a key to number. To query logs, you can specify a numeric range for the key.

Configurations

Supported types: `long` (long integers) and `double` (decimals). After you set the data type of a key to number, you must specify a numeric range for the key to query logs.

Query examples

To specify a numeric range from 1000 to 2000 (excluding 1000) for a key of the long type, you can use the following methods:

- Query syntax for numbers. For example:

```
longKey > 1000 and longKey <= 2000
```

- Query grammar for numeric ranges. For example:

```
longKey in (1000 2000]
```

For more information about query syntax, see [Query syntax](#).

30.4.6.4. JSON indexes

Log Service can query and analyze logs in the JSON format. You can set the data type of indexes to JSON.

JSON texts include data of multiple types, including string, Boolean, number, array, and map. JSON-formatted data is self-parsed and flexible. You can use JSON-formatted data in various scenarios. In most cases, variable log fields are recorded in the JSON format. For example, HTTP request and response parameters are recorded in a log in the JSON format.

Log Service allows you to set the data type of index fields to JSON so that you can query and analyze logs in the JSON format.

Configurations

- Log Service can parse JSON-formatted fields and generate indexes for all the fields of the text and Boolean types.

```
json_string.key_map.key_text : test_value
json_string.key_map.key_bool : true
```

- To query fields of the double or long type that is not in a JSON array, you can specify a JSON path.

```
Set the data type of the key_map.key_long field to long.
Search condition: json_string.key_map.key_long > 50
```

- To query fields of the text, double, or long type that is not in a JSON array, you can enable the Analytics feature and use SQL statements to analyze these fields.

```
json_string.key_map.key_long > 10 | select count(*) as c ,
"json_string.key_map.key_text" group by
"json_string.key_map.key_text"
```

Note

- JSON objects and JSON arrays are not supported.
- Fields cannot be contained in JSON arrays.
- Fields of the Boolean type can be converted into the text type.
- To query and analyze logs, JSON-formatted fields must be enclosed with double quotation marks (" ").

- Log Service cannot parse invalid JSON-formatted data.

Log Service does not stop parsing logs until it detects an invalid field.

In the following example, data after the key_3 field is truncated and lost in the following text. Log Service can parse the json_string.key_map.key_2 field and the contents before this field.

```
"json_string":
{
  "key_1" : "value_1",
  "key_map" :
  {
    "key_2" : "value_2",
    "key_3" : "valu
```

Query syntax

To query a specific key, you must add the JSON parent path to the query statement as the prefix of the key. The query syntax for the fields of the text and numeric types is the same for both JSON-formatted data and other data. For more information, see [Query syntax](#).

Query example

The following table lists the keys included in the sample log. The data type of the `message` key is JSON.

Number	Key	Type
0	time	N/A
1	class	text
2	status	long
3	latency	double
4	message	json

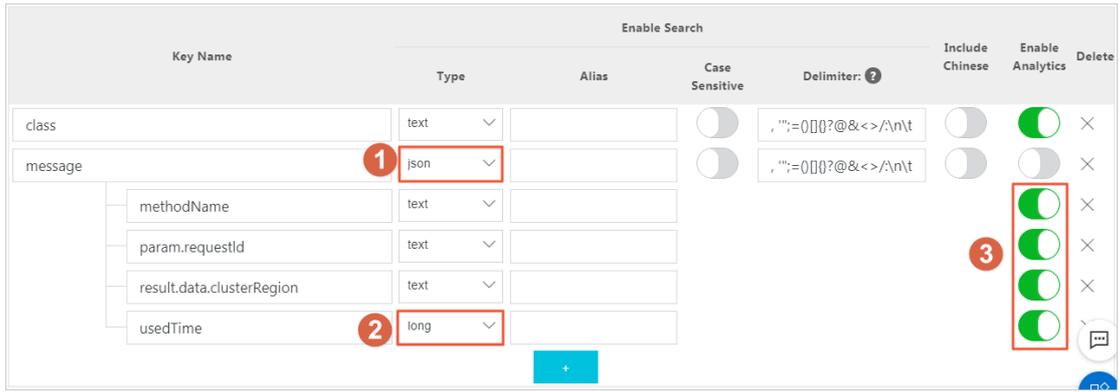
```

0. time:2018-01-01 12:00:00
1. class:central-log
2. status:200
3. latency:68.75
4. message:
{
  "methodName": "getProjectInfo",
  "success": true,
  "remoteAddress": "1.1.1.1:11111",
  "usedTime": 48,
  "param": {
    "projectId": "d3f0c96a-51b0-4166-a850-f4175dde7323",
    "projectName": "ali-log-test-project",
    "requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"
  },
  "result": {
    "message": "successful",
    "code": "200",
    "data": {
      "clusterRegion": "ap-southeast-1",
      "ProjectName": "ali-log-test-project",
      "CreateTime": "2017-06-08 20:22:41"
    },
    "success": true
  }
}

```

You can set indexes for the log as follows:

Set an index



In the preceding figure:

- ① specifies that Log Service can query data of the string and Boolean types in JSON fields.
- ② specifies that Log Service can query data of the long type.
- ③ enables SQL analysis for specified fields.

Examples

1. Query data of the string and Boolean types

Note

- You do not need to configure JSON fields.
- JSON maps and arrays are automatically expanded and can include hierarchical levels. Separate multiple levels with periods (.).

```
message.traceInfo.requestId : 92.137_1518139699935_5599
message.param.projectName : ali-log-test-project
message.success : true
message.result.data.ProjectStatus : Normal
```

2. Query fields of the double and long types

Note Each JSON field must be configured and cannot be contained in an array.

```
message.usedTime > 40
```

3. Use SQL statements to analyze fields

Note

- Each JSON field must be configured and cannot be contained in an array.
- Each field to be queried must be enclosed with double quotation marks (" ") or be configured with an alias.

```
* | select avg("message.usedTime") as avg_time ,
"message.methodName" group by "message.methodName"
```

30.4.7. Query syntax and functions

30.4.7.1. Search syntax

This topic describes the search syntax that is used in Log Service.

Search types

After you [enable and configure the index feature](#) of a Logstore, you can enter a search statement on the search and analysis page to [query logs](#).

A query statement consists of two sub-statements in sequence: a search statement and an analytic statement. A search statement specifies one or more search conditions and returns the log entries that match the search conditions. You can execute a search statement to perform a full-text search or field-specific search.

- Full-text search

During full-text search, a log entry is considered a key-value pair. The value in the key-value pair indicates the content of the log entry. A full-text search statement returns the log entries that include or exclude the specified keywords.

Full-text search is divided into basic full-text search, phrase search, and wildcard-based search.

- Basic full-text search: You can specify keywords and operators in search conditions of a search statement. You can then execute the search statement to query the log entries that match the search conditions.

For example, the `a and b` statement returns the log entries that include the `a` and `b` keywords.

- Phrase search: A phrase is a string that is enclosed in double quotation marks (""). Substrings in a phrase are separated by space characters. Each substring is a keyword.

For example, the `"http error"` statement returns the log entries that contain the `http` and `error` keywords. This statement is equivalent to `http and error`.

- Wildcard-based search: You can use an asterisk (`*`) or a question mark (`?`) as a wildcard character in a keyword. Each keyword that includes wildcards can contain 1 to 64 characters in length and cannot start with a wildcard character. If a search condition contains a keyword that includes a wildcard character, Log Service returns a maximum of 100 log entries and each log entry contains a word that matches the keyword pattern.

For example, if you execute the `addr?` statement, Log Service returns a maximum of 100 log entries and each log entry contains a word that is prefixed with `addr`.

When you use wildcard-based search, note the following information:

- A keyword cannot start with an asterisk (`*`) or a question mark (`?`).
- The more accurate the keyword is, the more accurate the search results will be.
- Wildcard-based search is not supported for a keyword that contains more than 64 characters in length.
- A search statement returns a maximum of 100 log entries that match the search conditions.

- Field-specific search

After you configure the field index, you can search log entries based on the keys and values of the fields in the field index. For a field of the `DOUBLE` or `LONG` type, you can specify a value range for search. For example, the `Latency>5000 and Method:Get* and not Status:200` statement returns the log entries that meet the following conditions: The value of the `Latency` field is greater than 5000, the value of the `Method` field is prefixed with `Get`, and the value of the `Status` field is not 200.

You can perform a basic query or combined query, depending on the data types of the fields in the field index. For more information, see [Overview](#).

Additional considerations

- If you execute a search statement to perform both full-text search and field-specific search and you set different delimiters for the two search types, the delimiter that is set for field-specific search is used.
- You must set the data type of a field to `DOUBLE` or `LONG` before you specify a value range to search the field. If the data type of a field is not `DOUBLE` or `LONG` or the value range syntax is incorrect, the field-specific

search condition is considered a full-text search condition. In this case, unexpected search results may be returned.

- If you change the data type of a field from TEXT to DOUBLE or LONG, only the equal-to operator (=) can be used to search for the log entries that are collected before the change.

Operators

The following table lists the operators that are supported by search statements.

Operator	Description
and	A binary operator. The syntax is <code>query1 and query2</code> . It indicates the intersection of the search results of <code>query1</code> and <code>query2</code> . The default operator between keywords is <code>and</code> .
or	A binary operator. The syntax is <code>query1 or query2</code> . It indicates the union of the search results of <code>query1</code> and <code>query2</code> .
not	A binary operator. The syntax is <code>query1 not query2</code> . It indicates that the log entries that match <code>query1</code> but do not match <code>query2</code> are returned. The syntax is equivalent to <code>query1-query2</code> . You can also use the <code>not query1</code> syntax. It indicates that the log entries that do not match <code>query1</code> are returned.
(,)	The operator that merges one or more sub-conditions into one search condition. The search based on a sub-condition that is enclosed in parentheses () is performed first.
:	The operator that is used to specify a pattern of key-value pairs. The syntax is <code>term1:term2</code> . If the key or value contains reserved characters such as spaces and colons (:), use double quotation marks ("") to enclose the entire key or value.
"	The operator that converts another operator into a common character. All terms enclosed in double quotation marks ("") are considered keywords rather than operators. In a field-specific search statement, you can enclose the entire key or value in double quotation marks.
\	The operator that escapes a double quotation mark. The escaped double quotation mark is considered a symbol instead of an operator. Example: <code>"\"</code> .
	The pipeline operator that is used to chain a search statement and an analytic statement. The analytic statement that follows the pipeline operator is executed based on the result of the search statement that the pipeline operator follows. Example: <code>query1 select count(1)</code> .
count	The count operator that is used to summarize the number of log entries.
*	The wildcard character that is used to replace zero or more characters. For example, the <code>que*</code> statement returns the log entries with a word that is prefixed with <code>que</code> . <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note A wildcard-based search statement returns a maximum of 100 log entries that match the search condition.</p> </div>

Operator	Description
?	The wildcard character that replaces a single character. The <code>qu?ry</code> statement returns the log entries with a word that is prefixed with <code>qu</code> , is suffixed with <code>ry</code> , and contains a character in between.
<code>__topic__</code>	The operator that specifies zero or more topics from which to query log entries. Example: <code>__topic__:mytopicname</code> .
<code>__tag__</code>	The operator that specifies a tag value of a tag key to query. Example: <code>__tag__:tagkey:tagvalue</code> .
<code>source</code>	The operator that specifies the IP address of a log source whose log entries you want to query. Example: <code>source:127.0.0.1</code> .
>	The greater-than operator. You can use this operator to query the log entries whose value of a field is greater than a specified number. Example: <code>latency > 100</code> .
>=	The greater-than-or-equal-to operator. You can use this operator to query the log entries whose value of a field is greater than or equal to a specified number. Example: <code>latency >= 100</code> .
<	The less-than operator. You can use this operator to query the log entries whose value of a field is less than a specified number. Example: <code>latency < 100</code> .
<=	The less-than-or-equal-to operator. You can use this operator to query the log entries whose value of a field is less than or equal to a specified number. Example: <code>latency <= 100</code> .
=	The equal-to operator. You can use this operator to query the log entries whose value of a field is equal to a specified number. Example: <code>latency = 100</code> .
<code>in</code>	The operator that is used to query the log entries whose value of a field falls in a specified range. Brackets <code>[]</code> indicate closed intervals and parentheses <code>()</code> indicate open intervals. The beginning number and ending number of the range are enclosed in brackets or parentheses and separated by one or more space characters. The <code>in</code> operator must be in lowercase. Example: <code>latency in [100 200]</code> or <code>latency in (100 200)</code> .

 Note

- All operators except the `in` operator are case-insensitive.
- You can use the following operators, which are sorted in descending order of precedence: `:`, `"`, `()`, `and`, `not`, and `or`.
- Log Service uses the following operators: `sort`, `asc`, `desc`, `group by`, `avg`, `sum`, `min`, `max`, and `limit`. If you need to use these operators as keywords, enclose them in double quotation marks (`"`).

Search statement examples

Expected search result	Search statement
Log entries that contain a and b	<code>a and b</code> or <code>a b</code>

Expected search result	Search statement
Log entries that contain a or b	a or b
Log entries that contain a but do not contain b	a not b
Log entries that do not contain a	not a
Log entries that contain a and b but do not contain c	a and b not c
Log entries that contain a or b and contain c	(a or b) and c
Log entries that contain a or b but do not contain c	(a or b) not c
Log entries that contain a and b and may contain c	a and b or c
Log entries whose FILE field contains apsara	FILE:apsara
Log entries whose FILE field contains apsara and shennong	FILE:"apsara shennong" , FILE:apsara FILE:shennong , or FILE:apsara and FILE:shennong
Log entries that contain the following keyword: and	and
Log entries whose FILE field contains apsara or shennong	FILE:apsara or FILE:shennong
Log entries whose file info field contains apsara	"file info":apsara
Log entries that contain double quotation mark (")	\"
Log entries with words that are prefixed with shen	shen*
Log entries whose FILE field is prefixed with shen	FILE:shen*
Log entries whose value of the FILE field is shen*	FILE: "shen**"
Log entries with words that are prefixed shen, are suffixed with ong, and contain a single character in between	shen?ong
Log entries with words that are prefixed with shen and words that are prefixed with aps	shen* and aps*
Log entries of topic1 and topic2	__topic__:topic1 or __topic__ : topic2
Log entries with a tag whose key is tagkey1 and value is tagvalue2	__tag__ : tagkey1 : tagvalue2
Log entries whose value of the latency field is greater than or equal to 100 and less than 200	latency>= 100 and latency < 200 or latency in [100 200)
Log entries whose value of the latency field is greater than 100	latency > 100

Expected search result	Search statement
Log entries that do not contain spider and whose http_referer field does not contain opx	<code>not spider not bot not http_referer:opx</code>
Log entries whose cdnIP field is not empty	<code>not cdnIP:""</code>
Log entries that do not contain the cdnIP field	<code>not cdnIP:*</code>
Log entries that contain the cdnIP field	<code>cdnIP:*</code>
Log entries that contain a specified URL	<code>* select * where url = 'www.xxxxx.com'</code>

Topic-specific search

Each Logstore is divided into one or more topics. You can divide a Logstore into multiple topics if you need level-2 categories of log entries. When you query logs, you can specify topics to increase efficiency.

In a search statement, you can specify one or more topics to query. If no topic is specified, log entries are queried from all topics.

For example, you can classify log entries into multiple topics based on domain names.

Log topics

time	ip	method	url	host	topic
1481270421	127.0.0.1	POST	/users?u=1	a.aliyun.com	siteA
1481270422	127.0.0.1	POST	/users?u=1	a.aliyun.com	siteA
1481270423	127.0.0.1	POST	/users?u=1	b.aliyun.com	siteB
1481270424	127.0.0.1	POST	/users?u=1	b.aliyun.com	siteB
1481270425	127.0.0.1	POST	/users?u=1	c.aliyun.com	siteC
1481270426	127.0.0.1	POST	/users?u=1	c.aliyun.com	siteC
1481270427	127.0.0.1	POST	/users?u=1	d.aliyun.com	siteD
1481270428	127.0.0.1	POST	/users?u=1	d.aliyun.com	siteD
1481270429	127.0.0.1	POST	/users?u=1	e.aliyun.com	siteE
1481270430	127.0.0.1	POST	/users?u=1	e.aliyun.com	siteE

Syntax of topic-specific search:

- In a search statement, you can specify one or more topics to query. If no topic is specified, log entries are queried from all topics.
- The topic-specific search syntax is `__topic__:topicName`. You can also specify a topic in a URL.
- You can query log entries from multiple topics. For example, the `__topic__:topic1 or __topic__:topic2` statement returns the log entries in topic1 and topic2.

30.4.7.2. LiveTail

This topic describes how to use LiveTail to monitor and analyze log data. LiveTail is an interactive feature provided in the Log Service console to monitor and extract key log data in real time.

Prerequisites

- LiveTail is available only after logs are collected.
- LiveTail can only monitor and extract log data collected by Logtail.

Context

In online O&M scenarios, you often need to monitor collected log data in real time and extract key information from the latest log data to locate error causes. In traditional O&M, you must run the `tail -f` command on servers to monitor log files in real time. To easily obtain the required real-time log information, you can include the `grep` or `grep -v` command to filter log entries by keyword. To simplify online O&M, Log Service provides LiveTail in the console to monitor and analyze online log data in real time.

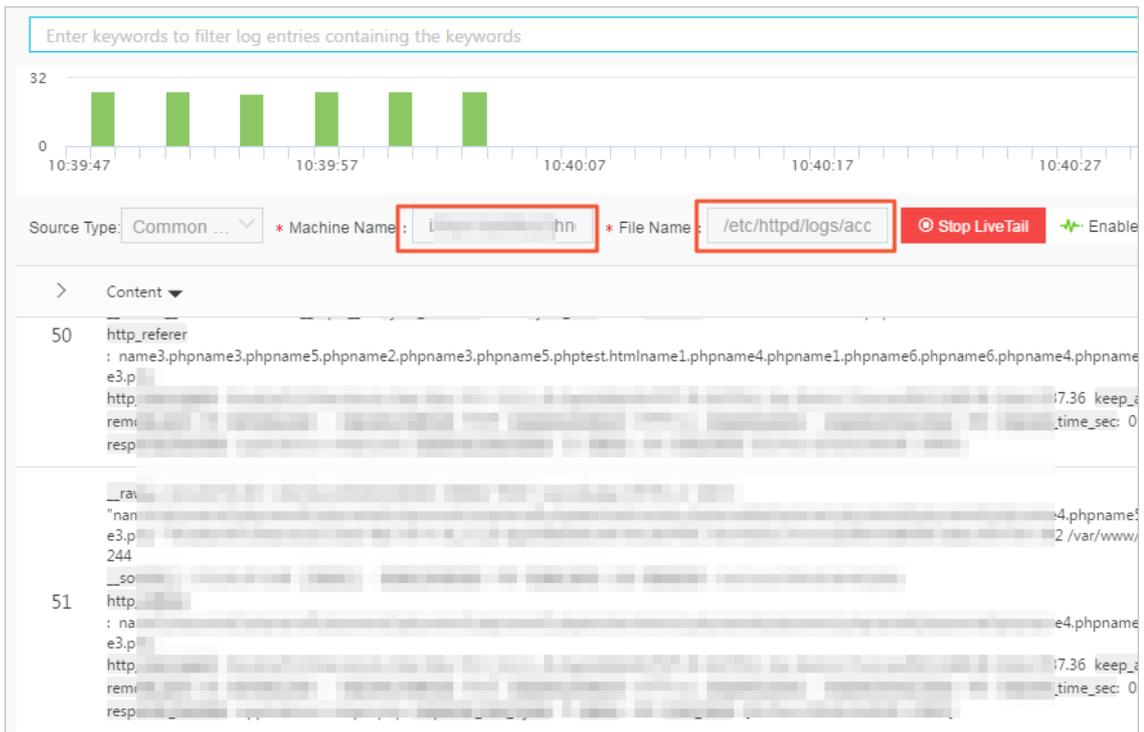
Benefits

- Monitors real-time log information and filters log data by keyword.
- Logs collected based on the log collection configurations are identified by index.
- Log fields are delimited. This allows you to query contextual logs that contain delimiters.
- Log files can be tracked based on a single log entry and monitored in real time without the need to connect to online servers.

Use LiveTail to monitor logs in a Logstore in real time

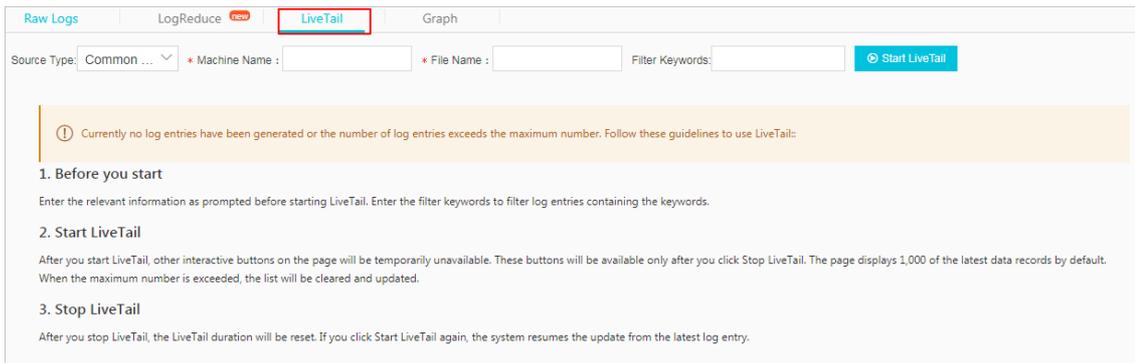
1. Log on to the Log Service console.
2. In the Projects section, click the target project.
3. Click the  icon next to the name of the Logstore, and then select Search & Analysis.
4. (Optional)Start LiveTail.
 - i. On the Raw Logs tab, click the  icon next to the sequence number of the specified raw log entry, and then select LiveTail. The system starts LiveTail and starts timing. The Source Type, Machine Name, and File Name fields are automatically filled in based on the specified raw log entry.

After LiveTail is started, log data collected by Logtail is displayed in order on the page in real time. The latest log data is displayed at the bottom of the page by default. You can view the latest log data without the need to drag the scroll bar. Up to 1,000 log entries can be displayed on the page. If more than 1,000 log entries are collected, the page is automatically refreshed to show the latest 1,000 log entries.



- ii. (Optional)You can also enter keywords in the search box to display log entries that contain the keywords in the monitoring list. By filtering log entries that contain the keyword, you can monitor specific log entries in real time.
5. Customize LiveTail.

i. On the Search & Analysis page, click the LiveTail tab.



ii. Configure LiveTail.

Parameter	Required	Description
Source Type	Yes	The source of log entries. Valid values: <ul style="list-style-type: none"> ■ Physical servers ■ Kubernetes containers ■ Docker
Machine Name	Yes	The name of the server from which log entries are collected.
File Name	Yes	The full path and name of the log file.
Filter Keywords	No	A keyword. After you set a keyword, only log entries that contain the keyword are displayed in the log monitoring list.

iii. Click **Start LiveTail**. After LiveTail is started, log data collected by Logtail is displayed in order on the page in real time. The latest log data is displayed at the bottom of the page by default. You can view the latest log data without the need to drag the scroll bar. Up to 1,000 log entries can be displayed on the page. When more than 1,000 log entries are collected, the page is refreshed to show the latest 1,000 log entries.

6. To analyze logs during real-time log monitoring, click **Stop LiveTail**.

After you stop LiveTail, the LiveTail timing and the real-time log data update also stop.

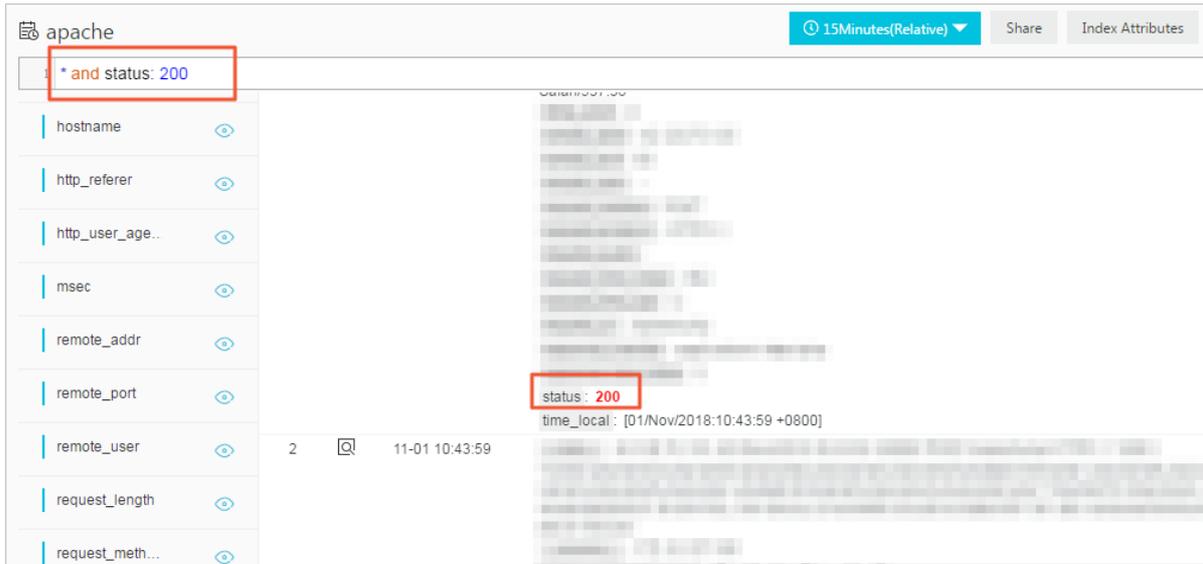
Log Service provides multiple methods to analyze exceptions that are found during log monitoring. For more information, see [Use LiveTail to analyze logs](#).

Use LiveTail to analyze logs

After you stop LiveTail, real-time log updates in the log monitoring list also stop. You can analyze and fix errors that are found during log monitoring.

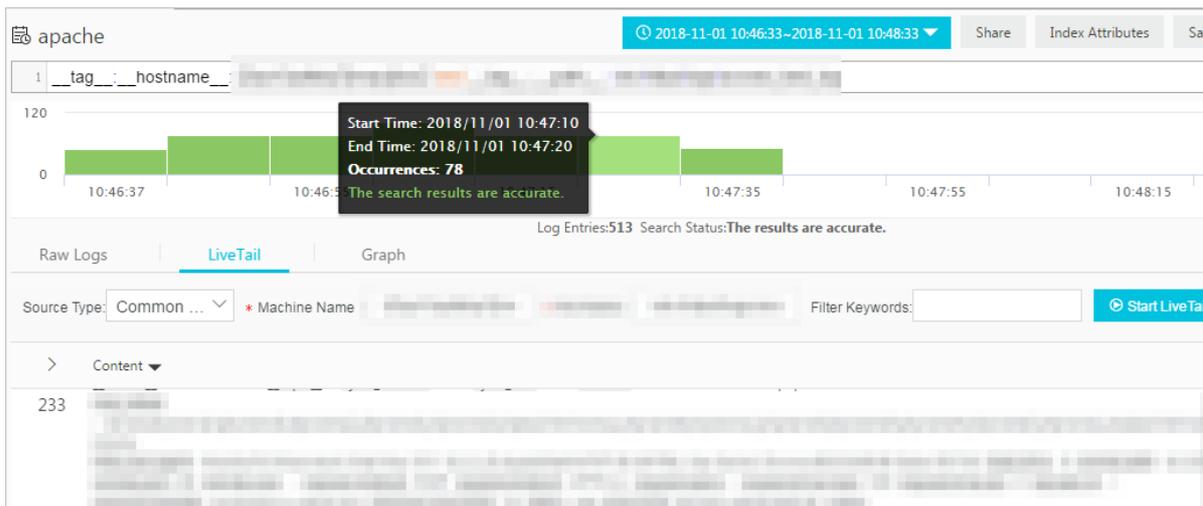
- View log entries that contain specified fields

Log fields are delimited. You can click the value of the specified exception field on the LiveTail tab. Then, you are automatically forwarded to the Raw logs tab and the value of the exception field is used as a keyword to filter all log entries that contain the field and the keyword. You can also analyze log entries that contain the keyword based on contextual queries and statistical charts.



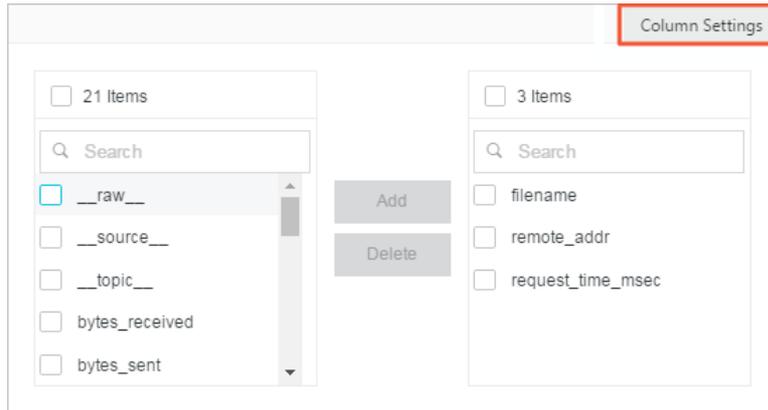
- Narrow down the time range of a log query based on the log distribution histogram

After LiveTail is started, the log distribution histogram is updated at the same time. If you find a distribution exception (for example, a big increase in the number of log entries) in a time range, you can click the corresponding green rectangle to narrow down the time range of the log query. The timeline of the raw logs is associated with the timeline that you click on the LiveTail tab. You can view all relevant raw logs and log distribution details during this time range.



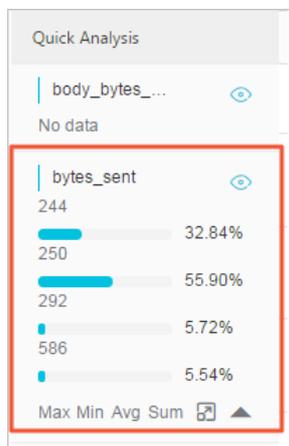
- Highlight key information based on column settings

On the LiveTail tab, click Column Settings in the upper-right corner of the log monitoring list. Then, you can specify a field as a separate column to highlight relevant data.



- Analyze log data

On the **LiveTail** tab, click the arrow in the upper-left corner of the log monitoring list to show the Quick Analysis pane. The time range for quick analysis starts from the time when LiveTail is started and ends at the time when LiveTail is stopped. The quick analysis provided on the LiveTail tab is the same as that provided on the Raw Logs tab. For more information, see [Quick analysis](#).



30.4.7.3. LogReduce

This topic describes how to use LogReduce to group log data in Log Service. The LogReduce feature groups similar log entries by detecting same log patterns during text log collection.

Context

The LogReduce feature allows you to group text logs of multiple formats. You can locate errors, detect anomalies, roll back versions, and perform other O&M operations in DevOps scenarios. You can also detect network intrusions to ensure data security. In addition, you can save the log grouping result to a dashboard as an analysis chart, and then view the grouped data in real time.

Benefits

- Various formats of logs such as Log4j logs, JSON-formatted logs, and syslog logs can be grouped.
- Hundreds of millions of log entries can be grouped in seconds.
- Log entries can be grouped in any pattern.
- Raw log entries can be retrieved based on the signature of log entries grouped in a pattern.
- The number of log entries grouped in a log pattern in different time ranges can be compared.
- The precision of log grouping can be adjusted based on your needs.

Billing method

After the LogReduce feature is enabled, the size of indexes increases by 10% of the raw log size. For example, if the size of raw log data is 100 GB per day, the size of log indexes increases by 10 GB.

Raw log size	Index percentage	Size of indexes generated by LogReduce	Index size
100 GB	20% (20 GB)	100 × 10%	30 GB
100 GB	40% (40 GB)	100 × 10%	50 GB
100 GB	100% (100 GB)	100 × 10%	110 GB

Enable LogReduce of a Logstore

The LogReduce feature is disabled by default.

1. [Log on to the Log Service console.](#)
2. Click the target project in the Projects section.
3. Click the  icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Configure an index.
 - o If you have enabled the index feature and configured indexes for the Logstore, choose **Index Attributes > Modify**.
 - o If you have not enabled the index feature, click **Enable**.
5. Set the index attributes and turn on the **LogReduce** switch. Click **OK**. After LogReduce is enabled, Log Service groups log data that has been collected. Then, you can perform the following operations:
 - o [View log grouping results and raw logs](#)
 - o [Adjust the precision of log grouping](#)
 - o [Compare the number of log entries grouped in different time ranges](#)

View log grouping results and raw logs

1. On the Search & Analysis page, enter a search and analytic statement in the search box, and then click **Search & Analytics**. You can use keywords to filter grouped log entries.

 **Note** SQL statements are not supported. This means analysis results cannot be grouped.

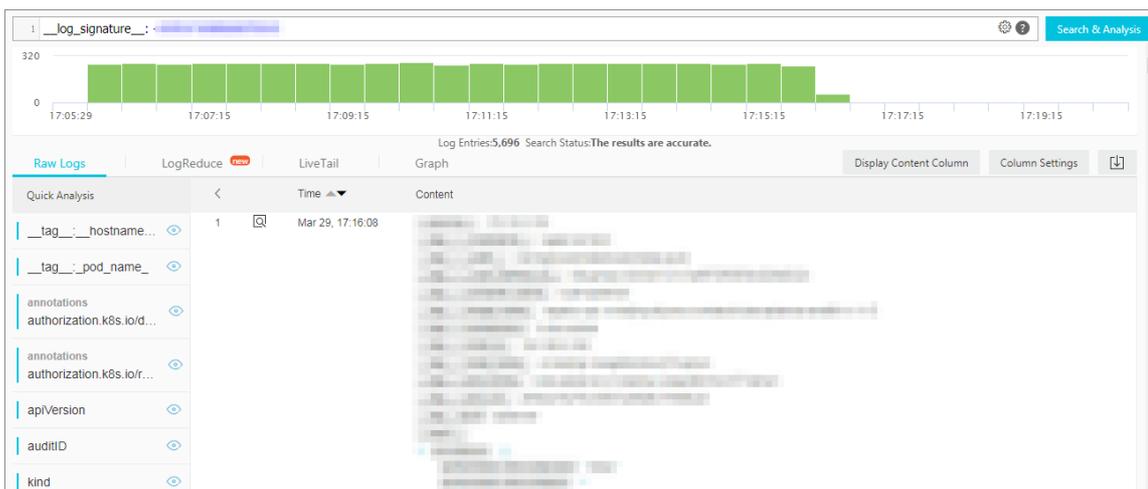
2. Click the **LogReduce** tab to view the log grouping result. The filtered log grouping result is displayed on the **LogReduce** tab.

Item	Description
Number	The sequence number of a log group.
Count	The number of log entries in a log group.
Pattern	The log pattern. Each log group has one or more sub-patterns.

- o Move the pointer over a value in the **Count** column to view the sub-patterns of the corresponding log group and the percentage of each sub-pattern in the log group. You can also click the plus sign (+) before the count value to expand the sub-pattern list.

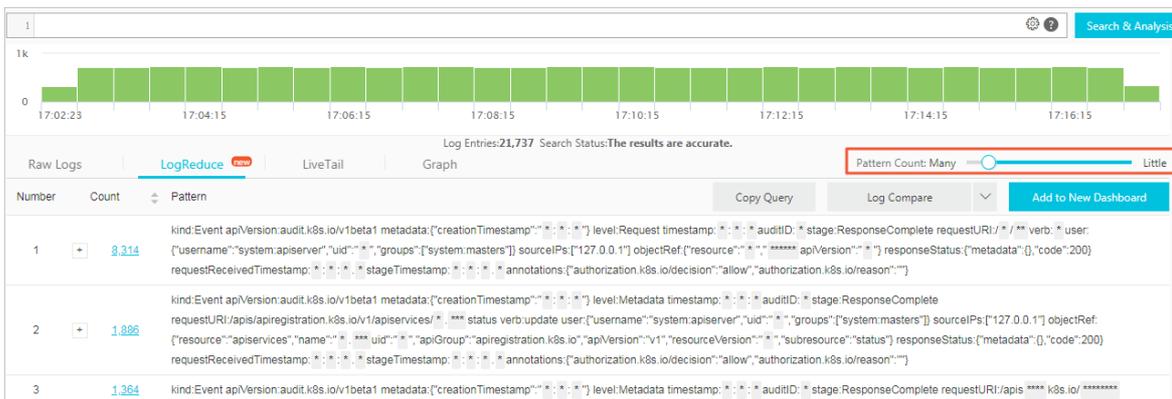


Click a count value to view the raw log entries of the corresponding log group.



Adjust the precision of log grouping

- On the Search & Analysis page, click the LogReduce tab.
- In the upper-right corner of the tab, drag the Pattern Count slider to adjust the precision of log grouping.
 - If you drag the slider towards Many, you can obtain a more precise log grouping result with more detailed patterns.
 - If you drag the slider towards Little, you can obtain a less precise log grouping result with less detailed patterns.



Compare the number of log entries grouped in different time ranges

Click Log Compare on the LogReduce tab, select a time range, and then click OK.

The screenshot shows the Log Service console interface. At the top, there is a SQL query editor with the following query: `select v.signature, v.pattern, v.count, v.count_compare, v.diff from (select compare_log_reduce(3, 86400) as v from log) order by v.diff desc limit 1000 log_reduce(precision)`. Below the query editor is a graph area. The main part of the interface is a table with columns: Number, Pre_Count, Count, Diff, and Pattern. The table contains four rows of log group data. A 'Log Compare' dropdown menu is open over the table, showing options for time ranges: 5Minutes, 15Minutes, 1Hour, 4Hours, 1Day, 1Week, and 30Days. The '1Day' option is selected. There are also input fields for 'Start Time' (2019-04-10 10:17:15) and 'End Time' (2019-04-10 10:32:15), and an 'OK' button.

Item	Description
Number	The sequence number of a log group.
Pre_Count	The number of log entries grouped by the current pattern in the previous time range.
Count	The number of log entries grouped by the current pattern in the current time range.
Diff	The difference between the Pre_Count value and Count value.
Pattern	The pattern of a log group.

SQL statement examples:

- Obtain a log grouping result.

- SQL statement:

```
* | select a.pattern, a.count, a.signature, a.origin_signatures from (select log_reduce(3) as a from log) limit 1000
```

Note When you view the log grouping result in the Log Service console, you can click Copy Query to obtain the relevant SQL statement.

- Input parameter: log_reduce (precision)

precision: an integer from 1 to 16 that can be set as the log grouping precision. A smaller value indicates a higher precision and more patterns. The default value is 3.

- Returned fields:

- pattern: the sub-patterns of log entries in a log group.
- count: the number of log entries in a log group.
- signature: the log pattern of a log group.
- origin_signatures: the original signature of a log group. You can use this field to query raw log entries of the log group.

- Compare the number of log entries grouped in different time ranges.

- SQL statement:

```
* | select v.pattern, v.signature, v.count, v.count_compare, v.diff from (select compare_log_reduce(3, 86400) as v from log) order by v.diff desc limit 1000
```

 **Note** After you click **Log Compare** in the Log Service console, you can click **Copy Query** to obtain the SQL statement.

- Input parameters: `compare_log_reduce(precision, compare_interval)`
 - **precision**: an integer from 1 to 16 that can be set as the log grouping precision. A smaller value indicates a higher precision and more patterns. The default value is 3.
 - **compare_interval**: the number of seconds between when the previous log entries and the current log entries were generated. The value of this parameter must be a positive integer.
- Returned fields:
 - **pattern**: the sub-patterns of log entries in a log group.
 - **signature**: the log pattern of a log group.
 - **count**: the number of log entries in a log group.
 - **count_compare**: the number of log entries for a same-pattern log group within the specified time range.
 - **Diff**: the difference between the values of the count field and the count_compare field.

30.4.7.4. Contextual query

This topic describes the contextual query feature provided in the Log Service console. You can use this feature to query the full context of the log file where specified log entries are obtained.

Prerequisites

The index feature is enabled.

Context

The contextual query feature identifies the server and file where a specified log entry resides. It then queries several log entries before and after the log entry in the original log file. This helps you locate errors during troubleshooting.

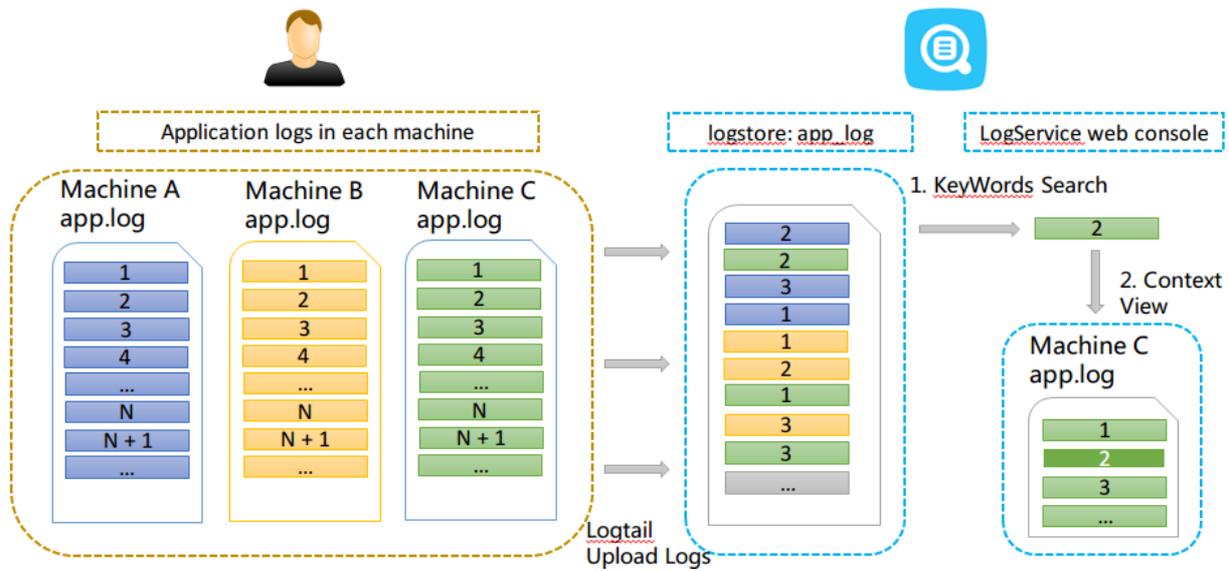
Scenarios

For example, a transaction on an O2O takeout website is logged in an application log file on a server as follows: **User logon > Browse products > Select an item > Add to a shopping cart > Place an order > Payment > Deduction > Generate an order.**

If the order fails, the O&M personnel must locate the cause at the earliest opportunity. In traditional contextual queries, the O&M personnel must be authorized before logging on to each server where the application is deployed. Then, the O&M personnel must use the order ID as a keyword to search application log files to locate the cause of the failure.

In Log Service, the O&M personnel can perform the following steps to locate the cause of the failure:

1. Install Logtail on servers. Then log on to the Log Service console to add the servers to machine groups and configure log collection. After the configurations are complete, Logtail starts to upload incremental logs.
2. On the search and analysis page of the Log Service console, specify the time range and find the error log entry based on the order ID.
3. Based on the error log entry, page up until you find log entries related to the error log entry. For example, you may want to find a log entry that records a credit card payment failure.



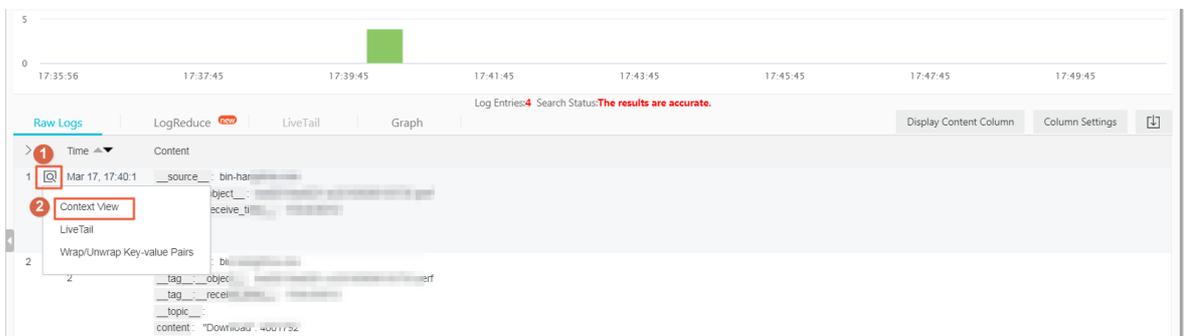
Note The contextual query feature does not support syslog.

Benefits

- Intrusions into applications or changes to log file formats are avoided.
- You can view contextual log entries of a specified log entry in a log file on a server that has been registered in the Log Service console. This helps you avoid logging on to each server to search for logs that you want.
- You can specify the time range based on the time when an event occurs to locate suspicious log entries. Then you can perform a contextual query in the Log Service console. This improves troubleshooting efficiency.
- Data loss caused by log file rotation or insufficient storage space is avoided. You can view historical log data in the Log Service console at any time.

Procedure

1. Log on to the Log Service console.
2. Click the target project in the Projects section.
3. Click the icon next to the name of the Logstore, and then select Search & Analysis.
4. Enter a search and analytic statement, select a time range, and then click Search & Analytics. On the query results page, if the Context View icon is available in the drop-down list of the icon to the left of a log entry, the log entry supports contextual query.



5. Click the icon to the left of a log entry, and select Context View from the drop-down list. On the page that appears, view the contextual log entries of the selected log entry.
6. Scroll up and down to view more contextual log entries. To view earlier or later contextual log entries, click

Old or New.

30.4.7.5. Saved search

This topic describes how to save a search and analytic statement as a saved search for a Logstore. The saved search feature allows you to search and analyze log data in an efficient way.

Prerequisites

The index feature is enabled and configured.

Context

If you need to frequently run a search and analytic statement, you can save the statement as a saved search. In later queries, you can click the name of the saved search on the left side of the search page to run the statement and view the result. You can also use the saved search in alert configurations. Log Service periodically runs the search and analytic statement and sends an alert if a query result meets the trigger condition.

If you want to select **Open Saved Search** in the Event Action field when you configure drill-down analysis, you must preset a saved search and set a placeholder in the query statement. For more information, see [Drill-down analysis](#).

Procedure

1. [Log on to the Log Service console](#).
2. Click the  icon next to the name of the Logstore, and then select **Search & Analysis**.
3. Enter a search and analytic statement in the search box, set a time range, and then click **Search & Analyze**.
4. Click **Save Search** in the upper-right corner of the page.



5. Configure the saved search.
 - i. Enter a **Saved Search Name**.
 - The name can contain only lowercase letters, digits, hyphens (-), and underscores (_).
 - The name must start and end with a lowercase letter or digit.
 - The name must be 3 to 63 characters in length.
 - ii. Check the values of the **Logstores**, **Topic**, and **Query** parameters. If the values of the **Logstores** and **Topic** parameters do not meet your requirements, follow these steps: Return to the **Logstores** page. On this page, find and click the name of the target Logstore. On the page that appears, enter the search and analytic statement in the **Search & Analyze** search box, and then click **Save Search** again.

- iii. (Optional) Select a part of the query statement, and then click **Generate Variable**. The generated variable is a placeholder variable. You can set the placeholder name in the **Variable Name** field. The selected characters are displayed in the **Default Value** field.

Note If you use this saved search for drill-down analysis in another chart where the variable is the same as the **Variable Name**, the **Default Value** is replaced with the chart value that you click to trigger the drill-down event. The search and analytic statement with the replaced chart value is executed. For more information, see [Drill-down analysis](#).

Saved Search Details ✕

* Saved Search Name

Attributes

Logstores

Topic

Query

Select the query statement to generate a placeholder variable. You can configure a drill-down configuration to replace the variable.

Variable Config

Variable Name: Default Value: Matching Mode: ✕

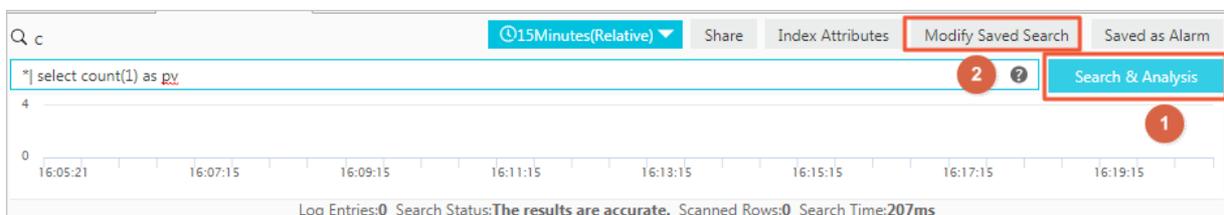
Result

6. Click **OK**.

What's next

To modify a saved search, perform the following operations:

Enter a new search and analytic statement, click **Search & Analytics** to run the statement, and then click **Modify Saved Search**.



30.4.7.6. Quick analysis

This topic describes the quick analysis feature of Log Service. You can use this feature to query log data with one click. This feature allows you to analyze the distribution of a field in a specified time range and reduce the query cost of key data.

Features

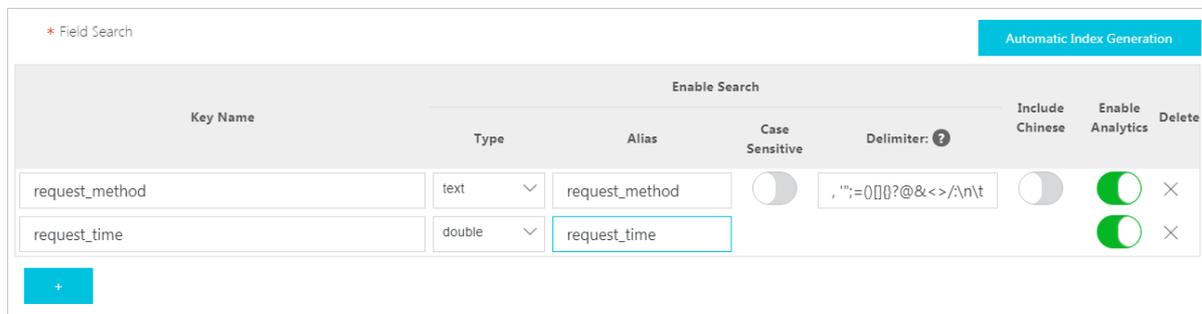
- Groups the first 100,000 values of a `text` field and provides statistics for the top 10 groups.
- Generates `approx_distinct` statements for `text` fields.
- Allows you to perform histogram-based statistics for the approximate distribution of `long` or `double` fields.
- Allows you to search for the maximum, minimum, and average of `long` or `double` fields and calculate the sum of the fields.
- Generates a query statement based on the quick analysis feature.

Prerequisites

Field indexes are configured.

- Indexes are configured for the fields that you need to search and analyze. For more information about how to enable the indexing feature, see [Enable the index feature and configure indexes for a Logstore](#).
- The name of a field is specified as the `key`. The data type, alias, and delimiter of the field are configured.

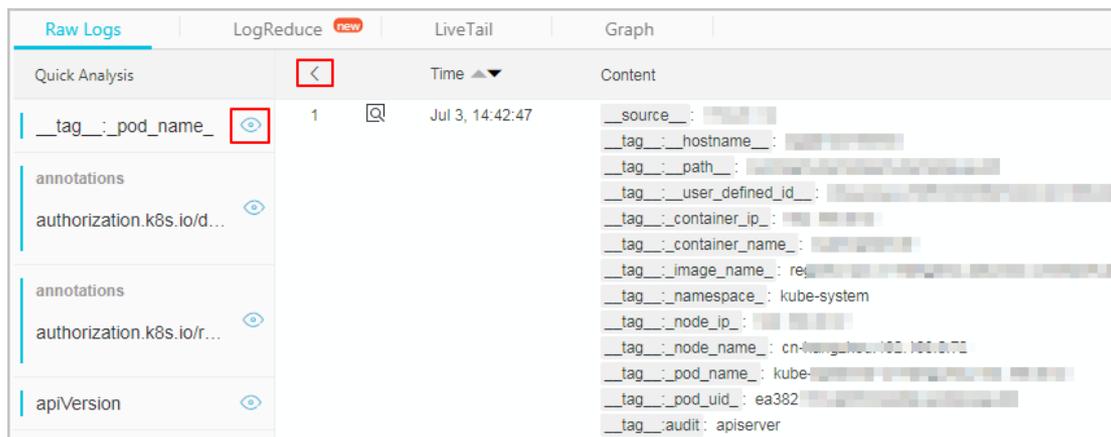
For example, if a log entry contains the `request_method` and `request_time` fields, you can configure indexes for the two fields, as shown in the following figure.



Instructions

After you configure indexes for specified fields, you can go to the Search & Analysis page and click the **Raw Logs** tab to view the specified fields. The fields are listed in the **Quick Analysis** pane on the left of the raw log entries. You can click the icon above the serial number to hide the Quick Analysis pane. You can also click the **Eye** icon to perform quick analysis based on the **Current Time Zone** and **Current Search** conditions.

Quick analysis



Data of the text type

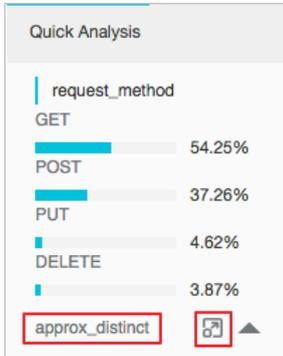
- Group and analyze log data

Click the **Eye** icon next to a `text` field to group the first 100,000 field values and return the percentages of the top 10 groups.

The following statement is used:

```
$Search | select ${keyName} , pv, pv *1.0/sum(pv) over() as percentage from( select count(1) as pv , "${keyName}" from (select "${keyName}" from log limit 100000) group by "${keyName}" order by pv desc) order by pv desc limit 10
```

The following figure shows the grouping and analytics result of the `request_method` field. `GET` requests account for the majority of requests.



- Calculate the number of unique values in a field

Under the target fields in the **Quick Analysis** pane, click **Count Distinct Values** to calculate the number of unique values in the `${keyName}` field.

- Fill the Search & Analyze search box with the grouping and analytics statement

Click the **Count Distinct Values** button on the right of the  icon. The Search & Analyze search box is filled with the grouping and analytics statement. You can edit the statement.

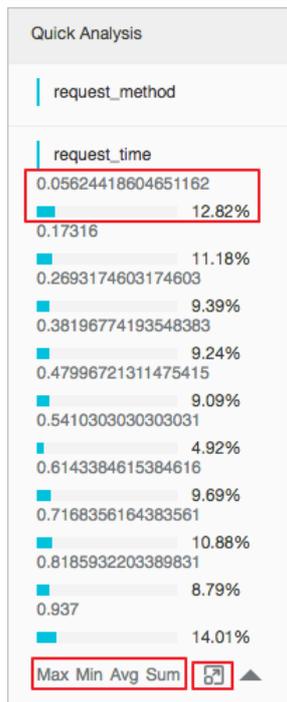
Data of the long and double types

- Display approximate distribution by using histograms

The number of `long` and `double` field values is large. The preceding grouping and analytics method is not suitable for the long and double data types. You can use the following statement to divide field values into 10 groups and display the approximate distribution of the values in a histogram:

```
$Search | select numeric_histogram(10, ${keyName})
```

The following figure shows the approximate distribution of the values in the `request_time` field. The largest percentage of request time is distributed around 0.059 seconds.



- Perform quick analysis by using the `Max` , `Min` , `Avg` , and `Sum` functions
 You can click `Max` under a field to search for the maximum value, `Min` to search for the minimum value, `Avg` to calculate the average value, and `Sum` to calculate the sum of the values.
- Fill the Search & Analyze search box with the query statement of the histogram approximate distribution
 Click the  icon next to `Sum` . The Search & Analyze search box is filled with the query statement of the histogram approximate distribution. You can edit the statement.

30.4.7.7. Other features

Log Service allows you to query log data by using various statements. It also provides many features for data search and analysis. This topic describes the raw log query, chart, saved search, dashboard, quick analysis, and alert features that Log Service supports.

Raw log query

After the index feature is enabled, enter one or more keywords in the search box and set the query time range. Then, click `Search & Analyze` to view the number of log entries displayed in a histogram, the raw logs, and the available charts.

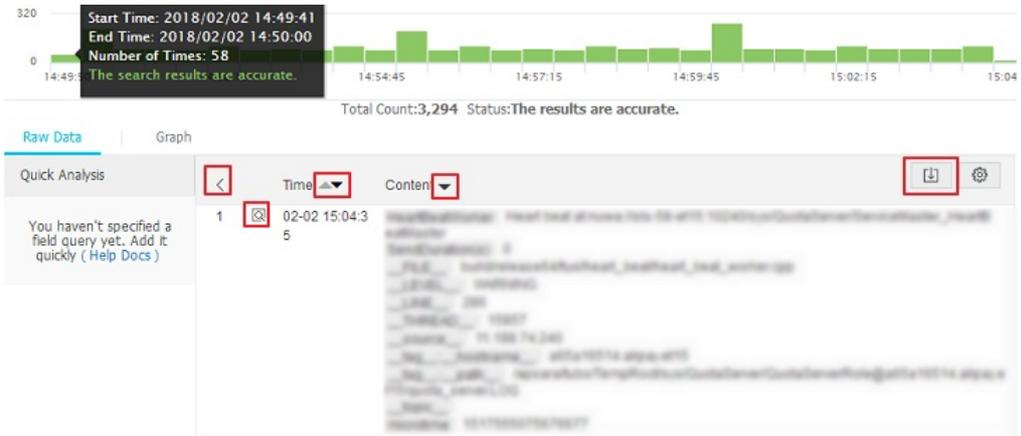
The histogram shows the time-based distribution of log entry occurrences. In the histogram, you can view the changes in the number of matched log entries over a period of time. You can click a rectangular to view the information about the occurrences within the specified time.

On the Raw Logs tab, you can view the matched log data in chronological order.

- You can click the triangle symbol next to `Time` to switch between the chronological order and reverse chronological order.
- You can click `Display Content Column` and then you can select `New Line` or `Full Line` to display the log entries. You can also set `Truncate Character String`.
- You can click the keyword of a field value in the log content to view all log entries that contain this keyword.

- You can click the  icon in the upper-right corner of the Raw Logs tab to download the query results in the CSV format. You can also click **Column Settings** to select fields from the displayed raw log entries. This provides you with an easier way to view target field values of each raw log entry in new columns.
- You can click **Context View** to view the 15 log entries before and after the current log entry in a log file. For more information, see [Contextual query](#).

Note The contextual query feature applies only to the data that is uploaded by using Logtail.



Charts

After you have enabled the index feature and entered a search and analytic statement, you can select a chart on the **Graph** tab to visualize the query results.

- Multiple types of charts are provided, such as the table and line chart. You can select a chart type and configure the chart based on your business requirements.
- You can add charts to a dashboard. For more information, see [Create and delete a dashboard](#).
- You can configure drill-down analysis for a chart. After you add a chart to a dashboard, you can click the chart data to trigger a drill down event and explore in-depth details about the data. For more information about how to configure drill-down analysis, see [Drill-down analysis](#).

Dashboards

Log Service provides the dashboard feature that can visualize the results of search and analytic statements. For more information, see [Create and delete a dashboard](#).

Alerts

Log Service allows you to configure alerts for your dashboard or the charts on your dashboard. You can configure rules to receive alerts from Log Service.

For more information, see [Configure alerts](#).

30.4.8. Analysis grammar

30.4.8.1. General aggregate functions

This topic describes the syntax and examples of general aggregate functions.

The search and analytics feature of Log Service allows you to use general aggregate functions for log analysis. The following table describes the supported general aggregate functions.

Function	Description	Example
<code>arbitrary(x)</code>	Returns a random value from among the values in the x field.	<code>latency > 100 select arbitrary(method)</code>
<code>avg(x)</code>	Returns the arithmetic mean of all values in the x field.	<code>latency > 100 select avg(latency)</code>
<code>checksum(x)</code>	Returns a Base64-encoded checksum of the values of the x field.	<code>latency > 100 select checksum(method)</code>
<code>count(*)</code>	Calculates the number of values of a field.	-
<code>count(x)</code>	Counts the number of non-null values of the x field.	<code>latency > 100 select count(method)</code>
<code>count(digit)</code>	Counts the number of values of a field. The <code>count(digit)</code> function is equivalent to the <code>count(1)</code> and <code>count(*)</code> functions.	-
<code>count_if(x)</code>	Returns the number of the occurrences of the TRUE value.	<code>latency > 100 select count_if(url like '%abc')</code>
<code>geometric_mean(x)</code>	Returns the geometric mean of the values in the x field.	<code>latency > 100 select geometric_mean(latency)</code>
<code>max_by(x,y)</code>	Returns the value of the x field that corresponds to the maximum value of the y field.	<code>* select min_by(method,latency,3)</code> : queries the method that corresponds to the maximum latency.
<code>max_by(x,y,n)</code>	Returns n values of the x field that corresponds to the n largest value of the y field.	<code>* select min_by(method,latency,3)</code> : queries the three methods that corresponds to the three maximum latencies.
<code>min_by(x,y)</code>	Returns the value of the x field that corresponds to the smallest value of the y field.	<code>* select min_by(x,y)</code> : queries the method that corresponds to the minimum latency.
<code>min_by(x,y,n)</code>	Returns n values of the x field that corresponds to the n smallest values of the y field.	<code>* select min_by(method,latency,3)</code> : queries the three methods that corresponds to the three minimum latencies.
<code>max(x)</code>	Returns the maximum value among all values in the x field.	<code>latency > 100 select max(inflow)</code>
<code>min(x)</code>	Returns the minimum value among all values in the x field.	<code>latency > 100 select min(inflow)</code>
<code>sum(x)</code>	Returns the sum among all values in the x field.	<code>latency > 10 select sum(inflow)</code>

Function	Description	Example
<code>bitwise_and_agg(x)</code>	Returns the bitwise AND of all values in the x field in two's complement representation.	-
<code>bitwise_or_agg(x)</code>	Returns the bitwise OR of all values in the x field in two's complement representation.	-

30.4.8.2. Security check functions

Security check functions in Log Services are designed based on the globally shared WhiteHat Security asset library. This topic describes security check functions that you can use to check whether an IP address, domain name, or URL in logs is secure.

Scenarios

- O&M personnel of enterprises and institutions in Internet, gaming, information, and other industries that require robust O&M services can use security check functions to identify suspicious requests or attacks. They can also use the functions to implement in-depth analysis and defend against potential attacks.
- O&M personnel of enterprises and institutions in banking, securities, e-commerce, and other industries that require strong protection for internal assets can use security check functions to identify requests to suspicious websites and downloads initiated by trojans. Then the O&M personnel can take immediate actions to prevent potential losses.

Features

- **Reliability:** built upon the globally shared WhiteHat Security asset library that is updated in a timely manner.
- **Efficiency:** capable of screening millions of IP addresses, domain names, and URLs within seconds.
- **Ease of use:** supports the analysis of network logs by using the `security_check_ip`, `security_check_domain`, and `security_check_url` functions.
- **Flexibility:** supports interactive queries, report creation, and alert configurations and subsequent actions.

Functions

Function	Description	Example
<code>security_check_ip</code>	<p>Checks whether an IP address is secure.</p> <ul style="list-style-type: none"> • The value 1 indicates that the specified IP address is suspicious. • The value 0 indicates that the specified IP address is secure. 	<pre>select security_check_ip(real_client_ip)</pre>
<code>security_check_domain</code>	<p>Checks whether a domain name is secure.</p> <ul style="list-style-type: none"> • The value 1 indicates that the specified domain name is suspicious. • The value 0 indicates that the specified domain name is secure. 	<pre>select security_check_domain(site)</pre>

Function	Description	Example
security_check_url	<p>Checks whether a URL is secure.</p> <ul style="list-style-type: none"> The value 1 indicates that the specified URL is suspicious. The value 0 indicates that the specified URL is secure. 	<pre>select security_check_domain(concat(host, url))</pre>

Examples

- Check external suspicious requests and generate reports

For example, an e-commerce enterprise collects logs from its NGINX servers and wants to scan suspicious client IP addresses. To do this, the enterprise can pass the ClientIP field in logs that are collected from the NGINX servers to the `security_check_ip` function and filter out IP addresses associated with the returned value 1. Then the enterprise can query the countries where the IP addresses are located and ISPs to which the IP addresses belong.

SQL statement for this scenario:

```
* | select ClientIP, ip_to_country(ClientIP) as country, ip_to_provider(ClientIP) as provider, count(1) as PV where security_check_ip(ClientIP) = 1 group by ClientIP order by PV desc
```

Display the ISPs and countries in a map.



- Check internal suspicious requests and send alerts

For example, a securities operator collects logs of its internal devices that access the Internet through gateways. To check requests to suspicious websites, the operator can run the following statement:

```
* | select client_ip, count(1) as PV where security_check_ip(remote_addr) = 1 or security_check_site(site) = 1 or security_check_url(concat(site, url)) = 1 group by client_ip order by PV desc
```

The operator can save this statement as a saved search and configure an alert. An alert is triggered when a client frequently accesses suspicious websites. The statement in the alert can be configured to run every five minutes to check if a client has frequently (more than five times) accessed suspicious websites in the past one hour. The following figure shows the configurations of an alert.

创建告警
✕

告警配置

通知

* 告警名称 5/64

* 添加到仪表盘 创建 ▼ 12/64

* 图表名称 5/64

查询语句

* | select client_ip, count(1) as PV where security_check_ip(remote_addr) = 1 or security_check_site(site) = 1 or security_check_url(concat(site, url)) = 1 group by client_ip order by PV desc

* 查询区间 🕒 1小时 (相对) ▼

* 检查频率 固定间隔 ▼ + - 分钟 ▼

* 触发条件

支持加 (+)、减 (-)、乘 (*)、除 (/)、取模 (%) 5种基础运算符和大于 (>)、大于等于 (>=)、小于 (<)、小于等于 (<=)、等于 (==)、不等于 (!=)、正则匹配 (==)、正则不匹配 (!~) 8种比较运算符。 [帮助文档](#)

高级选项 ▼

* 触发通知阈值 + -

* 通知间隔 5分钟 ▼

30.4.8.3. Map functions

This topic describes the syntax and examples of map functions.

The following table describes the supported map functions.

Function	Description	Example
Subscript operator []	Retrieves the value corresponding to a specified key from a map.	-
histogram(x)	Groups the values of the parameter x and calculates the number of occurrences of each value. The syntax is equivalent to <code>select count group by x</code> .	The statement <code>latency > 10 select histogram(status)</code> is equivalent to the statement <code>latency > 10 select count(1) group by status</code> .

🔍 Note The returned data is in the JSON format.

Function	Description	Example
<code>histogram_u(x)</code>	<p>Groups the values of the parameter <code>x</code> and calculates the number of occurrences of each value.</p> <p> Note The returned data is in the format of multiple rows and columns.</p>	<p>The statement <code>latency > 10 select histogram_u(status)</code> is equivalent to the statement <code>latency > 10 select count(1) group by status</code>.</p>
<code>map_agg(Key,Value)</code>	<p>Returns a random value of the key in the format of a map that consists of key-value pairs.</p>	<pre>latency > 100 select map_agg(method,latency)</pre>
<code>multimap_agg(Key,Value)</code>	<p>Returns all values of the key in the format of a map that consists of key-value pairs.</p>	<pre>latency > 100 select multimap_agg(method,latency)</pre>
<code>cardinality(x) → bigint</code>	<p>Returns the cardinality of the map <code>x</code>.</p>	-
<code>element_at(map <K, V> , key) → V</code>	<p>Returns the value for the specified key.</p>	-
<code>map() → map <unknown, unknown></code>	<p>Returns an empty map.</p>	-
<code>map(array <K> , array <V>) → map <K,V></code>	<p>Returns a map where each key-value pair consists of two elements from two separate arrays.</p>	<pre>SELECT map(ARRAY[1,3], ARRAY[2,4]); - {1 -> 2, 3 -> 4}</pre>
<code>map_from_entries(array <row<K, V>>) → map <K,V></code>	<p>Converts a multi-dimensional array to a map.</p>	<pre>SELECT map_from_entries(ARRAY[(1, 'x'), (2, 'y')]); - {1 -> 'x', 2 -> 'y'}</pre>
<code>map_concat(map1 <K, V> , map2 <K, V> , ..., mapN <K, V>) → map <K,V></code>	<p>Returns a map that is the union of all specified maps. If a key is found in multiple specified maps, the value of the key in the returned map is the value of the key that occurs in the last specified map.</p>	-
<code>map_filter(map <K, V> , function) → map <K,V></code>	<p>For more information, see the <code>map_filter</code> function in Lambda functions.</p>	-
<code>map_keys(x <K, V>) → array <K></code>	<p>Returns an array of keys in the specified map.</p>	-
<code>map_values(x <K, V>) → array <V></code>	<p>Returns an array of values in the specified map.</p>	-

30.4.8.4. Approximate functions

This topic describes the syntax and examples of approximate functions that Log Service supports for log analysis.

The following table describes the supported approximate functions.

Function	Description	Example
<code>approx_distinct(x)</code>	Returns the approximate number of distinct values of the x field.	-
<code>approx_percentile(x,percentage)</code>	Returns the value located at the specified approximate percentage among the sorted values of the x field.	<code>approx_percentile(x,0.5)</code> : returns the median among the sorted values of the x field.
<code>approx_percentile(x,percentages)</code>	Returns values located at multiple specified approximate percentages among the sorted values of the x field. This function works in a similar manner to the <code>approx_percentile(x,percentage)</code> function.	<code>approx_percentile(x,array[0.1,0.2])</code>
<code>numeric_histogram(buckets, Value)</code>	<p>Distributes all values of the <i>Value</i> field into multiple buckets. The <i>buckets</i> parameter specifies the number of buckets.</p> <p>The key of every bucket and the number of values in a bucket are returned. This function is equivalent to <code>select count group by</code> .</p> <p>Note The response is in the JSON format.</p>	<p><code>method:POST select numeric_histogram(10,latency)</code> : distributes the latencies of POST requests into 10 buckets and calculates the number of latencies in each bucket.</p>
<code>numeric_histogram_u(buckets, Value)</code>	<p>Distribute all values of the <i>Value</i> field into multiple buckets. The <i>buckets</i> parameter specifies the number of buckets.</p> <p>The key of every bucket and the number of values in a bucket are returned. This function is equivalent to <code>select count group by</code> .</p> <p>Note The returned data is in the format of multiple rows and columns.</p>	<p><code>method:POST select numeric_histogram(10,latency)</code> : distributes the latency data of POST requests into 10 buckets and calculates the number of latency data in each bucket.</p>

Note The number of values in every bucket is evenly distributed. It is returned along with the average value of all values in a bucket.

30.4.8.5. Mathematical statistics functions

This topic describes the syntax and examples of mathematical statistics functions.

The search and analytics feature of Log Service allows you to use mathematical statistics functions for log analysis. The following table describes the supported mathematical statistics functions.

Function	Description	Example
<code>corr(y, x)</code>	Returns the correlation coefficient of input values. The result ranges from 0 to 1.	<code>latency>100 select corr(latency,request_size)</code>
<code>covar_pop(y, x)</code>	Returns the population covariance of input values.	<code>latency>100 select covar_pop(request_size,latency)</code>
<code>covar_samp(y, x)</code>	Returns the sample covariance of input values.	<code>latency>100 select covar_samp(request_size,latency)</code>
<code>regr_intercept(y, x)</code>	Returns the linear regression intercept of input values. y is the dependent value. x is the independent value.	<code>latency>100 select regr_intercept(request_size,latency)</code>
<code>regr_slope(y,x)</code>	Returns the linear regression slope of input values. y is the dependent value. x is the independent value.	<code>latency>100 select regr_slope(request_size,latency)</code>
<code>stddev(x)</code> or <code>stddev_samp(x)</code>	Returns the sample standard deviation of the values in the x field.	<code>latency>100 select stddev(latency)</code>
<code>stddev_pop(x)</code>	Returns the population standard deviation of the values in the x field.	<code>latency>100 select stddev_pop(latency)</code>
<code>variance(x)</code> or <code>var_samp(x)</code>	Returns the sample variance of the values in the x field.	<code>latency>100 select variance(latency)</code>
<code>var_pop(x)</code>	Returns the population variance of the values in the x field.	<code>latency>100 select variance(latency)</code>

30.4.8.6. Mathematical calculation functions

This topic describes the syntax and examples of mathematical calculation functions.

By including mathematical calculation functions in SQL statements, you can perform mathematical calculation on log query results.

Mathematical operators

Mathematical operators include the plus sign (+), minus sign (-), multiplication sign (*), division sign (/), and percent sign (%). These operators can be used in SELECT statements.

Example:

```
*|select avg(latency)/100 , sum(latency)/count(1)
```

Mathematical calculation functions

Log Service supports the following mathematical calculation functions.

Function	Description
<code>abs(x)</code>	Returns the absolute values of the values in the x field.
<code>cbrt(x)</code>	Returns the cube roots of the values in the x field.
<code>ceiling(x)</code>	Returns the rounded-up nearest integers of the values in the x field.
<code>cosine_similarity(x,y)</code>	Returns the cosine similarity between the sparse vectors x and y.
<code>degrees</code>	Converts angles in radians to degrees.
<code>e()</code>	Returns the Euler's number.
<code>exp(x)</code>	Returns Euler's number raised to the power of the values in the x field.
<code>floor(x)</code>	Returns the rounded-down nearest integers of the values in the x field.
<code>from_base(string,radix)</code>	Returns the radix number representation of a string.
<code>ln(x)</code>	Returns the natural logarithm of x.
<code>log2(x)</code>	Returns the base 2 logarithm of x.
<code>log10(x)</code>	Returns the base 10 logarithm of x.
<code>log(x,b)</code>	Returns the base b logarithm of x.
<code>pi()</code>	Returns the constant Pi.
<code>pow(x,b)</code>	Returns x raised to the power of b.
<code>radians(x)</code>	Converts angle x in degrees to radians.
<code>rand()</code>	Returns a random number.
<code>random(0,n)</code>	Returns a random number from 0 to n (exclusive).
<code>round(x)</code>	Returns x rounded to the nearest integer.
<code>round(x,y)</code>	Returns x rounded to y decimal places. For example, <code>round(1.012345,2) = 1.01</code> .
<code>sqrt(x)</code>	Returns the square root of x.
<code>to_base(x,radix)</code>	Returns the radix number representation of x.
<code>truncate(x)</code>	Returns x rounded to integer by dropping digits after the decimal point.
<code>acos(x)</code>	Returns the arc cosine of x.

Function	Description
<code>asin(x)</code>	Returns the arc sine of x.
<code>atan(x)</code>	Returns the arc tangent of x.
<code>atan2(y,x)</code>	Returns the arc tangent of y/x.
<code>cos(x)</code>	Returns the cosine of x.
<code>sin(x)</code>	Returns the sine of x.
<code>cosh(x)</code>	Returns the hyperbolic cosine of x.
<code>tan(x)</code>	Returns the tangent of x.
<code>tanh(x)</code>	Returns the hyperbolic tangent of x.
<code>infinity()</code>	Returns the value representing positive infinity.
<code>is_infinity(x)</code>	Determines whether x is infinite.
<code>is_finity(x)</code>	Determines whether x is finite.
<code>is_nan(x)</code>	Determines whether x is not-a-number.

30.4.8.7. String functions

This topic describes string functions that Log Service supports for log data search and analytics.

This following table lists the functions and their descriptions.

Function	Description
<code>chr(x)</code>	Converts an integer to an ASCII character. For example, the result of <code>chr(65)</code> is <code>A</code> .
<code>codepoint(x)</code>	Converts an ASCII character to a code point of the integer type. For example, the result of <code>codepoint('A')</code> is <code>65</code> .
<code>length(x)</code>	Returns the length of a string.
<code>levenshtein_distance(string1, string2)</code>	Returns the Levenshtein distance of string1 and string2.
<code>lower(string)</code>	Converts all uppercase characters in a string into lowercase characters.
<code>lpad(string, size, padstring)</code>	Pads a string to the specified size. If the length of the string is shorter than the specified size, padstring is used to left pad the string. If the length of the string is longer than the specified size, the string is truncated with the specified size.

Function	Description
<code>rpad(string, size, padstring)</code>	Right pads a <code>string</code> with the specified padding. The implementation is similar to that of the <code>lpad</code> function.
<code>ltrim(string)</code>	Removes whitespace characters from the left side of a string.
<code>replace(string, search)</code>	Removes all occurrences of a substring <code>search</code> from a string.
<code>replace(string, search,rep)</code>	Replaces all occurrences of a substring <code>search</code> in a string with another substring <code>rep</code> .
<code>reverse(string)</code>	Reverses a string.
<code>rtrim(string)</code>	Removes whitespace characters from the right side of a string.
<code>split(string,delimiter,limit)</code>	Splits a string based on the specified delimiter and returns an array with the maximum number of elements at <code>limit</code> . The index of the first element in the array is 1.
<code>split_part(string,delimiter,offset)</code>	Splits a string based on a delimiter into an array of substrings and returns the element with the index specified by the <code>offset</code> parameter.
<code>split_to_map(string, entryDelimiter, keyValueDelimiter)</code> → <code>map<varchar, varchar></code>	Splits a string based on the <code>entryDelimiter</code> into multiple entries, each of which is then split based on the <code>keyValueDelimiter</code> into a key and value. This function returns a <code>map</code> .
<code>position(substring IN string)</code>	Returns the position of the first occurrence of the specified substring in a string.
<code>strpos(string, substring)</code>	Returns the position of the first occurrence of the specified substring in a string. Positions start with 1. If the substring is found, 0 is returned.
<code>substr(string, start)</code>	Returns a substring from the start position. Positions start with 1.
<code>substr(string, start, length)</code>	Returns a substring of a specified length from start position. Positions start with 1. The <code>length</code> parameter specifies the length of the substring returned.
<code>trim(string)</code>	Removes leading and trailing whitespace characters from a string.
<code>upper(string)</code>	Converts all lowercase characters in a string into uppercase characters.
<code>concat(string,string...)</code>	Concatenates multiple strings into a single string.
<code>hamming_distance (string1,string2)</code>	Returns the Hamming distance of <code>string1</code> and <code>string2</code> .

Note Strings must be enclosed in single quotation marks ('). Double quotation marks (") are used to enclose field names. For example, `a='abc'` indicates `field a = string 'abc'`, and `"a"="abc"` indicates `field a = field abc`.

30.4.8.8. Date and time functions

This topic describes the available time functions, date functions, interval functions, and a time series padding function in Log Service. You can use these functions when you analyze data.

Date and time data types

- **Unix timestamp:** specifies the number of seconds that have elapsed since 1970-01-01 00:00:00 UTC. The value is in the format of an integer. For example, `1512374067` indicates `Mon Dec 4 15:54:27 CST 2017`. The `_time__` field in every log entry is of this type.
- **Timestamp:** specifies the date and time in the format of a string. For example, `2017-11-01 13:30:00`.

Date functions

The following table lists the commonly used date functions that are supported in Log Service.

Function	Description	Example
<code>current_date</code>	Returns the current date.	<code>latency>100 select current_date</code>
<code>current_time</code>	Returns the current time.	<code>latency>100 select current_time</code>
<code>current_timestamp</code>	Returns the current timestamp. This function is equivalent to the combination of the <code>current_date</code> and <code>current_time</code> functions.	<code>latency>100 select current_timestamp</code>
<code>current_timezone()</code>	Returns the current time zone.	<code>latency>100 select current_timezone()</code>
<code>from_iso8601_timestamp(string)</code>	Parses an ISO 8601 formatted string into a timestamp that specifies the time zone.	<code>latency>100 select from_iso8601_timestamp(iso8601)</code>
<code>from_iso8601_date(string)</code>	Parses an ISO 8601 formatted string into a date.	<code>latency>100 select from_iso8601_date(iso8601)</code>
<code>from_unixtime(unixtime)</code>	Parses a Unix timestamp into a timestamp.	<code>latency>100 select from_unixtime(1494985275)</code>
<code>from_unixtime(unixtime,string)</code>	Parses a Unix timestamp into a timestamp based on the time zone that is specified by the string parameter.	<code>latency>100 select from_unixtime(1494985275,'Asia/Shanghai')</code>
<code>localtime</code>	Returns the local time.	<code>latency>100 select localtime</code>
<code>localtimestamp</code>	Returns the local timestamp.	<code>latency>100 select localtimestamp</code>
<code>now()</code>	Returns the current date and time. This function is equivalent to the <code>current_timestamp</code> function.	N/A
<code>to_unixtime(timestamp)</code>	Parses a timestamp into a Unix timestamp.	<code>* select to_unixtime('2017-05-17 09:45:00.848 Asia/Shanghai')</code>

Time functions

The following table lists the time functions that you can use in Log Service to parse time in the formats supported in MySQL, such as %a, %b, and %y.

Function	Description	Example
<code>date_format(timestamp, format)</code>	Formats a timestamp in the specified format.	<code>latency>100 select date_format(date_parse('2017-05-17 09:45:00','%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')</code>
<code>date_parse(string, format)</code>	Parses a formatted string into a timestamp.	<code>latency>100 select date_format(date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')</code>

Formats

Format	Description
%a	The week day in abbreviation, such as Sun and Sat.
%b	The month in abbreviation, such as Jan and Dec.
%c	The month in the numeric format. Valid values: 1 to 12.
%D	The day of the month with a suffix, such as 0th, 1st, 2nd, and 3rd.
%d	The day of the month in the numeric format. Valid values: 01 to 31.
%e	The day of the month in the numeric format. Valid values: 1 to 31.
%H	The hour that applies the 24-hour clock convention. Valid values: 00 to 23.
%h	The hour that applies the 12-hour clock convention. Valid values: 01 to 12.
%l	The hour that applies the 12-hour clock convention. Valid values: 1 to 12.
%i	The minute in the numeric format. Valid values: 00 to 59.
%j	The day of the year. Valid values: 001 to 366.
%k	The hour. Valid values: 0 to 23.
%l	The hour. Valid values: 1 to 12.
%M	The month. Valid values: January to December.
%m	The month in the numeric format. Valid values: 01 to 12.
%p	The abbreviation that indicates the morning or afternoon. Valid values: a.m. and p.m..
%r	The time that applies the 12-hour clock convention: <code>hh:mm:ss AM/PM</code> .
%S	The second. Valid values: 00 to 59.
%s	The second. Valid values: 00 to 59.

Format	Description
%T	The time that applies the 24-hour clock convention, formatted in hh:mm:ss .
%U	The week of the year. Sunday is the first day of a week. Valid values: 00 to 53.
%u	The week of the year. Monday is the first day of a week. Valid values: 00 to 53.
%V	The week of the year. Sunday is the first day of a week. This format is used together with %X. Valid values: 01 to 53.
%v	The week of the year. Monday is the first day of a week. This format is used in combination with %x. Valid values: 01 to 53.
%W	The day of the week. Valid values: Sunday to Saturday.
%w	The day of the week. Valid values: 0 to 6. The value 0 indicates Sunday.
%Y	The year in the 4-digit format.
%y	The year in the 2-digit format.
%%	Escapes the second percent sign (%).

Truncation functions

Log Service supports a truncation function, which can truncate a time by the second, minute, hour, day, month, or year. Typically, this function is used for time-based analytics.

- **Syntax**

```
date_trunc(unit, x)
```

- **Parameters**

The value of the x parameter can be a timestamp or Unix timestamp.

The following table lists the values of the unit parameter and the responses when the x parameter is set to 2001-08-22 03:04:05.000 .

Unit	Response
second	2001-08-22 03:04:05.000
minute	2001-08-22 03:04:00.000
hour	2001-08-22 03:00:00.000
day	2001-08-22 00:00:00.000
week	2001-08-20 00:00:00.000
month	2001-08-01 00:00:00.000
quarter	2001-07-01 00:00:00.000
year	2001-01-01 00:00:00.000

- **Example**

The `date_trunc` function is applicable only to analytics at a fixed time interval. To implement analytics at a flexible interval, for example, every 5 minutes, you need to use a `GROUP BY` clause according to the mathematical modulus method.

```
* | SELECT count(1) as pv, __time__ - __time__% 300 as minute5 group by minute5 limit 100
```

In the preceding statement, `%300` indicates that modulo and truncation are performed every 5 minutes.

The following example shows how to use the `date_trunc` function:

```
*|select date_trunc('minute', __time__) as t,
      truncate (avg(latency) ) ,
      current_date
      group by t
      order by t desc
      limit 60
```

Interval functions

Interval functions perform interval-related calculation. For example, you can use interval functions to add or subtract an interval based on a date, or calculate the interval between two dates.

Function	Description	Example
<code>date_add (unit, value, timestamp)</code>	Adds an interval <code>value</code> of the <code>unit</code> type to a <code>timestamp</code> . To subtract an interval, use a negative <code>value</code> .	<code>date_add('day', -7, '2018-08-09 00:00:00')</code> : indicates seven days before August 9.
<code>date_diff(unit, timestamp1, timestamp2)</code>	Returns the time difference between <code>timestamp1</code> and <code>timestamp2</code> expressed in terms of <code>unit</code> .	<code>date_diff('day', '2018-08-02 00:00:00', '2018-08-09 00:00:00') = 7</code>

The following table lists the values of the `unit` parameter that are supported by the interval functions.

Value	Description
millisecond	The millisecond.
second	The second.
minute	The minute.
hour	The hour.
day	The day.
week	The week.
month	The month.
quarter	The quarter, namely, three months.
year	The year.

Time series padding function

The time series padding function is used to pad time series and corresponding data.

Note This function must be used in combination with the `group by time order by time` clause. When used together, the `order by` clause does not support the `desc` sorting method.

• **Syntax**

```
time_series(time_column, window, format, padding_data)
```

• **Parameters**

Parameter	Description
<i>time_column</i>	The name of the time field in a log entry. The default field name is <code>__time__</code> . The field value is of the LONG or TIMESTAMP type.
<i>window</i>	The time window for a data query. It is composed of a number and a unit. Unit: s (seconds), m (minutes), H (hours), and d (days). For example, 2h, 5m, or 3d.
<i>format</i>	The MySQL time format displayed.
<i>padding_data</i>	The content to be added for a time point. Valid values: <ul style="list-style-type: none"> 0: adds 0. null: adds null. last: adds the value corresponding to the last time point. next: adds the value corresponding to the next time point. avg: adds the average value of the last and next values.

• **Example**

The following statement is used to format data every 2 hours:

```
*| select time_series(__time__, '2h', '%Y-%m-%d %H:%i:%s', '0') as stamp, count(*) as num from log group by stamp order by stamp
```

30.4.8.9. URL functions

This topic describes the syntax of URL functions and provides examples.

URL functions extract fields from standard URLs. A standard URL is described as follows:

```
[protocol:][//host[:port]][path][? query][#fragment]
```

Common URL functions

Function	Description	Example	
		Statement	Response
<code>url_extract_fragment(url)</code>	Extracts the fragment identifier from a URL and returns the fragment identifier of the VARCHAR type.	<pre>* select url_extract_fragment('https://sls.console.aliyun.com/#/project/dashboard-demo/categoryList')</pre>	<pre>/project/dashboard-demo/categoryList</pre>

Function	Description	Example	
		Statement	Response
<code>url_extract_host(url)</code>	Extracts the host information from a URL and returns the host information of the VARCHAR type.	<pre>* select url_extract_host('http://www.aliyun.com/product/sls')</pre>	<code>www.aliyun.com</code>
<code>url_extract_parameter(url, name)</code>	Extracts the value of the name parameter in the query from a URL and returns the value of the VARCHAR type.	<pre>* select url_extract_parameter('http://www.aliyun.com/product/sls?userid=testuser','userid')</pre>	<code>testuser</code>
<code>url_extract_path(url)</code>	Extracts the path from a URL and returns the path of the VARCHAR type.	<pre>* select url_extract_path('http://www.aliyun.com/product/sls?userid=testuser')</pre>	<code>/product/sls</code>
<code>url_extract_port(url)</code>	Extracts the port number from a URL and returns the port number of the BIGINT type.	<pre>* select url_extract_port('http://www.aliyun.com:80/product/sls?userid=testuser')</pre>	<code>80</code>
<code>url_extract_protocol(url)</code>	Extracts the protocol from a URL and returns the protocol of the VARCHAR type.	<pre>* select url_extract_protocol('http://www.aliyun.com:80/product/sls?userid=testuser')</pre>	<code>http</code>
<code>url_extract_query(url)</code>	Extracts the query string from a URL and returns the query string of the VARCHAR type.	<pre>* select url_extract_query('http://www.aliyun.com:80/product/sls?userid=testuser')</pre>	<code>userid=testuser</code>
<code>url_encode(value)</code>	Encodes a URL.	<pre>* select url_encode('http://www.aliyun.com:80/product/sls?userid=testuser')</pre>	<code>http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser</code>
<code>url_decode(value)</code>	Decodes a URL.	<pre>* select url_decode('http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser')</pre>	<code>http://www.aliyun.com:80/product/sls?userid=testuser</code>

30.4.8.10. Regular expression functions

This topic describes the available regular expression functions. You can use these functions when you query and analyze data in Log Service.

A regular expression function parses a string and returns the required substrings.

The following table lists common regular expression functions.

Function	Description	Example
<code>regexp_extract_all(string, pattern)</code>	Returns an array where each element is a substring that matches the regular expression. These substrings derive from the specified string.	The returned result of <code>* SELECT regexp_extract_all('5a 67b 890m', '\d+')</code> is <code>['5','67','890']</code> . The returned result of <code>* SELECT regexp_extract_all('5a 67a 890m', '(\d+)a')</code> is <code>['5a','67a']</code> .
<code>regexp_extract_all(string, pattern, group)</code>	Returns an array where each element is a part of a substring that matches the regular expression. This part is the content in the group parameter value of the <code>()</code> of a substring that derives from the specified string.	The returned result of <code>* SELECT regexp_extract_all('5a 67a 890m', '(\d+)a',1)</code> is <code>['5','67']</code> .
<code>regexp_extract(string, pattern)</code>	Returns the first substring of the specified string that matches the regular expression.	The returned result of <code>* SELECT regexp_extract('5a 67b 890m', '\d+')</code> is <code>'5'</code> .
<code>regexp_extract(string, pattern, group)</code>	Returns a part of the first substring that matches the regular expression. This part is the content in the group parameter value of the <code>()</code> of the substring that derives from the specified string.	The returned result of <code>* SELECT regexp_extract('5a 67b 890m', '(\d+)([a-z]+)',2)</code> is <code>'a'</code> .
<code>regexp_like(string, pattern)</code>	Returns a Boolean value. If the string and its substrings cannot match the regular expression, the value <code>False</code> is returned.	The returned result of <code>* SELECT regexp_like('5a 67b 890m', '\d+m')</code> is <code>True</code> .
<code>regexp_replace(string, pattern, replacement)</code>	Replaces the substrings of the specified string that match the regular expression with the value of the replacement parameter.	The returned result of <code>* SELECT regexp_replace('5a 67b 890m', '\d+', 'a')</code> is <code>'aa ab am'</code> .
<code>regexp_replace(string, pattern)</code>	Removes the substrings of the specified string that match the regular expression. This function is equivalent to <code>regexp_replace(string, pattern, '')</code> .	The returned result of <code>* SELECT regexp_replace('5a 67b 890m', '\d+')</code> is <code>'a b m'</code> .
<code>regexp_split(string, pattern)</code>	Returns an array where each element is a substring of the specified string that is split based on the regular expression.	The returned result of <code>* SELECT regexp_split('5a 67b 890m', '\d+')</code> is <code>['a','b','m']</code> .

30.4.8.11. JSON functions

JSON functions can convert a string into a JSON type and extract JSON fields. The two major JSON data types are map and array. If a string cannot be converted to a value of the JSON type, null value is returned.

For information about how to expand JSON data into multiple rows, see [UNNEST function](#).

The following table lists the JSON functions that Log Service supports:

Function	Description	Example
<code>json_parse(string)</code>	Converts a string to a JSON-formatted data.	<code>SELECT json_parse('[1, 2, 3]') :</code> returns a JSON array.
<code>json_format(json)</code>	Converts JSON-formatted data to a string.	<code>SELECT json_format(json_parse('[1, 2, 3]')) :</code> returns a string.
<code>json_array_contains(json, value)</code>	Determines whether a value exists in a JSON array or in a string that contains a JSON array.	<code>SELECT json_array_contains(json_parse('[1, 2, 3]'), 2)</code> or <code>SELECT json_array_contains('[1, 2, 3]', 2)</code>
<code>json_array_get(json_array, index)</code>	Retrieves the element at the specified index in the JSON array. This function is equivalent to <code>json_array_contains</code> .	<code>SELECT json_array_get(['a', 'b', 'c'], 0):</code> returns 'a'
<code>json_array_length(json)</code>	Returns the length of the JSON array.	<code>SELECT json_array_length('[1, 2, 3]')</code> : returns 3
<code>json_extract(json, json_path)</code>	Extracts a value from a JSON object and returns a JSON object. The <code>json_path</code> expression works in a similar manner to <code>\$.store.book[0].title</code> .	<code>SELECT json_extract(json, '\$.store.book')</code>
<code>json_extract_scalar(json, json_path)</code>	Returns a string. This function works in a similar manner to <code>json_extract</code> .	N/A
<code>json_size(json,json_path)</code>	Retrieves the length of a JSON object or array.	<code>SELECT json_size('[1, 2, 3]')</code> : returns 3.

30.4.8.12. Type conversion functions

Type conversion functions convert the data type of a specified value or column in a query.

You can use the index attribute feature of Log Service to set the data type of a field to LONG, DOUBLE, TEXT, or JSON. You can also query fields of various data types, including BIGINT, DOUBLE, VARCHAR, and TIMESTAMP. To query fields of a specific data type, you can use type conversion functions to convert the data type configured in an index into the data type that you use in a query.

Syntax

 **Note** We recommend that you use the `try_cast()` function if a log contains dirty data. Otherwise, a query may fail due to the dirty data.

- Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, the query is terminated.

```
cast([key|value] AS type)
```

- Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, NULL is returned for the value, and the query continues.

```
try_cast([key|value] AS type)
```

Parameter	Description
key	The key of a field whose value data type is to be converted.
value	The field value whose data type is to be converted into the specified type.

Example

- To convert the numeric value 123 to a value of the VARCHAR type, use the following statement:

```
cast(123 AS varchar)
```

- To convert the data type of the uid field values to the VARCHAR type, use the following statement:

```
cast(uid AS varchar)
```

30.4.8.13. IP functions

This topic describes the syntax of IP functions and provides examples.

IP functions can identify whether an IP address is an intranet or Internet IP address. IP functions can also identify the country, province, and city where an IP address resides. For information about geohash functions, see [Geography functions](#).

Function	Description	Example
<code>ip_to_domain(ip)</code>	Identifies whether an IP address is an intranet or Internet IP address. This function returns intranet or internet.	<code>SELECT ip_to_domain(ip)</code>
<code>ip_to_country(ip)</code>	Identifies the country where an IP address resides.	<code>SELECT ip_to_country(ip)</code>
<code>ip_to_province(ip)</code>	Identifies the province where an IP address resides.	<code>SELECT ip_to_province(ip)</code>
<code>ip_to_city(ip)</code>	Identifies the city where an IP address resides.	<code>SELECT ip_to_city(ip)</code>
<code>ip_to_geo(ip)</code>	Identifies the longitude and latitude of the city where an IP address resides. The result is returned in the format of <code>latitude, longitude</code> .	<code>SELECT ip_to_geo(ip)</code>
<code>ip_to_city_geo(ip)</code>	Identifies the longitude and latitude of the city where an IP address resides. Each city has only one longitude and latitude. The result is returned in the format of <code>latitude, longitude</code> .	<code>SELECT ip_to_city_geo(ip)</code>

Function	Description	Example
<code>ip_to_provider(ip)</code>	Identifies the network service provider that assigns an IP address.	<code>SELECT ip_to_provider(ip)</code>
<code>ip_to_country(ip,'en')</code>	Identifies the country where an IP address resides. The function returns a country code.	<code>SELECT ip_to_country(ip,'en')</code>
<code>ip_to_country_code(ip)</code>	Identifies the country where an IP address resides. The function returns a country code.	<code>SELECT ip_to_country_code(ip)</code>
<code>ip_to_province(ip,'en')</code>	Identifies the province where an IP address resides.	<code>SELECT ip_to_province(ip,'en')</code>
<code>ip_to_city(ip,'en')</code>	Identifies the city where an IP address resides.	<code>SELECT ip_to_city(ip,'en')</code>

Example

- To query the number of requests that are not sent from an intranet, run the following statement:

```
* | SELECT count(1) where ip_to_domain(ip)! ='intranet'
```

- To query the top 10 provinces from which requests are sent, run the following statement:

```
* | SELECT count(1) as pv, ip_to_province(ip) as province GROUP BY province order by pv desc limit 10
```

Sample response

```
[
  {
    "__source__": "",
    "__time__": "1512353137",
    "province": "Zhejiang",
    "pv": "4045"
  }, {
    "__source__": "",
    "__time__": "1512353137",
    "province": "Shanghai",
    "pv": "3727"
  }, {
    "__source__": "",
    "__time__": "1512353137",
    "province": "Beijing",
    "pv": "954"
  }, {
    "__source__": "",
    "__time__": "1512353137",
    "province": "intranet IP address",
    "pv": "698"
  }, {
    "__source__": "",
    "__time__": "1512353137",
    "province": "Guangdong",
    "pv": "472"
  }, {
    "__source__": "",
    "__time__": "1512353137",
    "province": "Fujian",
    "pv": "71"
  }
]
```

The response contains an intranet IP address. You can use the `SELECT` statement to filter requests that are sent from the IP address.

- To query the top 10 provinces from which intranet requests are sent, run the following statement:

```
* | SELECT count(1) as pv, ip_to_province(ip) as province WHERE ip_to_domain(ip) != 'intranet' GROUP BY province ORDER BY pv desc limit 10
```

- To query the average request latency, the maximum request latency, and the request with the maximum latency from each country, run the following statement:

```
* | SELECT AVG(latency),MAX(latency),MAX_BY(requestId, latency) ,ip_to_country(ip) as country group by country limit 100
```

- To query the average latency of requests supported by each network service provider, run the following statement:

```
* | SELECT AVG(latency) , ip_to_provider(ip) as provider group by provider limit 100
```

- To query the longitude and latitude of the city to which an IP address belongs and show the city on a map, run the following statement:

```
* | SELECT count(1) as pv , ip_to_geo(ip) as geo group by geo order by pv desc
```

The following table shows the format of the result.

pv	geo
100	35.3284,-80.7459

30.4.8.14. GROUP BY syntax

This topic describes the GROUP BY syntax.

GROUP BY statements support multiple columns. A GROUP BY statement allows you to specify a column alias in a SELECT statement to as the corresponding KEY.

Example:

```
method:PostLogstoreLogs |select avg(latency),projectName,date_trunc('hour',__time__) as hour group by projectName, hour
```

The hour alias indicates the third SELECT column `date_trunc('hour',__time__)` . This improves the performance of complicated queries.

The GROUP BY statement supports the GROUPING SETS, CUBE, and ROLLUP clauses.

Example:

```
method:PostLogstoreLogs |select avg(latency) group by cube(projectName,logstore)
method:PostLogstoreLogs |select avg(latency) group by GROUPING SETS ( ( projectName,logstore), (projectName,method))
method:PostLogstoreLogs |select avg(latency) group by rollup(projectName,logstore)
```

Examples

- Use GROUP BY based on time

Each log has a built-in time column named `__time__` . When the analytics feature is enabled on one of the columns, the statistics of the time column are included.

The `date_trunc` function can truncate the time column to minute, hour, day, month, and year. The `date_trunc` function accepts an aligned unit and a column of the Unix timestamp type, such as `__time__` .

- PV statistics per hour and per minute

```
* | SELECT count(1) as pv , date_trunc('hour',__time__) as hour group by hour order by hour limit 100
* | SELECT count(1) as pv , date_trunc('minute',__time__) as minute group by minute order by minute limit 100
```

 **Note** 100 specifies that up to 100 rows can be retrieved. If a LIMIT clause is not specified, up to 10 rows of data are retrieved by default.

- `date_trunc` functions are available only for statistics at a fixed time interval. For statistics based on flexible time dimensions, for example, every 5 minutes, run a `GROUP BY` statement based on the mathematical modulus method.

```
* | SELECT count(1) as pv, __time__ - __time__% 300 as minute5 group by minute5 limit 100
```

In the preceding statement, `%300` indicates that the time is truncated in mod every 5 minutes.

- Retrieve non-aggregation columns from a `GROUP BY` statement

In standard SQL, if a `GROUP BY` statement is used during the `SELECT` operation, Log Service selects only the raw data of the column specified in the `GROUP BY` statement or performs aggregation on any columns.

For example, the following statement is invalid. Log Service cannot determine which row of `b` to return during the `GROUP BY` operation based on `a`, because `b` is not a `GROUP BY` column.

```
*|select a, b , count(c) group by a
```

Instead, you can use the `arbitrary()` function to return `b`.

```
*|select a, arbitrary(b), count(c) group by a
```

30.4.8.15. Window functions

This topic describes the syntax for window functions.

Window functions are used to perform calculations across rows of a log. Common SQL aggregate functions calculate the results of only one row or aggregate all rows into one row for calculation. Window functions support cross-row calculation and fill the calculation results in each row.

Syntax for window functions is

```
SELECT key1, key2, value,
       rank() OVER (PARTITION BY key2
                   ORDER BY value DESC) AS rnk
FROM orders
ORDER BY key1,rnk
```

The important part is

```
rank() OVER (PARTITION BY KEY1 ORDER BY KEY2 DESC)
```

`rank()` is an aggregate function. You can use any functions in analysis syntax or the function listed in this topic. `PARTITION BY` indicates the buckets based on which values are calculated.

Special aggregate functions used in windows

Function	Description
<code>rank()</code>	Returns the rank of a value within a group of values. The rank is one plus the number of preceding rows that are not peers of the current row.
<code>row_number()</code>	Returns a unique, sequential number for each row.
<code>first_value(x)</code>	Returns the first value of the window. In most cases, the function is used to obtain the maximum value after the values of the window are sorted.

Function	Description
last_value(x)	Returns the last value of the window. In most cases, the function is used to obtain the minimum value after the values of the window are sorted.
nth_value(x, offset)	Returns the value at the specified offset from the beginning of the window.
lead(x,offset,default_value)	Returns the value at offset rows after the current row in the window. If the target row does not exist, the default_value is returned.
lag(x,offset,default_value)	Returns the value at offset rows after the current row in the window. If the target row does not exist, the default_value is returned.

Example

- Rank the salaries of employees in their respective departments

```
* | select department, persionId, sallary , rank() over(PARTITION BY department order by sallary desc) as sallary_rank
k order by department,sallary_rank
```

Results

department	persionId	sallary	sallary_rank
dev	john	9000	1
dev	Smith	8000	2
dev	Snow	7000	3
dev	Achilles	6000	4
Marketing	Blan Stark	9000	1
Marketing	Rob Stark	8000	2
Marketing	Sansa Stark	7000	3

- Calculate the salaries of employees as percentages in their respective departments

```
* | select department, persionId, sallary *1.0 / sum(sallary) over(PARTITION BY department ) as sallary_percentage
```

Results

department	persionId	sallary	sallary_percentage
dev	john	9,000	0.3
dev	Smith	8,000	0.26
dev	Snow	7000	0.23
dev	Achilles	6000	0.2
Marketing	Blan Stark	9000	0.375
Marketing	Rob Stark	8000	0.333

department	persionId	sallary	sallary_percentage
Marketing	Sansa Stark	7000	0.29

- Calculate the daily UV increase over the previous day

```
* | select day ,uv, uv *1.0 /(lag(uv,1,0) over() ) as diff_percentage from
(
select approx_distinct(ip) as uv, date_trunc('day',__time__) as day from log group by day order by day asc
)
```

Results

day	uv	diff_percentage
2017-12-01 00:00:00	100	null
2017-12-02 00:00:00	125	1.25
2017-12-03 00:00:00	1,500	1.2
2017-12-04 00:00:00	175	1.16
2017-12-05 00:00:00	2,000	1.14
2017-12-06 00:00:00	225	1.125
2017-12-07 00:00:00	250	1.11

30.4.8.16. HAVING syntax

This topic describes the HAVING syntax.

The LogSearch/Analytics feature of Log Service supports the standard SQL HAVING clause. The HAVING clause is used with the GROUP BY statement to filter GROUP BY results.

Example:

```
method :PostLogstoreLogs |select avg(latency),projectName group by projectName HAVING avg(latency) > 100
```

Difference between HAVING and WHERE clauses

The HAVING clause is used to filter the aggregation and calculation results after you run the GROUP BY statement. The WHERE clause is used to filter the raw data during the aggregation calculation.

Example

Calculate the average rainfall of each province where the temperature is higher than 10°C, and only show the provinces with average rainfall greater than 100 mL in the final results:

```
* | select avg(rain) ,province where temperature > 10 group by province having avg(rain) > 100
```

30.4.8.17. ORDER BY syntax

This topic describes the ORDER BY syntax.

The ORDER BY keyword is used to sort output results. You can sort output results based on only one column.

- Syntax format

```
order by column name [desc|asc]
```

- Example

```
method :PostLogstoreLogs |select avg(latency) as avg_latency,projectName group by projectName
HAVING avg(latency) > 5700000
order by avg_latency desc
```

30.4.8.18. LIMIT syntax

The LIMIT clause is used to limit the number of returned rows.

Syntax formats

Log Service supports the following LIMIT syntax formats:

- Reads only the first N rows:

```
limit N
```

- Reads N rows starting from the S-th row:

```
limit S , N
```

Note

- If you use the LIMIT clause to paginate results, the final results rather than the intermediate results of the SQL query are obtained.
- You cannot apply the LIMIT clause to subqueries. For example, the following statement is not supported:

```
* | select count(1) from ( select distinct(url) from limit 0,1000)
```

- If you use the LIMIT clause for pagination, the offset value cannot exceed 1,000,000. For example, in the `limit S , N` clause, the sum of S and N cannot exceed 1,000,000, and the value of N cannot exceed 10,000.

Example

- To obtain the first 100 rows of results, run the following statement.

```
* | select distinct(url) from log limit 100
```

- To obtain a total of 1,000 results from row 0 to row 999, run the following statement.

```
* | select distinct(url) from log limit 0,1000
```

- To obtain a total of 1,000 results from row 1,000 to row 1,999, run the following statement:

```
* | select distinct(url) from log limit 1000,1000
```

30.4.8.19. Syntax for CASE statements and if() functions

This topic describes the syntax for CASE statements and if() functions.

CASE statements are used to classify continuous data. For example, you can use the following CASE statement to extract information from `http_user_agent` and classify the information into two types: Android and iOS.

```
SELECT
CASE
WHEN http_user_agent like '%android%' then 'android'
WHEN http_user_agent like '%ios%' then 'ios'
ELSE 'unknown' END
as http_user_agent,
count(1) as pv
group by http_user_agent
```

Examples

- Calculate the ratio of requests whose status code is 200 to all requests

```
* | SELECT
sum(
CASE
WHEN status =200 then 1
ELSE 0 end
) *1.0 / count(1) as status_200_percentage
```

- Calculate the distribution of latencies

```
* | SELECT `
CASE
WHEN latency < 10 then 's10'
WHEN latency < 100 then 's100'
WHEN latency < 1000 then 's1000'
WHEN latency < 10000 then 's10000'
else 's_large' end
as latency_slot,
count(1) as pv
group by latency_slot
```

Syntax for if() functions

An if() function works in the same way as a CASE statement does.

```
CASE
WHEN condition THEN true_value
[ ELSE false_value ]
END
```

- if(condition, true_value)

If the condition is true, the true_value column is returned. Otherwise, null is returned.
- if(condition, true_value, false_value)

If the condition is true, the true_value column is returned. Otherwise, the false_value column is returned.

Syntax for coalesce() functions

A coalesce() function returns the first non-null value from multiple columns.

```
COALESCE (value1, value2 [...])
```

Syntax for the nullif() function

If value1 is equal to value2, null is returned. Otherwise, value1 is returned.

```
nullif(value1, value2)
```

Syntax for the try() function

The try() function catches underlying exceptions, such as division by zero errors, and returns null.

```
try(expression)
```

30.4.8.20. Nested subqueries

This describes how to use nested subqueries when you query logs.

You can use nested queries to perform more complicated queries.

Nested queries differ from non-nested queries in the need for specifying the FROM clause in the SQL statement. You must specify the `from log` keyword in each SQL statement to read raw data from logs.

Example:

```
* | select sum(pv) from
(
  select count(1) as pv from log group by method
)
```

30.4.8.21. Array functions

This topic describes the syntax of array functions. It also provides examples that show how to use these functions.

Function	Description	Example
Subscript operator []	The subscript operator [] is used to obtain an element in the array.	N/A
array_distinct	Returns the distinct elements in the array.	N/A
array_intersect(x, y)	Returns the intersection of the x and y arrays.	N/A
array_union(x, y) → array	Returns the union of the x and y arrays.	N/A
array_except(x, y) → array	Returns the subtraction of the x and y arrays.	N/A

Function	Description	Example
<code>array_join(x, delimiter, null_replacement) → varchar</code>	<p>Joins string arrays with the delimiter into a string and replaces null values with <code>null_replacement</code>.</p> <p> Note The maximum size of the returned result of this <code>array_join</code> function is 1 KB. If the returned result exceeds 1 KB, the extra data will be truncated.</p>	N/A
<code>array_max(x) → x</code>	Returns the maximum value in the <code>x</code> array.	N/A
<code>array_min(x) → x</code>	Returns the minimum value in the <code>x</code> array.	N/A
<code>array_position(x, element) → bigint</code>	Returns the subscript of the element in the <code>x</code> array. The subscript starts from 1. The value 0 is returned if no subscript is found.	N/A
<code>array_remove(x, element) → array</code>	Removes the element from the array.	N/A
<code>array_sort(x) → array</code>	Sorts the array and moves null values to the end.	N/A
<code>cardinality(x) → bigint</code>	Returns the array size.	N/A
<code>concat(array1, array2, ..., arrayN) → array</code>	Concatenates arrays.	N/A
<code>contains(x, element) → boolean</code>	Returns true if the <code>x</code> array contains the specified element.	N/A
<code>filter(array, function) → array</code>	For more information about this Lambda function, see the <code>filter()</code> function in Lambda functions .	N/A
<code>flatten(x) → array</code>	Flattens an <code>array(array(T))</code> to an <code>array(T)</code> by concatenating the contained arrays.	N/A
<code>reduce(array, initialState, inputFunction, outputFunction) → x</code>	For more information, see the <code>reduce()</code> function in Lambda functions .	N/A
<code>reverse(x) → array</code>	Returns an array that has the reversed order of the <code>x</code> array.	N/A
<code>sequence(start, stop) → array</code>	Generates a sequence of elements from <code>start</code> to <code>stop</code> . The difference between elements is 1.	N/A
<code>sequence(start, stop, step) → array</code>	Generates a sequence of elements from <code>start</code> to <code>stop</code> . The difference between elements is <code>step</code> .	N/A

Function	Description	Example
<code>sequence(start, stop, step) → array</code>	Generates a sequence of timestamps from start to stop. The difference between timestamps is step. The type of step can be either INTERVAL DAY TO SECOND or INTERVAL YEAR TO MONTH.	N/A
<code>shuffle(x) → array</code>	Shuffles the array.	N/A
<code>slice(x, start, length) → array</code>	Returns a subset of the x array starting from the start value with the specified length.	N/A
<code>transform(array, function) → array</code>	For more information, see the <code>transform()</code> function in Lambda functions .	N/A
<code>zip(array1, array2[, ...]) → array</code>	Merges the specified arrays. The M-th element of the N-th argument will be the N-th field of the M-th output element.	<pre>SELECT zip(ARRAY[1, 2], ARRAY['1b', null, '3b']); -- [ROW(1, '1b'), ROW(2, null), ROW(null, '3b')]</pre>
<code>zip_with(array1, array2, function) → array</code>	For more information, see the <code>zip_with()</code> function in Lambda functions .	N/A
<code>array_agg (key)</code>	An aggregate function that returns an array from values in the key column.	<pre>* select array_agg(key)</pre>
<code>array_transpose(array[array[x,y,z], array[a,b,c]])</code>	Returns a new matrix by transposing the values of the previous matrix from rows to columns.	N/A

30.4.8.22. Binary string functions

This topic describes the syntax of binary string functions. It also provides examples that show how to use these functions.

Data of the VARBINARY type is different from data of the VARCHAR type.

Function	Description
Concatenation operator ()	The result of <code>a b</code> is <code>ab</code> .
<code>length(binary) → bigint</code>	Returns the length in binary.
<code>concat(binary1, ..., binaryN) → varbinary</code>	Connects the binary strings, which is equivalent to .
<code>to_base64(binary) → varchar</code>	Converts a binary string to a Base64 string.
<code>from_base64(string) → varbinary</code>	Converts a Base64 string to a binary string.
<code>to_base64url(binary) → varchar</code>	Converts a string to a URL-safe Base64 string.
<code>from_base64url(string) → varbinary</code>	Converts a URL-safe Base64 string to a binary string.
<code>to_hex(binary) → varchar</code>	Converts a binary string to a hexadecimal string.

Function	Description
<code>from_hex(string) → varbinary</code>	Converts a hexadecimal string to a binary string.
<code>to_big_endian_64(bigint) → varbinary</code>	Convert a number to a binary string in big endian mode.
<code>from_big_endian_64(binary) → bigint</code>	Converts a binary string in big endian mode to a number.
<code>md5(binary) → varbinary</code>	Calculates the MD5 value of a binary string.
<code>sha1(binary) → varbinary</code>	Calculates the SHA1 value of a binary string.
<code>sha256(binary) → varbinary</code>	Calculates the SHA256 hash value of a binary string.
<code>sha512(binary) → varbinary</code>	Calculate the SHA512 value of a binary string.
<code>xxhash64(binary) → varbinary</code>	Calculates the xxhash64 value of a binary string.

30.4.8.23. Bitwise operations

This topic describes the syntax for bitwise operations. It also provides examples that show how to use these operations.

Function	Description	Example
<code>bit_count(x, bits) → bigint</code>	Count the number of 1 in the binary expression of x.	<pre>SELECT bit_count(9, 64); — 2</pre> <pre>SELECT bit_count(9, 8); — 2</pre> <pre>SELECT bit_count(-7, 64); — 62</pre> <pre>SELECT bit_count(-7, 8); — 6</pre>
<code>bitwise_and(x, y) → bigint</code>	Perform the AND operation on x and y in the binary form.	N/A
<code>bitwise_not(x) → bigint</code>	Calculate the opposite values of all bits of x in the binary form.	N/A
<code>bitwise_or(x, y) → bigint</code>	Perform the OR operation on x and y in the binary form.	N/A
<code>bitwise_xor(x, y) → bigint</code>	Perform the XOR operation on x and y in the binary form.	N/A

30.4.8.24. Interval-valued comparison and periodicity-valued comparison functions

Log Service allows you to use interval-valued comparison and periodicity-valued comparison functions to query and analyze log data.

You can use the functions to compare the value for one period with that for a previous period.

Function	Description	Example
----------	-------------	---------

Function	Description	Example
<pre>compare(value, time_window)</pre>	<p>This function compares the value calculated for the current period with that calculated for the period before time_window.</p> <p>The data type of the values to be compared is Double or Long. The time_window parameter is measured in seconds. This function returns an array.</p> <p>Possible return values include the value for the current period, the value for the period before time_window, and the ratio of the current value to the value before time_window.</p>	<pre>* select compare(pv , 86400) from (select count(1) as pv from log)</pre>
<pre>compare(value, time_window1, time_window2)</pre>	<p>This function compares the current value with the values for periods before time_window1 and time_window2. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before time_window1, value before time_window2, current value/value before time_window1, current value/value before time_window2].</p>	<pre>* select compare(pv, 86400, 172800) from (select count(1) as pv from log)</pre>
<pre>compare(value, time_window1, time_window2, time_window3)</pre>	<p>This function compares the value for the current period with the values for periods before time_window1, time_window2, and time_window3. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before time_window1, value before time_window2, value before time_window3, current value/value before time_window1, current value/value before time_window2, current value/value before time_window3].</p>	<pre>* select compare(pv, 86400, 172800,604800) from (select count(1) as pv from log)</pre>

Function	Description	Example
<code>ts_compare(value, time_window)</code>	<p>This function compares the value for the current period with the values for periods before <code>time_window1</code> and <code>time_window2</code> and returns a JSON array. The comparison results are in the JSON array format, where the values must be returned in the following order: [current value, value before <code>time_window1</code>, current value/value before <code>time_window1</code>, Unix timestamp that indicates the start time of the previous period].</p> <p>This function is used to compare time series values. You must specify the <code>GROUP BY</code> keyword for the time column in SQL statements.</p>	<p>For example, <code>* select t, ts_compare(pv, 86400) as d from(select date_trunc('minute',__time__) as t, count(1) as pv from log group by t order by t) group by t</code> specifies that the function compares the calculation result of every minute in the current period with that of every minute in the last period.</p> <p>The comparison result is <code>d: [1251.0,1264.0, 0.9897151898734177, 1539843780.0,1539757380.0]t:2018-10-19 14:23:00.000</code> .</p>

Examples

- Calculate the ratio of the PV for an hour on a day to that for the same time period on a previous day. The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

Statement for query and analysis:

```
* | select compare( pv , 86400) from (select count(1) as pv from log)
```

In the preceding statement, 86400 indicates 86,400 seconds before the current period.

Return results:

```
[9.0,19.0,0.47368421052631579]
```

In the preceding results,

- 9.0 is the PV for the period from 2018-07-25 14:00:00 to 2018-07-25 15:00:00.
- 19.0 is the PV for the period from 2018-07-24 14:00:00 to 2018-07-24 15:00:00.
- 0.47368421052631579 is the ratio of the PV for the current period to that for a previous period.

If you want to expand the array into three columns of numbers, the analysis statement is

```
* | select diff[1],diff[2],diff[3] from(select compare( pv , 86400) as diff from (select count(1) as pv from log))
```

- Calculate the PV ratio for every minute of the current hour to that in the same time period as the day before, and output the results in a line chart.

- Calculate the PV ratio for every minute of the current hour to that in the same time period as the day before. The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

Statement for query and analysis:

```
* | select t, compare( pv , 86400) as diff from (select count(1) as pv, date_format(from_unixtime(__time__), '%H:%i') as t from log group by t) group by t order by t
```

Return results:

t	diff
14:00	[9520.0,7606.0,1.2516434393899554]

t	diff
14:01	[8596.0,8553.0,1.0050274757395066]
14:02	[8722.0,8435.0,1.0340248962655603]
14:03	[7499.0,5912.0,1.2684370771312586]

In the preceding table, t indicates the time in the format of Hour:Minute . The contents of the diff column are included in an array that contains the following values:

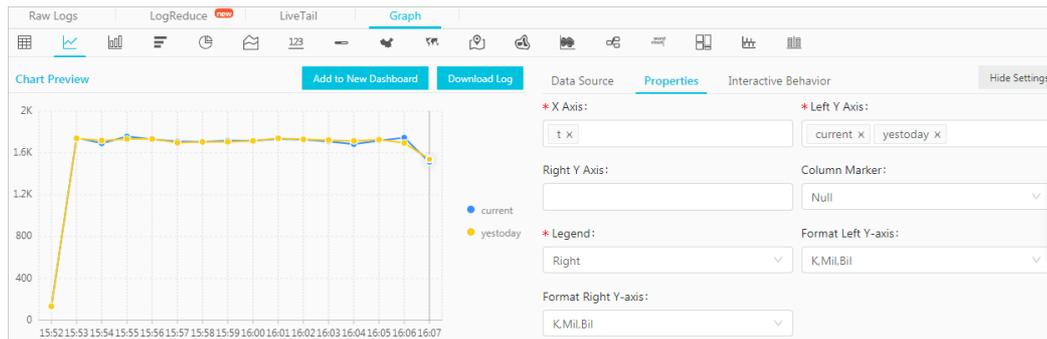
- The PV value of the current period.
- The PV value of the previous period.
- The ratio of the PV value for the current period to that for the previous period.

ii. To show the query results in a line chart, use the following statement:

```
*|select t, diff[1] as current, diff[2] as yestoday, diff[3] as percentage from(select t, compare( pv , 86400) as diff from (select count(1) as pv, date_format(from_unixtime(__time__), '%H:%i') as t from log group by t) group by t order by t)
```

The two lines indicate the PV values of a day and the day before.

Line chart



30.4.8.25. Comparison functions and operators

This topic describes the comparison functions and operators in Log Service. You can use these functions and operators to query and analyze log data.

Comparison functions and operators

A comparison function compares two parameter values of any comparable data types, such as INTEGER, BIGINT, DOUBLE, and TEXT.

Comparison operators

A comparison operator is used to compare two values. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to

Operator	Description
>=	Greater than or equal to
=	Equal to
<>	Not equal to
!=	Not equal to

Range operator BETWEEN

The BETWEEN operator determines whether a value falls in a specified closed interval.

- If the value falls in the specified closed interval, TRUE is returned. Otherwise, FALSE is returned.

Example: `SELECT 3 BETWEEN 2 AND 6;` . The statement is true, and TRUE is returned.

The preceding statement is equivalent to `SELECT 3 >= 2 AND 3 <= 6;` .

- The BETWEEN operator can be put behind the NOT operator to test whether a value falls out of a specified closed interval.

Example: `SELECT 3 NOT BETWEEN 2 AND 6;` . The statement is false, and FALSE is returned.

The preceding statement is equivalent to `SELECT 3 < 2 OR 3 > 6;` .

- If any of the three values is NULL, NULL is returned.

IS NULL and IS NOT NULL

The IS NULL and IS NOT NULL operators test whether a value is NULL.

IS DISTINCT FROM and IS NOT DISTINCT FROM

These operators are similar to the EQUAL TO and NOT EQUAL TO operators. However, IS DISTINCT FROM and IS NOT DISTINCT FROM can determine whether a NULL value exists.

Examples:

```
SELECT NULL IS DISTINCT FROM NULL; -- false
SELECT NULL IS NOT DISTINCT FROM NULL; -- true
```

The DISTINCT operator compares parameter values under multiple conditions, as described in the following table.

a	b	a = b	a <> b	a DISTINCT b	a NOT DISTINCT b
1	1	TRUE	FALSE	FALSE	TRUE
1	2	FALSE	TRUE	TRUE	FALSE
1	NULL	NULL	NULL	TRUE	FALSE
NULL	NULL	NULL	NULL	FALSE	TRUE

GREATEST and LEAST

These operators are used to obtain the maximum or minimum value from a row of field values.

Example:

```
select greatest(1,2,3) -- Returns 3.
```

Quantified comparison predicates: ALL, ANY, and SOME

The ALL, ANY, and SOME quantifiers can be used to determine whether a parameter value meets specified conditions.

- ALL is used to determine whether a parameter value meets all conditions. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.
- ANY is used to determine whether a parameter value meets a condition. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.
- SOME is used to determine whether a parameter value meets a condition. SOME is equivalent to ANY.
- ALL, ANY, and SOME must immediately follow comparison operators.

ALL and ANY support comparison under multiple conditions, as described in the following table.

Expression	Description
A = ALL (...)	Returns TRUE if A matches all values.
A <> ALL (...)	Returns TRUE if A does not match a value.
A < ALL (...)	Returns TRUE if A is smaller than the smallest value.
A = ANY (...)	Returns TRUE if A is equal to a value. This statement is equivalent to A IN (...).
A <> ANY (...)	Returns TRUE if A does not match a value.
A < ANY (...)	Returns TRUE if A is smaller than the largest value.

Examples:

```
SELECT 'hello' = ANY (VALUES 'hello', 'world'); -- true
SELECT 21 < ALL (VALUES 19, 20, 21); -- false
SELECT 42 >= SOME (SELECT 41 UNION ALL SELECT 42 UNION ALL SELECT 43); -- true
```

30.4.8.26. Lambda functions

This topic describes Lambda functions and provides some examples. You can use Lambda functions to analyze log data in Log Service

Lambda expressions

Lambda expressions use the arrow operator `->`.

Examples:

```
x -> x + 1
(x, y) -> x + y
x -> regexp_like(x, 'a+')
x -> x[1] / x[2]
x -> IF(x > 0, x, -x)
x -> COALESCE(x, 0)
x -> CAST(x AS JSON)
x -> x + TRY(1 / 0)
```

Most MySQL expressions can be used in Lambda functions.

filter(array<T>, function<T, boolean>) → ARRAY<T>

Returns an array whose elements are filtered from the specified array based on the Lambda expression.

Examples:

```
SELECT filter(ARRAY [], x -> true); -- []
SELECT filter(ARRAY [5, -6, NULL, 7], x -> x > 0); -- [5, 7]
SELECT filter(ARRAY [5, NULL, 7, NULL], x -> x IS NOT NULL); -- [5, 7]
```

map_filter(map<K, V>, function<K, V, boolean>) → MAP<K,V>

Returns a map whose elements are filtered based on the Lambda expression. The map is generated from the map function.

Examples:

```
SELECT map_filter(MAP(ARRAY[], ARRAY[]), (k, v) -> true); -- {}
SELECT map_filter(MAP(ARRAY[10, 20, 30], ARRAY['a', NULL, 'c']), (k, v) -> v IS NOT NULL); -- {10 -> a, 30 -> c}
SELECT map_filter(MAP(ARRAY['k1', 'k2', 'k3'], ARRAY[20, 3, 15]), (k, v) -> v > 10); -- {k1 -> 20, k3 -> 15}
```

reduce(array<T>, initialState S, inputFunction<S, T, S>, outputFunction<S, R>) → R

The reduce function starts from the initial state, traverses each element in the array, and then calls `inputFunction(S,T)` to generate a new state. After all the elements in the array are traversed and the final state is generated, the reduce function calls `outputFunction` to assign the final state value to the result `R` and output the result. The procedure is described as follows:

1. Start from the initial state `S`.
2. Traverse each element `T`.
3. Calculate `inputFunction(S,T)` to generate a new state `S`.
4. Repeat steps 2 and 3 until the last element is traversed and has a new state.
5. Turn the final state `S` into the final result `R`.

Examples:

```
SELECT reduce(ARRAY [], 0, (s, x) -> s + x, s -> s); -- 0
SELECT reduce(ARRAY [5, 20, 50], 0, (s, x) -> s + x, s -> s); -- 75
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> s + x, s -> s); -- NULL
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> s + COALESCE(x, 0), s -> s); -- 75
SELECT reduce(ARRAY [5, 20, NULL, 50], 0, (s, x) -> IF(x IS NULL, s, s + x), s -> s); -- 75
SELECT reduce(ARRAY [2147483647, 1], CAST (0 AS BIGINT), (s, x) -> s + x, s -> s); -- 2147483648
SELECT reduce(ARRAY [5, 6, 10, 20], -- calculates arithmetic average: 10.25
  CAST(ROW(0.0, 0) AS ROW(sum DOUBLE, count INTEGER)),
  (s, x) -> CAST(ROW(x + s.sum, s.count + 1) AS ROW(sum DOUBLE, count INTEGER)),
  s -> IF(s.count = 0, NULL, s.sum / s.count));
```

transform(array<T>, function<T, U>) → ARRAY<U>

This Lambda function traverses each element in an array to generate a new result `U`.

Examples:

```
SELECT transform(ARRAY [], x -> x + 1); -- []
SELECT transform(ARRAY [5, 6], x -> x + 1); -- [6, 7] -- Increments each element by 1.
SELECT transform(ARRAY [5, NULL, 6], x -> COALESCE(x, 0) + 1); -- [6, 1, 7]
SELECT transform(ARRAY ['x', 'abc', 'z'], x -> x || '0'); -- ['x0', 'abc0', 'z0']
SELECT transform(ARRAY [ARRAY [1, NULL, 2], ARRAY[3, NULL]], a -> filter(a, x -> x IS NOT NULL)); -- [[1, 2], [3]]
```

zip_with(array<T>, array<U>, function<T, U, R>) → array<R>

This Lambda function merges two arrays and generates the element R in the new array based on element T and element U.

Examples:

```
SELECT zip_with(ARRAY[1, 3, 5], ARRAY['a', 'b', 'c'], (x, y) -> (y, x)) --Transposes the elements of the two arrays to gener
ate a new array. Result: [['a', 1], ['b', 3],['c', 5]]
SELECT zip_with(ARRAY[1, 2], ARRAY[3, 4], (x, y) -> x + y); -- Result: [4, 6]
SELECT zip_with(ARRAY['a', 'b', 'c'], ARRAY['d', 'e', 'f'], (x, y) -> concat(x, y)) -- Concatenates the elements of the two arra
ys to generate a new string. Result: ['ad', 'be', 'cf']
```

30.4.8.27. Logical functions

This topic describes the available logical functions in Log Service. You can use these functions to query and analyze log data.

Logical operators

Operator	Description	Example
AND	The result is TRUE if both values are TRUE.	a AND b
OR	The result is TRUE if either value is TRUE.	a OR b
NOT	The result is TRUE if the value is FALSE.	NOT a

Effect of NULL on logical operators

The following tables list the truth values when the values of a and b are TRUE, FALSE, and NULL, respectively.

Truth table 1

a	b	a AND b	a OR b
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL

a	b	a AND b	a OR b
NULL	TRUE	NULL	TRUE
NULL	FALSE	FALSE	NULL
NULL	NULL	NULL	NULL

Truth table 2

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

30.4.8.28. Field aliases

This topic describes how to specify an alias for a field and provides some examples.

A field name in an SQL statement must start with letters and contain digits and underscores (_).

If you have configured a field name that does not conform to the SQL standard (such as `User-Agent`), you must specify an alias for the field on the field index configuration page. The alias takes effect only for the duration of the SQL statement. The data is still stored under the original field name. You must specify the original name when you perform a search.

You can also specify an alias for a field in an SQL statement if the original name is long.

Sample aliases

Original field name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

30.4.8.29. JOIN operations between Logstores and Relational

Database Service (RDS) tables

This topic describes how to join Logstores in Log Service with RDS tables for queries and store the query results in RDS tables.

Procedure

1. Create a VPC. Create an RDS instance and specify the VPC to host the RDS instance. Then the VPC ID and the RDS instance ID are obtained.
2. Configure a whitelist for the RDS instance. Add the following CIDR blocks to the whitelist: `100.104.0.0/16` , `11.194.0.0/16` , and `11.201.0.0/16`
3. Create an external store Run the following statement to create an external store. Replace the parameter values based on your business needs.

```
{
  "externalStoreName": "storeName",
  "storeType": "rds-vpc",
  "parameter": {
    {
      "region": "cn-qingdao",
      "vpc-id": "vpc-m5eq4irc1pucp*****",
      "instance-id": "i-m5eeo2whsn*****",
      "host": "localhost",
      "port": "3306",
      "username": "root",
      "password": "*****",
      "db": "scmc",
      "table": "join_meta"
    }
  }
}
```

Parameters

Parameter	Description
region	The region where your RDS instance resides.
vpc-id	The ID of the VPC where your RDS instance resides.
instance-id	The ID of the RDS instance.
host	The ID of the ECS instance that is used to access the RDS instance.
port	The port of the ECS instance that is used to access the RDS instance.
username	The username that is used to log on to the RDS instance.
password	The password that is used to log on to the RDS instance.
db	The name of the database.
table	The name of the table with which the Logstore is joined.

 **Note** You can join a Logstore with an RDS table that resides only in the China (Beijing), China (Qingdao), and China (Hangzhou) regions.

4. JOIN query. Log on to the Log Service console. In the **Search & Analyze** search box, run a JOIN statement.

Supported JOIN syntax:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

```
[ INNER ] JOIN
LEFT [ OUTER ] JOIN
RIGHT [ OUTER ] JOIN
FULL [ OUTER ] JOIN
```

 **Note**

- You can join Logstores only to external tables.
- In the JOIN statement, you must first specify a Logstore before specifying an external store.
- You must specify the name of the external store instead of the name of an RDS table. The external store name automatically changes into the combination of the RDS database name and the name of the RDS table that you want to join with the Logstore.

Sample JOIN statement:

```
method:postlogstorelogs | select count(1) , histogram(logstore) from log l join join_meta m on l.projectid = cast(
m.ikey as varchar)
```

5. Store the query results to the RDS table. You can use the INSERT statement to insert the query results into the RDS table.

```
method:postlogstorelogs | insert into method_output select cast(methodasvarchar(65535)),count(1)fromloggroup
bymethod
```

Sample Python script:

```
# encoding: utf-8
from __future__ import print_function
from aliyun.log import *
from aliyun.log.util import base64_encodestring
from random import randint
import time
import os
from datetime import datetime

endpoint = os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', 'cn-chengdu.log.aliyuncs.com')
accessKeyId = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', '')
accessKey = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', '')
logstore = os.environ.get('ALIYUN_LOG_SAMPLE_LOGSTORE', '')
project = "ali-yunlei-chengdu"
client = LogClient(endpoint, accessKeyId, accessKey, token)

### Create an external store
res = client.create_external_store(project, ExternalStoreConfig("rds_store", "region", "rds-vpc", "vpc id", "instance-id", "instance-ip", "port", "username", "password", "db", "table"));
res.log_print()

### Obtain external store details
res = client.get_external_store(project, "rds_store");
res.log_print()

res = client.list_external_store(project, "");
res.log_print();

# Perform the JOIN operation.
req = GetLogStoreLogsRequest(project, logstore, From, To, "", "select count(1) from "+ logstore + " s join meta m on s.projectid = cast(m.ikey as varchar)");
res = client.get_logs(req)
res.log_print();

# Store query results to the RDS table
req = GetLogStoreLogsRequest(project, logstore, From, To, "", "insert into rds_store select count(1) from "+ logstore );
res = client.get_logs(req)
res.log_print();
```

30.4.8.30. Geospatial functions

This topic describes the available geospatial functions in Log Service. You can use these functions to query and analyze log data.

Concept of geometry

Geospatial functions support geometries in the well-known text (WKT) format.

Geometry formats

Geometry	WKT format
Point	POINT (0 0)
LineString	LINestring (0 0, 1 1, 1 2)

Geometry	WKT format
Polygon	<code>POLYGON ((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1))</code>
MultiPoint	<code>MULTIPOINT (0 0, 1 2)</code>
MultiLineString	<code>MULTILINESTRING ((0 0, 1 1, 1 2), (2 3, 3 2, 5 4))</code>
MultiPolygon	<code>MULTIPOLYGON (((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1)), ((-1 -1, -1 -2,</code>
GeometryCollection	<code>GEOMETRYCOLLECTION (POINT(2 3), LINESTRING (2 3, 3 4))</code>

Constructors

Constructor description

Function	Description
<code>ST_Point(double, double) → Point</code>	Returns a geometry point instance with the specified coordinate values.
<code>ST_LineFromText(varchar) → LineString</code>	Returns a geometry LineString instance from a WKT representation.
<code>ST_Polygon(varchar) → Polygon</code>	Returns a geometry polygon instance from a WKT representation.
<code>ST_GeometryFromText(varchar) → Geometry</code>	Returns a geometry instance from a WKT representation.
<code>ST_AsText(Geometry) → varchar</code>	Returns the WKT representation of a geometry.

Operations

Function	Description
<code>ST_Boundary(Geometry) → Geometry</code>	Returns the closure of the combinatorial boundary of a geometry.
<code>ST_Buffer(Geometry, distance) → Geometry</code>	Returns the geometry that represents all points whose distance from the specified geometry is shorter than or equal to the specified distance.
<code>ST_Difference(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set difference of the specified geometries.
<code>ST_Envelope(Geometry) → Geometry</code>	Returns the bounding rectangular polygon of a geometry.
<code>ST_ExteriorRing(Geometry) → Geometry</code>	Returns a line string that represents the exterior ring of the input polygon.
<code>ST_Intersection(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set intersection of two geometries.
<code>ST_SymDifference(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set symmetric difference of two geometries.

Relationship tests

Function	Description
<code>ST_Contains(Geometry, Geometry) → boolean</code>	Returns True if and only if no points of the second geometry lie in the exterior of the first geometry, and at least one point of the interior of the first geometry lies in the interior of the second geometry. Returns False if points of the second geometry are on the boundary of the first geometry.
<code>ST_Crosses(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries share some, but not all, interior points in common.
<code>ST_Disjoint(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries do not spatially intersect.
<code>ST_Equals(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries represent the same geometry.
<code>ST_Intersects(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries spatially intersect in two dimensions.
<code>ST_Overlaps(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries share space in the same dimension, but are not completely contained by each other.
<code>ST_Relate(Geometry, Geometry) → boolean</code>	Returns True if the first geometry is spatially related to the second geometry.
<code>ST_Touches(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries have at least one point in common, but their interiors do not intersect.
<code>ST_Within(Geometry, Geometry) → boolean</code>	Returns True if the first geometry is completely inside the second geometry. Returns False if the two geometries have points in common at the boundaries.

Accessors

Function	Description
<code>ST_Area(Geometry) → double</code>	Returns the two-dimensional Euclidean area of a geometry.
<code>ST_Centroid(Geometry) → Geometry</code>	Returns the point value that is the mathematical centroid of a geometry.
<code>ST_CoordDim(Geometry) → bigint</code>	Returns the coordinate dimension of a geometry.
<code>ST_Dimension(Geometry) → bigint</code>	Returns the inherent dimension of a geometry object, which must be less than or equal to the coordinate dimension.
<code>ST_Distance(Geometry, Geometry) → double</code>	Returns the minimum two-dimensional Cartesian distance (based on spatial ref) between two geometries in projected units.
<code>ST_IsClosed(Geometry) → boolean</code>	Returns True if the start and end points of the linestring are coincident.

Function	Description
ST_IsEmpty(Geometry) → boolean	Returns True if the specified geometry is an empty geometry, such as geometry collection, polygon, and point.
ST_IsRing(Geometry) → boolean	Returns True if and only if the line is closed and simple.
ST_Length(Geometry) → double	Returns the length of a LineString or multi-LineString by using Euclidean measurement on a two-dimensional plane (based on spatial ref) in projected units.
ST_XMax(Geometry) → double	Returns the X maximum of the bounding box of the geometry.
ST_YMax(Geometry) → double	Returns the Y maximum of the bounding box of the geometry.
T_XMin(Geometry) → double	Returns the X minimum of the bounding box of the geometry.
ST_YMin(Geometry) → double	Returns the Y minimum of the bounding box of the geometry.
ST_StartPoint(Geometry) → point	Returns the first point of a geometry LineString instance.
ST_EndPoint(Geometry) → point	Returns the last point of a geometry LineString instance.
ST_X(Point) → double	Returns the X coordinate of a point.
ST_Y(Point) → double	Returns the Y coordinate of a point.
ST_NumPoints(Geometry) → bigint	Returns the number of points in a geometry.
ST_NumInteriorRing(Geometry) → bigint	Returns the cardinality of the collection of interior rings of a polygon.

30.4.8.31. Geography functions

This topic describes the syntax of geography functions and provides some examples.

For information about functions that identify the country, province, city, ISP, and the longitude and latitude of a specified IP address, see [IP functions](#).

Geography functions

Function	Description	Example
geohash(string)	Returns the geohash value of the specified geographical location. The geographical location is represented by a string in the format of "latitude, longitude". The values for latitude and longitude are separated by a comma.	<pre>select geohash('34.1,120.6')= 'wwjcbdrnzs'</pre>
geohash(lat,lon)	Returns the geohash value of the specified geographical location. The geographical location is represented by two parameters that indicate the latitude and longitude.	<pre>select geohash(34.1,120.6)= 'wwjcbdrnzs'</pre>

30.4.8.32. JOIN syntax

The JOIN operation joins multiple tables by using one or more fields in the tables. You can join a Logstore created in Log Service with the Logstore itself, with another Logstore, or with an RDS table. This topic describes how to join different Logstores.

Procedure

1. Download the [latest version of the SDK for Python](#).
2. Call the GetProjectLogs operation to query logs.

Sample SDK

```
#!/usr/bin/env python
#encoding: utf-8
import time,sys,os
from aliyun.log.logexception import LogException
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.getlogsrequest import GetProjectLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listtopicsrequest import ListTopicsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
from aliyun.log.index_config import *
from aliyun.log.logtail_config_detail import *
from aliyun.log.machine_group_detail import *
from aliyun.log.acl_config import *
if __name__=='__main__':
    token = None
    endpoint = "http://cn-hangzhou.log.aliyuncs.com"
    accessKeyId = 'LTAIvKy7U'
    accessKey='6gXLNTLyCdfsfwrewrfhdkfdfsuiwu'
    client = LogClient(endpoint, accessKeyId, accessKey,token)
    logstore = "meta"
    # In the query statements, specify two Logstores, the query time ranges of both Logstores, and the key that you want to use to join the two Logstores.
    req = GetProjectLogsRequest(project,"select count(1) from sls_operation_log s join meta m on s.__date__>'2018-04-10 00:00:00' and s.__date__ < '2018-04-11 00:00:00' and m.__date__ >'2018-04-23 00:00:00' and m.__date__ <'2018-04-24 00:00:00' and s.projectid = cast(m.ikey as varchar)");
    res = client.get_project_logs(req)
    res.log_print();
    exit(0)
```

 **Note** For more information about the JOIN syntax and examples, see [Join](#).

30.4.8.33. UNNEST function

This topic describes the UNNEST function.

Scenarios

Log data is typically stored as primitive data types, such as string or number. In certain scenarios, log data may include complex data types, such as arrays, maps, and JSON objects. The UNNEST function can be used to transform complex data types into rows of primitive data types. This simplifies query and analysis.

Example:

```
__source__: 1.1.1.1
__tag__:__hostname__: vm-req-170103232316569850-tianchi111932.tc
__topic__: TestTopic_4
array_column: [1,2,3]
double_column: 1.23
map_column: {"a":1,"b":2}
text_column: Product
```

The values of the `array_column` field are arrays. To obtain the sum of elements of all `array_column` field values, you must traverse all elements of every array.

UNNEST function

Syntax	Description
<code>unnest(array) as table_alias(column_name)</code>	Expands an array into multiple rows. The column name of these rows is <code>column_name</code> .
<code>unnest(map) as table(key_name, value_name)</code>	Expands a map into multiple rows. <code>key_name</code> specifies the column name of the keys, and <code>value_name</code> specifies the column name of the values.

Note The UNNEST function is used to expand arrays or maps. If you want to expand a string, you must transform the string into a JSON object, and then convert the JSON object into an array or map. To do this, you can use the `cast(json_parse(array_column) as array(bigint))` function.

Traverse every element of an array

Expands an array into multiple rows by using the following SQL SELECT statement:

```
*| select array_column, a from log, unnest( cast( json_parse(array_column) as array(bigint) ) ) as t(a)
```

The UNNEST function `unnest(cast(json_parse(array_column) as array(bigint))) as t(a)` expands the array into multiple rows. The rows are stored in a derived table referenced as `t`, with the column referenced as `a`.

- Calculate the sum of the elements in an array:

```
*| select sum(a) from log, unnest( cast( json_parse(array_column) as array(bigint) ) ) as t(a)
```

- Perform a GROUP BY operation on all elements of an array:

```
*| select a, count(1) from log, unnest( cast( json_parse(array_column) as array(bigint) ) ) as t(a) group by a
```

Traverse every key and value of a map

- Traverse every key and value of a map:

```
* | select map_column , a,b  from log, unnest( cast( json_parse(map_column)  as map(varchar, bigint) ) ) as t(a,b)
```

- Perform a GROUP BY operation on all keys of a map:

```
* | select key, sum(value)  from log, unnest( cast( json_parse(map_column)  as map(varchar, bigint) ) ) as t(key,v  
alue)  GROUP BY key
```

Visualize the query results of the histogram and numeric_histogram functions.

- histogram

The histogram function works in a similar manner to the count group by syntax. For more information about the histogram function, see [Map functions](#).

In most cases, the histogram function returns a JSON object. The following is an example:

```
* | select histogram(method)
```

You can use the UNNEST function to expand JSON data into multiple rows. Then the data can be visualized. The following is an example:

```
* | select key , value  from( select histogram(method) as his from log) , unnest(his ) as t(key,value)
```

- numeric_histogram

The numeric_histogram function assigns a column of numeric values into multiple bins. This function is equivalent to a GROUP BY operation that is performed on a numeric value column. For more information about the syntax of the numeric_histogram function, see [Approximate functions](#).

```
* | select numeric_histogram(10,Latency)
```

Use the following SELECT statement to visualize the result:

```
* | select key,value from(select numeric_histogram(10,Latency) as his from log) , unnest(his) as t(key,value)
```

30.4.9. Machine learning syntax and functions

30.4.9.1. Overview

The machine learning feature of Log Service supports multiple algorithms and calling methods. You can use SELECT statements and machine learning functions to analyze the characteristics of a field or fields within a period of time.

Log Service offers multiple time series analysis algorithms to help you implement time series prediction, time series anomaly detection, time series decomposition, and multi-time series clustering. The algorithms are compatible with standard SQL statements. This greatly simplifies the use of the algorithms and improves the troubleshooting efficiency.

Features

- Supports various smooth operations on single-time series data.
- Supports algorithms related to the prediction, anomaly detection, change point detection, inflection point detection, and multi-period estimation of single-time series data.
- Supports decomposition operations on single-time series data.
- Supports various clustering algorithms of multi-time series data.
- Supports multi-field pattern mining (based on the sequence of numeric data or text).

Limits

- The specified time series data must be sampled based on the same interval.
- The specified time series data cannot contain data repeatedly sampled from the same time point.

Item	Description
Processing capacity of time-series data	Data can be collected from a maximum of 150,000 consecutive time points. If the data volume exceeds the processing capacity, you must aggregate the data or reduce the sampling amount.
Clustering capacity of the density-based clustering algorithm	A maximum of 5,000 time series curves, each of which cannot contain more than 1,440 time points.
Clustering capacity of the hierarchical clustering algorithm	A maximum of 2,000 time series curves, each of which cannot contain more than 1,440 time points.

Machine learning functions

Type	Function	Description
Smooth functions	ts_smooth_simple	Uses the Holt Winters algorithm to smooth time series data.
	ts_smooth_fir	Uses the finite impulse response (FIR) filter to smooth time series data.
	ts_smooth_iir	Uses the infinite impulse response (IIR) filter to smooth time series data.
Multi-period estimation functions	ts_period_detect	Forecasts time series data by period.
Change point detection functions	ts_cp_detect	Finds intervals with different statistical characteristics from time series data. The interval endpoints are change points.
	ts_breakout_detect	Finds the time points when statistics steeply increases or decreases from time series data.
Maximum value detection function	ts_find_peaks	Finds the local maximum value of time series data in a specified window.
Prediction and anomaly detection functions	ts_predicate_simple	Uses default parameters to model time series data and performs simple time series prediction and anomaly detection.
	ts_predicate_ar	Uses an autoregressive (AR) model to model time series data and performs simple time series prediction and anomaly detection.
	ts_predicate_arma	Uses an autoregressive moving average (ARMA) model to model time series data and performs simple time series prediction and anomaly detection.
	ts_predicate_arima	Uses an autoregressive integrated moving average (ARIMA) model to model time series data and performs simple time series prediction and anomaly detection.
	ts_regression_predict	Accurately predicts the trend for a single periodic time series with a certain tendency.

Type	Function	Description
Time series decomposition function	ts_decompose	Uses the Seasonal and Trend decomposition using Loess (STL) algorithm to decompose time series data.
Time series clustering functions	ts_density_cluster	Uses a density-based clustering method to cluster multiple pieces of time series data.
	ts_hierarchical_cluster	Uses a hierarchical clustering method to cluster multiple pieces of time series data.
	ts_similar_instance	Queries curves that are similar to a specified curve.
Frequent pattern statistics function	pattern_stat	Mines representative combinations of attributes among the given multi-attribute field samples to obtain the frequent pattern in statistical patterns.
Differential pattern statistics function	pattern_diff	Finds the pattern that causes differences between two collections under specified conditions.
Root cause analysis function	rca_kpi_search	When a time series metric is abnormal, you can use the root cause analysis function to analyze the dimension attributes that result in the abnormal metric in a timely manner.
Correlation analysis functions	ts_association_analysis	Quickly finds the metrics that are correlated with a specified metric among multiple observed metrics in the system.
ts_similar	Quickly finds the metrics that are correlated with specified time series data among multiple observed metrics in the system.	
Kernel density estimation function	kernel_density_estimation	Uses the smooth peak function to fit the observed data points, thus simulating the real probability distribution curve.

30.4.9.2. Smooth functions

This topic describes the smooth functions that you can use to smooth and filter specified time series curves. Filtering is the first step to discover the shapes of time series curves.

Functions

Function	Description
<code>ts_smooth_simple</code>	Uses the Holt-Winters algorithm to filter time series data. This function is the default smooth function.
<code>ts_smooth_fir</code>	Uses a finite impulse response (FIR) filter to filter time series data.
<code>ts_smooth_iir</code>	Uses an infinite impulse response (IIR) filter to filter time series data.

ts_smooth_simple

- Syntax:

```
select ts_smooth_simple(x, y)
```

- The following table lists the parameters of the function.

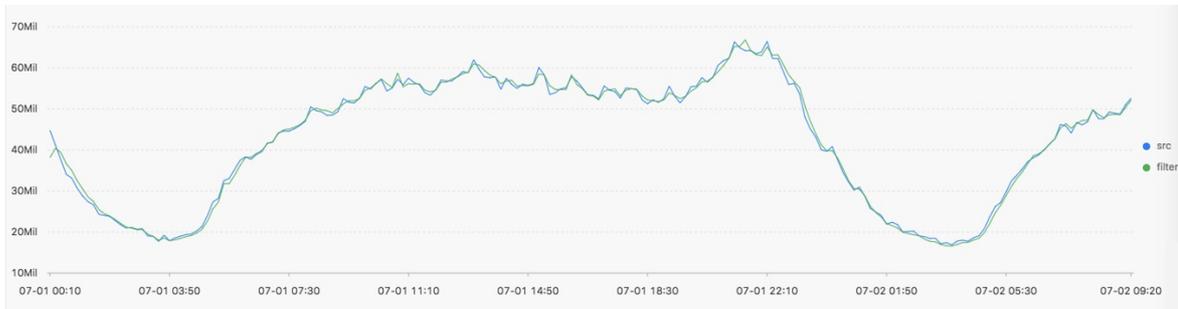
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	The Unix timestamp of the time series data. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-

- Example

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_simple(stamp, value) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from lo
g GROUP BY stamp order by stamp )
```

- The following figure shows the response.



- The following table lists the display items.

Item	Description	
Horizontal axis	unixtime	The Unix timestamp of time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

ts_smooth_fir

- Syntax:

- If you cannot determine filter parameters, use built-in window parameters in the following statement:

```
select ts_smooth_fir(x, y,winType,winSize)
```

- If you can determine filter parameters, you can specify the parameters as needed in the following statement:

```
select ts_smooth_fir(x, y,array[])
```

- The following table lists the parameters of the function.

Parameter	Description	Value
-----------	-------------	-------

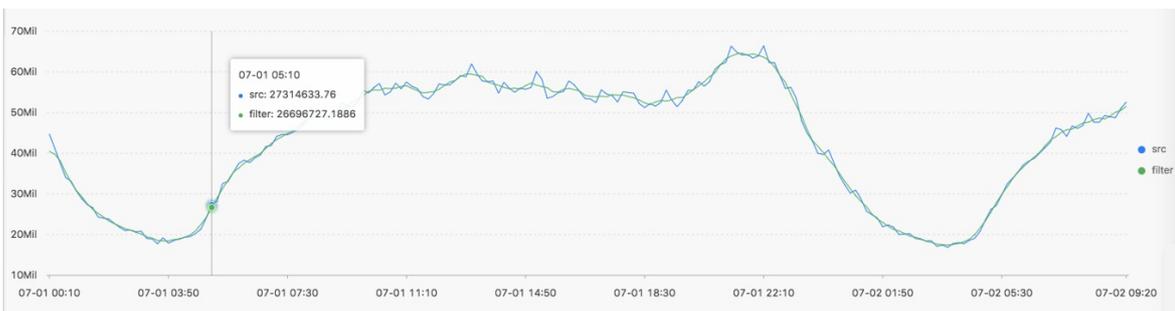
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>winType</i>	The type of window used for filtering.	Valid values: <ul style="list-style-type: none"> ○ rectangle: rectangle window. ○ hanning: hanning window ○ hamming: hamming window. ○ blackman: blackman window. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> ? Note We recommend that you select the rectangle window for better display effects. </div>
<i>winSize</i>	The length of the filtering window.	The value is of the LONG type. Valid values: 2 to 15.
<i>array[]</i>	The parameter used for FIR filtering.	The value is an array where the sum of elements is 1. For example, array[0.2, 0.4, 0.3, 0.1].

● Example 1

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_fir(stamp, value, 'rectangle', 4) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- The following figure shows the response.

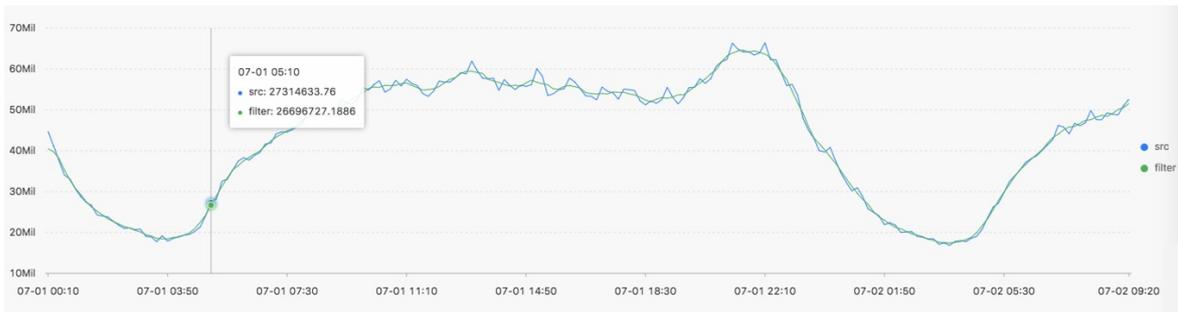


● Example 2

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_fir(stamp, value, array[0.2, 0.4, 0.3, 0.1]) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- The following figure shows the response.



- The following table lists the display items.

Item		Description
Horizontal axis	unixtime	The Unix timestamp of the time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

ts_smooth_iir

- Syntax:

```
select ts_smooth_iir(x, y, array[], array[] )
```

- The following table lists the parameters of the function.

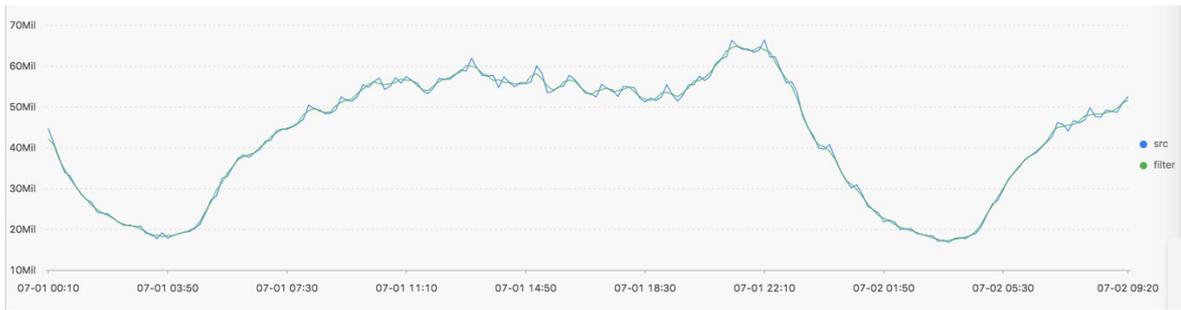
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>array[]</i>	The parameter used for IIR filtering in terms of x_i .	The value is an array where the sum of elements is 1. The length of the array ranges from 2 to 15. For example, array[0.2, 0.4, 0.3, 0.1].
<i>array[]</i>	The parameter used for IIR filtering in terms of y_{i-1} .	The value is an array where the sum of elements is 1. The length of the array ranges from 2 to 15. For example, array[0.2, 0.4, 0.3, 0.1].

- Example

- The search and analytic statement is shown as follows:

```
* | select ts_smooth_iir(stamp, value, array[0.2, 0.4, 0.3, 0.1], array[0.4, 0.3, 0.3]) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp )
```

- The following figure shows the response.



- The following table lists the display items.

Item		Description
Horizontal axis	unixtime	The Unix timestamp of the time series data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

30.4.9.3. Multi-period estimation functions

This topic describes the available multi-period estimation functions in Log Service. You can use the functions to estimate the periods of time series data in different time intervals and extract periods by using a series of operations such as Fourier transform (FT).

Functions

Function	Description
<code>ts_period_detect</code>	Estimates multiple periods of time series data.
<code>ts_period_classify</code>	Uses FT to calculate the periodicity of specified time series curves. This function can be used to identify periodic curves.

`ts_period_detect`

Syntax:

```
select ts_period_detect(x,y,minPeriod,maxPeriod)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>minPeriod</i>	The ratio of the minimum length of the estimation period to the total length of the time series data.	The parameter value must be a decimal number. Valid values: (0.0, 1.0].

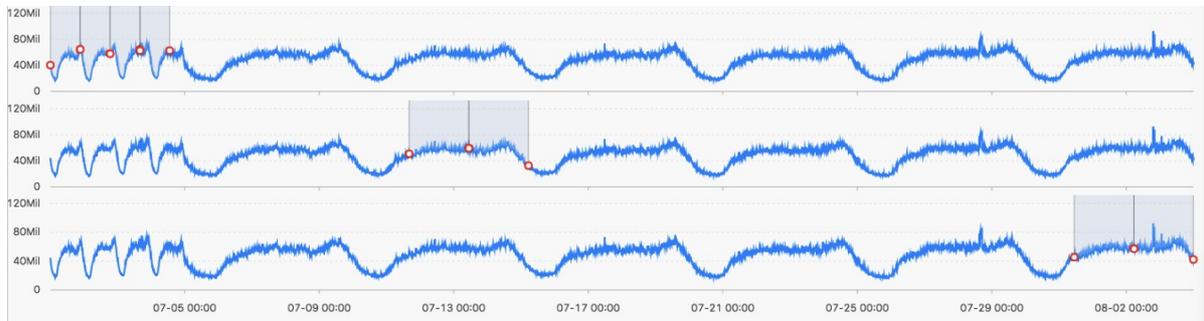
Parameter	Description	Value
<i>maxPeriod</i>	<p>The ratio of the maximum length of the estimation period to the total length of the time series data.</p> <p>Note The <i>maxPeriod</i> parameter value must be greater than the value of the <i>minPeriod</i> parameter.</p>	The parameter value must be a decimal number. Valid values: (0.0, 1.0].

Example:

- The search and analytic statement is shown as follows:

```
* | select ts_period_detect(stamp, value, 0.2, 1.0) from ( select __time__ - __time__ % 120 as stamp, avg(v) as value from m log GROUP BY stamp order by stamp )
```

- The following figure shows the response.



The following table lists the display items.

Display item	Description
<i>period_id</i>	An array with a length of 1. The element in the array indicates the sequence number of the period. The array [0] indicates the original time series curve.
<i>time_series</i>	The sequence of timestamps.
<i>data_series</i>	<p>The sequence of data at each timestamp.</p> <ul style="list-style-type: none"> If <i>period_id</i> is 0, the returned data is the original time series data. When <i>period_id</i> is not 0, the data returned is the filtered time series data.

ts_period_classify

Syntax:

```
select ts_period_classify(stamp,value,instanceName)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>stamp</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.

Parameter	Description	Value
value	The sequence of numeric data at each specified time point.	-
instanceName	The name of the time series curve.	-

Example:

- The search and analytic statement is shown as follows:

```
* and h : nu2h05202.nu8 | select ts_period_classify(stamp, value, name) from log
```

- The following figure shows the response.

line_name	prob	type
asg-2z2p9r6zf5ewg188pg5	1.0	-1.0
asg-bp1j8snc92p6v5pptgpi	0.07203668207039314	0.0
asg-wz99hse7u4ubopo5dt9o	0.0	0.0
asg-bp18oqni0gg96vy851e4	0.05590892692207093	0.0

The following table lists the display items.

Display item	Description
line_name	The name of the time series curve.
prob	The ratio of the primary period length to the length of the time series curve. Valid values: 0 to 1. You can set the value to 0.15 when you perform a test.
type	The type of the curve. Valid values: -1, -2, and 0. <ul style="list-style-type: none"> • The value 1 indicates that the time series curve length is too short (less than 64 points). • The value -2 indicates the time series curve has a high fault rate (the fault rate exceeds 20%). • The value 0 indicates the time series curve is periodic.

30.4.9.4. Change point detection functions

This topic describes the change point detection functions in Log Service. You can use the functions to detect the change points in time series data.

The change point detection functions can detect the following two kinds of change points:

- Statistics feature changes within a specified period of time
- Anomalies in time series data

Functions

Function	Description
<code>ts_cp_detect</code>	Finds intervals in which data has different statistics features. The interval endpoints are change points.
<code>ts_breakout_detect</code>	Finds the time points at which data experiences dramatic changes.

ts_cp_detect

Syntax:

- If you cannot specify an appropriate time window size, use the following syntax. The default window size used in the function is 10.

```
select ts_cp_detect(x, y, samplePeriod)
```

- To adjust the effect specific to your business environment, you can specify the minSize parameter in the following function.

```
select ts_cp_detect(x, y, minSize)
```

The following table lists the parameters of the function.

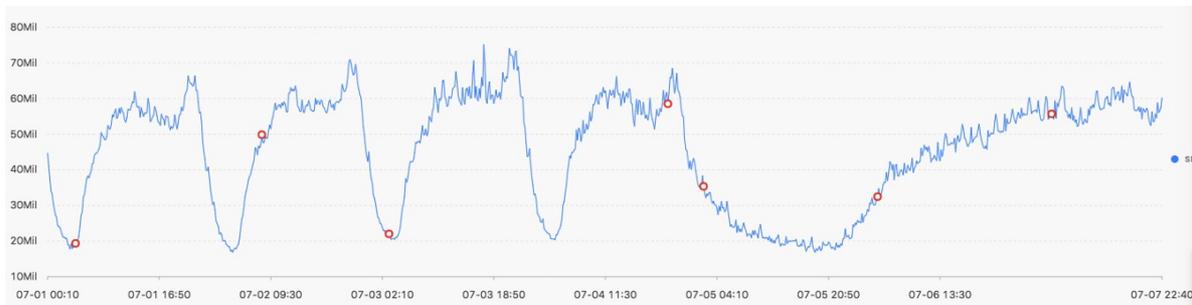
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>minSize</i>	The minimum length of time series data in a continuous interval.	The minimum value is 3 and the maximum value cannot exceed ten percent of the length of the specified time series data.

Example:

- The search and analytic statement is shown as follows:

```
* | select ts_cp_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.
	src	The unfiltered data, such as 1956092.7647745228.
Vertical axis	prob	The probability that a time point is a change point. Valid values: 0 to 1.

ts_breakout_detect

Syntax:

```
select ts_breakout_detect(x, y, winSize)
```

The following table lists the parameters of the function.

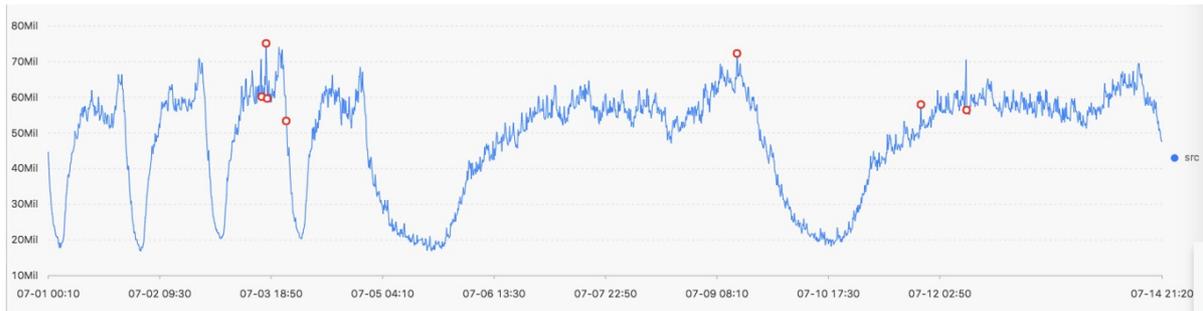
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>winSize</i>	The minimum length of time series data in a continuous interval.	The minimum value is 3 and the maximum value cannot exceed ten percent of the length of the specified time series data.

Example:

- The search and analytic statement is shown as follows:

```
* | select ts_breakout_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item	Description	
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.
Vertical axis	src	The unfiltered data, such as 1956092.7647745228.
	prob	The probability that a time point is a change point. Valid values: 0 to 1.

30.4.9.5. Maximum value detection function

This topic describes the available maximum value detection function in Log Service. You can use the functions to find the local maximum value of time series data in a specified window.

ts_find_peaks

Syntax:

```
select ts_find_peaks(x, y, winSize)
```

The following table lists the parameters of the function.

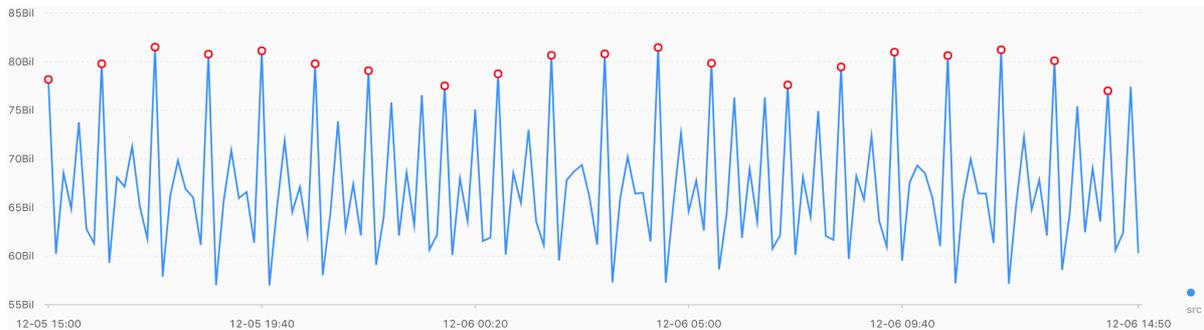
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	-
<i>winSize</i>	The minimum length of the detection window.	The value of the parameter is of the LONG type, ranging from 1 to the length of time series data. We recommend that you set this parameter to ten percent of the actual data length.

Example:

- The search and analytic statement is shown as follows:

```
* and h : nu2h05202.nu8 and m: NET | select ts_find_peaks(stamp, value, 30) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of time series data, measured in seconds, for example, 1537071480.
Vertical axis	src	The unfiltered data, such as 1956092.7647745228.
	peak_flag	Indicates whether the numeric value at the time point is the maximum value. Valid values: 1.0 and 0.0. <ul style="list-style-type: none"> • 1.0: The numeric value at the time point is the maximum value. • 0.0: The numeric value at the time point is not the maximum value.

30.4.9.6. Prediction and anomaly detection functions

Prediction and anomaly detection functions predict the trend of time series curves and identify the Ksigma and quantiles of the errors between a predicted curve and an actual curve. You can use the functions to detect anomalies.

Functions

Function	Description
<code>ts_predicate_simple</code>	Uses default parameters to model time series data and performs prediction and anomaly detection on time series data.
<code>ts_predicate_ar</code>	Uses an autoregressive (AR) model to model time series data and performs prediction and anomaly detection on time series data.
<code>ts_predicate_arma</code>	Uses an autoregressive moving average (ARMA) model to model time series data and performs prediction and anomaly detection on time series data.
<code>ts_predicate_arima</code>	Uses an autoregressive integrated moving average (ARIMA) model to model time series data and performs prediction and anomaly detection on time series data.
<code>ts_regression_predict</code>	Accurately predicts the trend for a periodic time series curve. Scenario: This function can be used to predict metering data, network traffic, financial data, and different business data that follows certain rules.
<code>ts_anomaly_filter</code>	Filters the anomalies detected from multiple time series curves based on the custom anomaly mode. The anomalies are detected during the anomaly detection. This function helps you find abnormal curves in a timely manner.

ts_predicate_simple

Syntax:

```
select ts_predicate_simple(x, y, nPred, isSmooth)
```

The following table lists the parameters of the function.

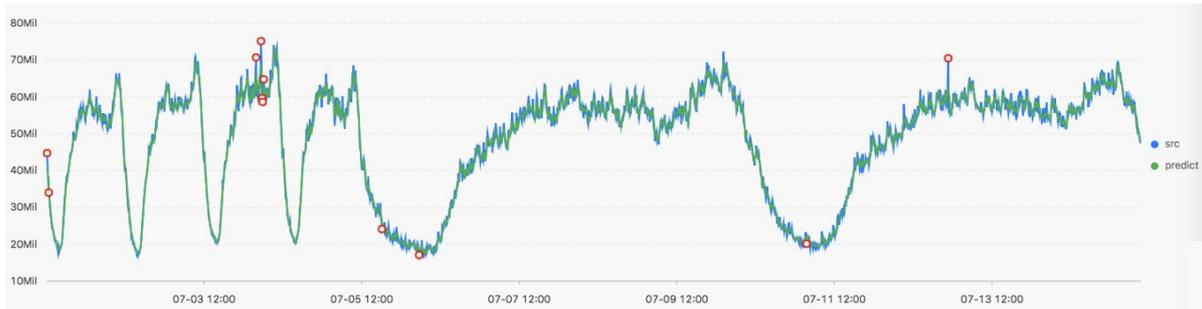
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>nPred</i>	The number of points for prediction.	The value is of the LONG data type and must be equal to or greater than 1.
<i>isSmooth</i>	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is True, which indicates to filter raw data.

Example:

- A search and analytic statement is shown as follows:

```
* | select ts_predicate_simple(stamp, value, 6) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from lo
g GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.
Vertical axis	src	The raw data.
	predict	The predicted data.
	upper	The upper limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	lower	The lower limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	anomaly_prob	The probability that the point is an anomaly. Valid values: [0, 1].

ts_predicate_ar

Syntax:

```
select ts_predicate_ar(x, y, p, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>p</i>	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 8].
<i>nPred</i>	The number of points for prediction.	The value is of the LONG data type. Valid values: [1, 5 × <i>p</i>].
<i>isSmooth</i>	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is true, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_ar(stamp, value, 3, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GR OUP BY stamp order by stamp)
```

Note The response is similar to that of the `ts_predicate_simple` function. For more information, see the response of the `ts_predicate_simple` function.

ts_predicate_arma

Syntax:

```
select ts_predicate_arma(x, y, p, q, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>p</i>	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 100].
<i>q</i>	The order of the ARMA model.	The value is of the LONG data type. Valid values: [2, 8].
<i>nPred</i>	The number of points for prediction.	The value is of the LONG data type. Valid values: [<i>p</i> , 5 <i>p</i>].
<i>isSmooth</i>	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is true, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_arma(stamp, value, 3, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

Note The response is similar to that of the `ts_predicate_simple` function. For more information, see the response of the `ts_predicate_simple` function.

ts_predicate_arima

Syntax:

```
select ts_predicate_arima(x,y, p, d, q, nPred, isSmooth)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A

Parameter	Description	Value
<i>p</i>	The order of the AR model.	The value is of the LONG data type. Valid values: [2, 8].
<i>d</i>	The order of the ARIMA model.	The value is of the LONG data type. Valid values: [1, 3].
<i>q</i>	The order of the ARMA model.	The value is of the LONG data type. Valid values: [2, 8].
<i>nPred</i>	The number of points for prediction.	The value is of the LONG type. Valid values: [<i>p</i> , 5 <i>p</i>].
<i>isSmooth</i>	Specifies whether to filter the raw data.	The value is of the Boolean type. The default value is True, which indicates to filter raw data.

An example search and analytic statement is shown as follows:

```
* | select ts_predicate_arma(stamp, value, 3, 1, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from m log GROUP BY stamp order by stamp)
```

 **Note** The response is similar to that of the `ts_predicate_simple` function. For more information, see the response of the `ts_predicate_simple` function.

ts_regression_predict

Syntax:

```
select ts_regression_predict(x, y, nPred, algotype, processType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>nPred</i>	The number of points for prediction.	The value is of the LONG data type. Valid values: [1, 500].

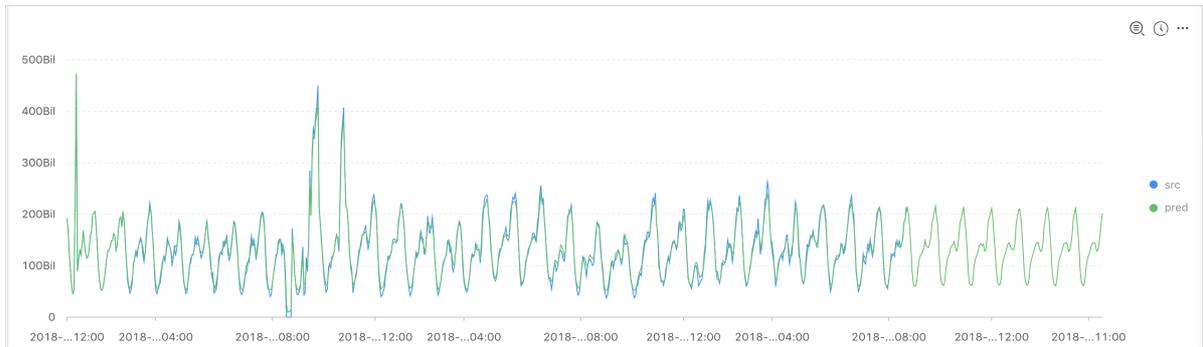
Parameter	Description	Value
<i>algotype</i>	The algorithm type for prediction.	Valid values: <ul style="list-style-type: none"> • origin: uses the Gradient Boosted Regression Tree (GBRT) algorithm for prediction. • forest: uses the GBRT algorithm for prediction based on the trend component decomposed by Seasonal and Trend decomposition using Loess (STL), and then uses the additive model to sum up the decomposed components and obtains the predicted data. • linear: uses the Linear Regression algorithm for prediction based on the trend components decomposed by STL, and then uses the additive model to sum up the decomposed components and obtains the predicted data.
<i>processType</i>	Specifies whether to preprocess the data.	Valid values: <ul style="list-style-type: none"> • 0: no additional data preprocessing is performed. • 1: removes abnormal data before prediction.

Example:

- A search and analytic statement is shown as follows:

```
* and h : nu2h05202.nu8 and m: NET | select ts_regression_predict(stamp, value, 200, 'origin') from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item		Description
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.
Vertical axis	src	The raw data.
	predict	The predicted data.

ts_anomaly_filter

Syntax:

```
select ts_anomaly_filter(lineName, ts, ds, preds, probs, nWatch, anomalyType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>lineName</i>	The name of each curve. The value is of the VARCHAR data type.	N/A
<i>ts</i>	The time sequence of the curve, which indicates the time of the current curve. The parameter value is an array of time points of the DOUBLE data type sorted in the ascending order.	N/A
<i>ds</i>	The actual value sequence of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
<i>preds</i>	The predicted value sequence of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
<i>probs</i>	The sequence of anomaly detection results of the curve. The parameter value is an array of data points with the same length as the ts parameter value.	N/A
<i>nWatch</i>	The number of the recently observed actual values on the curve. The value is of the LONG data type. The value must be smaller than the number of time points on the curve.	N/A
<i>anomalyType</i>	The type of anomaly to be filtered. The value is of the LONG data type.	Valid values: <ul style="list-style-type: none"> • 0: all anomalies. • 1: positive anomalies. • -1: negative anomalies.

Example:

- A search and analytic statement is shown as follows:

```
* | select res.name, res.ts, res.ds, res.preds, res.probs
  from (
    select ts_anomaly_filter(name, ts, ds, preds, probs, cast(5 as bigint), cast(1 as bigint)) as res
  from (
    select name, res[1] as ts, res[2] as ds, res[3] as preds, res[4] as uppers, res[5] as lowers, res[6] as probs
  from (
    select name, array_transpose(ts_predicate_ar(stamp, value, 10)) as res
  from (
    select name, stamp, value from log where name like '%asg-%') group by name)) );
```

- The following figure shows the response.

name	ts	ds	preds	probs
asg-bp1hylzdi2wx7civ0ivk	[1.5513696E9, 1.5513732E9, 1.5513768E9, 1.5513804E9]	[1,2,3,NaN]	[1,2,3,4]	[0,0,1,NaN]

30.4.9.7. Time series decomposition function

The time series decomposition function decomposes time series curves into curves that reveal the trend and periodicity of curves.

ts_decompose

Syntax:

```
select ts_decompose(x, y)
```

The following table lists the parameters of the function.

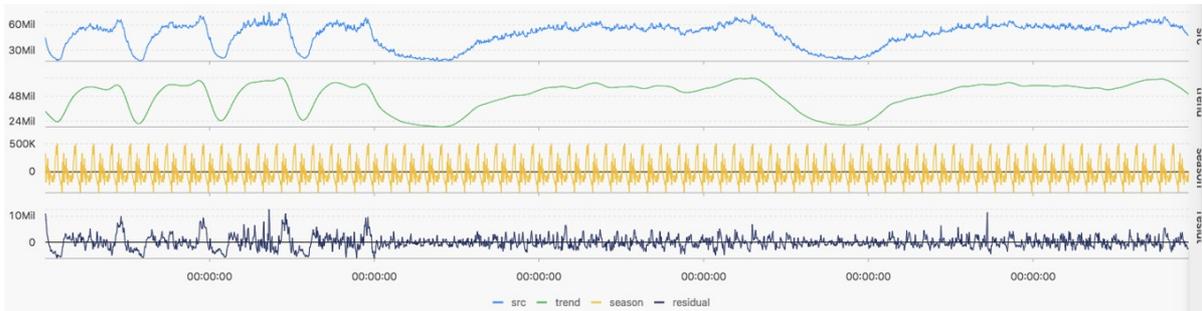
Parameter	Description	Value
x	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
y	The sequence of numeric data at each specified time point.	N/A

Example:

- A search and analytic statement is shown as follows:

```
* | select ts_decompose(stamp, value) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROU
P BY stamp order by stamp)
```

- The following figure shows the response.



The following table lists the display items.

Display item	Description	
Horizontal axis	unixtime	The Unix timestamp of the data. Unit: seconds.
Vertical axis	src	The raw time series data.
	trend	The decomposed data that indicates the trend of the time series data.
	season	The decomposed data that indicates the periodicity of the time series data.
	residual	The residual data decomposed from the time series data.

30.4.9.8. Time series clustering functions

You can use time series clustering functions to cluster multiple time series and obtain different curve shapes. Then, you can find the cluster center and identify curves with shapes that are different from other curve shapes in the cluster in a timely manner.

Functions

Function	Description
<code>ts_density_cluster</code>	Uses a density-based clustering method to cluster multiple time series.
<code>ts_hierarchical_cluster</code>	Uses a hierarchical clustering method to cluster multiple time series.
<code>ts_similar_instance</code>	Queries time series curves that are similar to a specified time series curve.

ts_density_cluster

Syntax:

```
select ts_density_cluster(x, y, z)
```

The following table lists the parameters of the function.

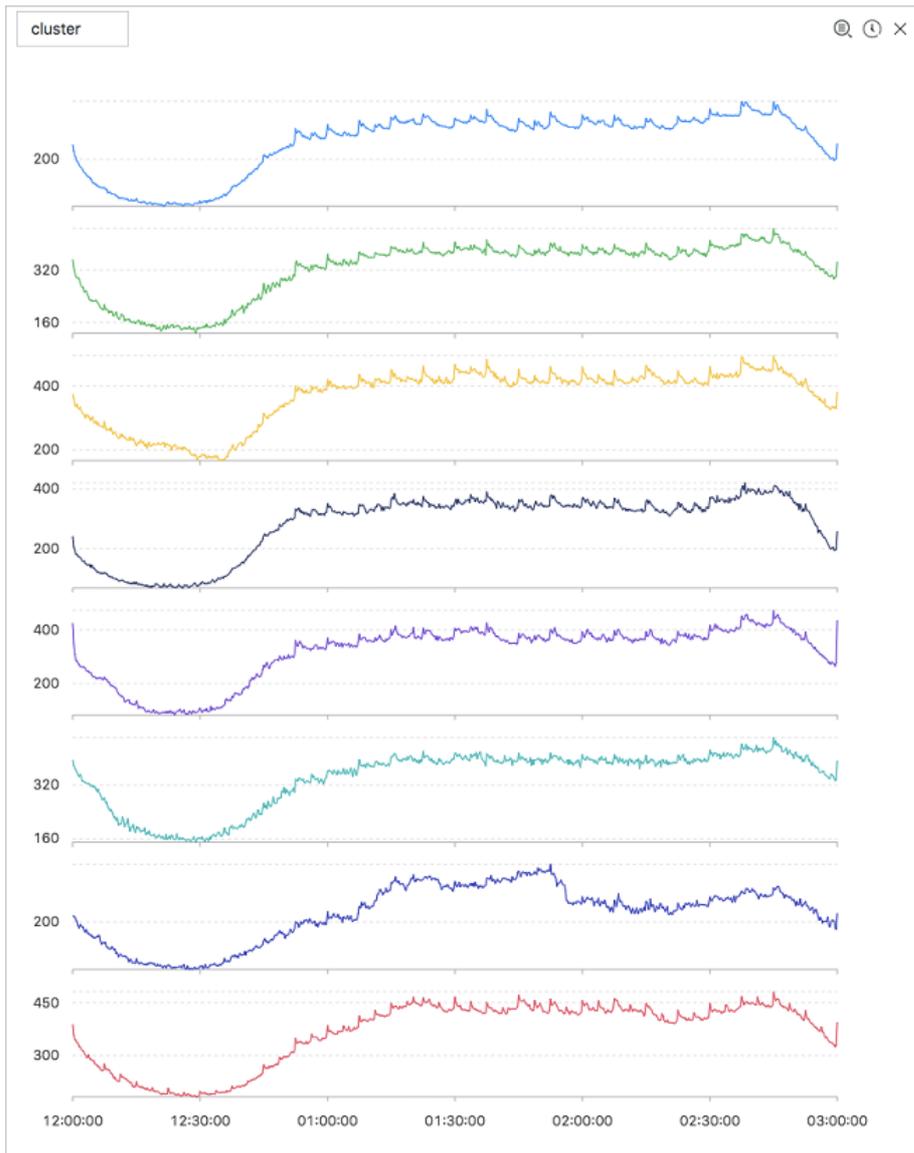
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>z</i>	The name of the metric that monitors the data at each time point.	The metric name is a string, for example, machine01.cpu_usr.

Example:

- A search and analytic statement is shown as follows:

```
* and (h: "machine_01" OR h: "machine_02" OR h: "machine_03") | select ts_density_cluster(stamp, metric_value, metric_name) from ( select __time__ - __time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY metric_name, stamp )
```

- The following figure shows the response.



The following table lists the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in any cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_hierarchical_cluster

Syntax:

```
select ts_hierarchical_cluster(x, y, z)
```

The following table lists the parameters of the function.

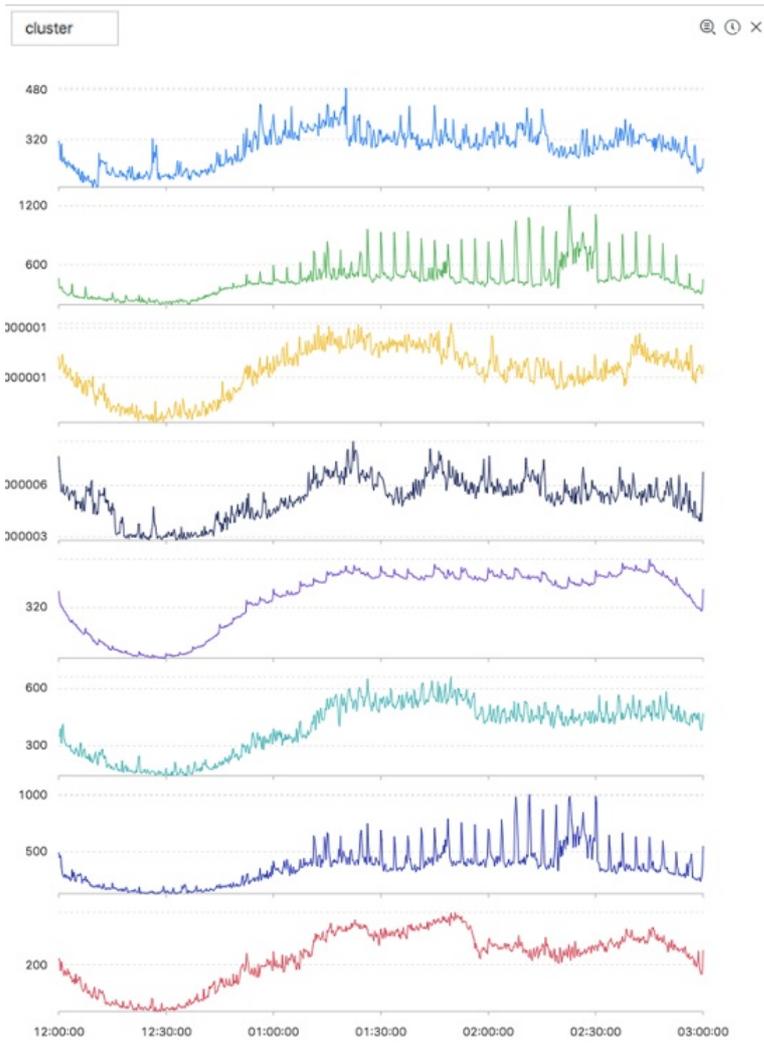
Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>z</i>	The name of the metric that monitors the data at each time point.	The metric name is a string, for example, machine01.cpu_usr.

Example:

- A search and analytic statement is shown as follows:

```
* and (h: "machine_01" OR h: "machine_02" OR h: "machine_03") | select ts_hierarchical_cluster(stamp, metric_value, metric_name) from ( select __time__ - __time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GR OUP BY stamp, metric_name order BY metric_name, stamp )
```

- The following figure shows the response.



The following table lists the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in any cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_similar_instance

Syntax:

```
select ts_similar_instance(x, y, z, instance_name, topK, metricType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>x</i>	The time sequence. The time points along the x axis are sorted in the ascending order.	Each time point is a Unix timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at each specified time point.	N/A
<i>z</i>	The name of the metric that monitors the data at each time point.	The metric name is a string, for example, machine01.cpu_usr.
<i>instance_name</i>	The name of the specified metric.	The metric name is a string, for example, machine01.cpu_usr.  Note You must specify an existing metric.
<i>topK</i>	The maximum number of curves that are similar to the specified curve can be returned.	N/A
<i>metricType</i>	{'shape', 'manhattan', 'euclidean'} : the metrics used to measure the similarity between time series curves.	N/A

An example search and analytic statement is shown as follows:

```
* and m: NET and m: Tcp and (h: "nu4e01524.nu8" OR h: "nu2i10267.nu8" OR h: "nu4q10466.nu8") | select ts_similar_instance(stamp, metric_value, metric_name, 'nu4e01524.nu8') from ( select __time__ - __time__ % 600 as stamp, sum(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY metric_name, stamp )
```

The following table lists the display items.

Display item	Description
instance_name	The list of metrics whose results are similar to the results of the specified metric.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.

30.4.9.9. Frequent pattern statistics function

The frequent pattern statistics function mines representative combinations of attributes from a specified multi-attribute field sample.

pattern_stat

Syntax:

```
select pattern_stat(array[col1, col2, col3], array['col1_name', 'col2_name', 'col3_name'], array[col5, col6], array['col5_name', 'col6_name'], supportScore, sample_ratio)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>array[col1, col2, col3]</i>	A column of values of the character data type.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
<i>array['col1_name', 'col2_name', 'col3_name']</i>	The field names of the values of the character data type.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
<i>array[col5, col6]</i>	A column of values of the numeric data type.	An array of values, for example, array[Inflow, OutFlow].
<i>array['col5_name', 'col6_name']</i>	The field names of the values of the numeric data type.	An array of field names, for example, array[originflow, 'OutFlow'].
<i>supportScore</i>	The support ratio of samples for pattern mining.	The value is of the DOUBLE data type. Valid values: (0,1).
<i>sample_ratio</i>	The sampling ratio. The default value is 0.1, which indicates that 10% of samples are used.	The value is of the DOUBLE data type. Valid values: (0,1).

- A search and analytic statement is shown as follows:

```
* | select pattern_stat(array[ Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent ], array[ 'Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent' ], array[ InFlow, OutFlow ], array[ 'InFlow', 'OutFlow' ], 0.45, 0.3) limit 1000
```

- Display item

Display item	Description
count	The number of samples in the current pattern.
supportScore	The support ratio for the current pattern.

Display item	Description
pattern	The content of the pattern, organized in the format that is defined by the query conditions.

30.4.9.10. Differential pattern statistics function

The differential pattern statistics function analyzes differential patterns of specified multi-field samples based on the specified condition. It helps you identify the causes of the differences under the current condition in a timely manner.

pattern_diff

Syntax:

```
select pattern_diff(array_char_value, array_char_name, array_numeric_value, array_numeric_name, condition, supportScore, posSampleRatio, negSampleRatio )
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>array_char_value</i>	A column of values of the character data type.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
<i>array_char_name</i>	The field names of the values of the character data type.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
<i>array_numeric_value</i>	A column of values of the numeric data type.	An array of values, for example, array[Inflow, OutFlow].
<i>array_numeric_name</i>	The field names of the values of the numeric data type.	An array of field names, for example, array[originflow, 'OutFlow'].
<i>condition</i>	The condition for filtering data. The value True indicates positive samples, and the value False indicates negative samples.	For example, Latency <= 300.
<i>supportScore</i>	The support ratio of positive and negative samples for pattern mining.	The value is of the DOUBLE data type. Valid values: (0,1].
<i>posSampleRatio</i>	The sampling ratio of positive samples. The default value is 0.5, indicating that 50% of positive samples are collected.	The value is of the DOUBLE data type. Valid values: (0,1].
<i>negSampleRatio</i>	The sampling ratio of negative samples. The default value is 0.5, indicating that 50% of positive samples are collected.	The value is of the DOUBLE data type. Valid values: (0,1].

Example:

- A search and analytic statement is shown as follows:

```
* | select pattern_diff(array[ Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent ], array[ 'Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent' ], array[ InFlow, OutFlow ], array[ 'InFlow', 'OutFlow' ], Latency > 300, 0.2, 0.1, 1.0) limit 1000
```

- Display item

Display item	Description
possupport	The support ratio of positive samples for the mined patterns.
posconfidence	The confidence level of the mined patterns in positive samples.
negsupport	The support ratio of negative samples for the mined patterns.
diffpattern	The content of the mined patterns.

30.4.9.11. Root cause analysis function

Log Service provides the alert and analytics features that help you quickly analyze data and locate anomalies of specific subdimensions of a metric. You can use the root cause analysis function to analyze the subdimension attributes that result in anomalies of the monitoring metric.

rca_kpi_search

Syntax

```
select rca_kpi_search(vvarchar_array, name_array, real, forecast, level)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>vvarchar_array</i>	The subdimension attributes.	The parameter value is formatted in an array, for example, array[col1, col2, col3].
<i>name_array</i>	The subdimension attribute names.	The parameter value is formatted in an array, for example, array['col1', 'col2', 'col3'].
<i>real</i>	The actual value of each subdimension attribute specified by the <i>vvarchar_array</i> parameter.	The parameter value is of the DOUBLE data type. Valid values: all real numbers.
<i>forecast</i>	The predicted value of each subdimension attribute specified by the <i>vvarchar_array</i> parameter.	The parameter value is of the DOUBLE data type. Valid values: all real numbers.
<i>level</i>	The number of subdimension attributes identified in the returned root cause set. The value 0 indicates that all root causes that are found are returned.	The parameter value is of the LONG data type. Valid values: [0, number of analyzed subdimensions]. The number of analyzed subdimensions is the length of the array specified by the <i>vvarchar_array</i> parameter.

Example:

- The search and analytic statement is shown as follows:

Use a subquery to obtain the actual value and predicted value of each subdimension attribute, and then call the `rca_kpi_search` function to analyze the root causes of anomalies.

```
* not Status:200 |
select rca_kpi_search(
  array[ ProjectName, LogStore, UserAgent, Method ],
  array[ 'ProjectName', 'LogStore', 'UserAgent', 'Method' ], real, forecast, 1)
from (
  select ProjectName, LogStore, UserAgent, Method,
  sum(case when time < 1552436040 then real else 0 end) * 1.0 / sum(case when time < 1552436040
  then 1 else 0 end) as forecast,
  sum(case when time >=1552436040 then real else 0 end) *1.0 / sum(case when time >= 1552436040
  then 1 else 0 end) as real
  from (
  select __time__ - __time__ % 60 as time, ProjectName, LogStore, UserAgent, Method, COUNT(*) as real
  from log GROUP by time, ProjectName, LogStore, UserAgent, Method )
  GROUP BY ProjectName, LogStore, UserAgent, Method limit 100000000)
```

- The following figure shows the response.



The following figure shows the structured response.

```
{
  "rcSets": [
    {
      "rcItems": [
        {
          "kpi": [{"attr": "xxx", "val": "xxx"}],
          "nleaf": 100,
          "change": 0.524543,
          "score": 0.1454543
        }
      ]
    }
  ]
}
```

The following table lists the display items.

Display item	Description
<i>rcSets</i>	The root cause sets. Each value of this parameter is an array.
<i>rcltems</i>	A specific root cause set.

Display item	Description
<i>kpi</i>	A specific item in the root cause set. Each item is formatted in an array where each element is a JSON object. The <code>attr</code> parameter indicates the subdimension name, and the <code>val</code> parameter indicates the attribute name under the subdimension.
<i>nleaf</i>	The number of leaf nodes that an item (KPI) in the root cause set covers in the original data. <div style="background-color: #e0f2f7; padding: 5px;"> Note Leaf node: the log entry that contains the finest-grained attribute information.</div>
<i>change</i>	The ratio of anomalies of leaf nodes in a KPI to the total anomalies in the root cause set that occurred at the same time point.
<i>score</i>	The abnormality score of the current KPI. Valid values: [0, 1].

The response is formatted in a JSON object as follows:

```

{
  "rcSets": [
    {
      "rcItems": [
        {
          "kpi": [
            {
              "attr": "country",
              "val": "*"
            },
            {
              "attr": "province",
              "val": "*"
            },
            {
              "attr": "provider",
              "val": "*"
            },
            {
              "attr": "domain",
              "val": "download.huya.com"
            },
            {
              "attr": "method",
              "val": "*"
            }
          ],
          "nleaf": 119,
          "change": 0.3180687806279939,
          "score": 0.14436007709620113
        }
      ]
    }
  ]
}

```

30.4.9.12. Correlation analysis functions

You can use a correlation analysis function to find the metrics that are correlated with a specified metric or time series data among multiple observed metrics in the system.

Functions

Function	Description
<code>ts_association_analysis</code>	Quickly finds the metrics that are correlated with a specified metric among multiple observed metrics in the system.

Function	Description
<code>ts_similar</code>	Quickly finds the metrics that are correlated with specified time series data among multiple observed metrics in the system.

ts_association_analysis

Syntax

```
select ts_association_analysis(stamp, params, names, indexName, threshold)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>stamp</i>	The Unix timestamp of the LONG data type.	-
<i>params</i>	The metrics to be analyzed, formatted in an array where each element is of the DOUBLE data type.	The parameter value is formatted in an array where each element is of the DOUBLE data type. For example, Latency, QPS, and NetFlow.
<i>names</i>	The names of the metrics to be analyzed.	The parameter value is formatted in an array where each element is of the VARCHAR data type. For example, Latency, QPS, and NetFlow.
<i>indexName</i>	The name of the target metric.	The parameter value is of the VARCHAR data type, for example, Latency.
<i>threshold</i>	The threshold of correlation between the metrics to be analyzed and the target metric.	The parameter value is of the DOUBLE data type. Valid values: [0, 1].

Response

- **name**: the name of the metric that meets the specified correlation condition with the target metric.
- **score**: the value of correlation between the returned metric and the target metric. Valid values: [0, 1].

Sample statement

```
* | select ts_association_analysis(
  time,
  array[inflow, outflow, latency, status],
  array['inflow', 'outflow', 'latency', 'status'],
  'latency',
  0.1) from log;
```

Sample response

```
| results      |
| ----- |
| ['latency', '1.0'] |
| ['outflow', '0.6265'] |
| ['status', '0.2270'] |
```

ts_similar

Syntax 1

```
select ts_similar(stamp, value, ts, ds)
select ts_similar(stamp, value, ts, ds, metricType)
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>stamp</i>	The Unix timestamp of the LONG data type.	-
<i>value</i>	The value of the metric to be analyzed. The parameter value is of the DOUBLE data type.	-
<i>ts</i>	The time sequence of the specified time series curve. The parameter value is formatted in an array where each element is of the DOUBLE data type.	-
<i>ds</i>	The sequence of numeric data of the specified time series curve.	-
<i>metricType</i>	The type of correlation between the measured curves. The parameter value is of the VARCHAR data type.	Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL

Syntax 2

```
select ts_similar(stamp, value, startStamp, endStamp, step, ds)
select ts_similar(stamp, value, startStamp, endStamp, step, ds, metricType )
```

The following table lists the parameters of the function.

Parameter	Description	Value
<i>stamp</i>	The Unix timestamp of the LONG data type.	-
<i>value</i>	The value of the metric to be analyzed. This parameter is of the DOUBLE data type.	-
<i>startStamp</i>	The start timestamp of the specified time series curve. The parameter value is of the LONG data type.	-

Parameter	Description	Value
<i>endStamp</i>	The end timestamp of the specified time series curve. The parameter value is of the LONG data type.	-
<i>step</i>	The time interval between two adjacent data points in a time series. The parameter value is of the LONG data type.	-
<i>ds</i>	The sequence of numeric data of the specified time series curve. The parameter is formatted in an array where each element is of the DOUBLE data type.	-
<i>metricType</i>	The type of correlation between the measured curves. The parameter value is of the VARCHAR data type.	Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL

- **Response**

score: the correlation between the analyzed metric and the specified time series curve. Valid values: [-1, 1].

- **Sample statement**

```
* | select vhost, metric, ts_similar(time, value, 1560911040, 1560911065, 5, array[5.1,4.0,3.3,5.6,4.0,7.2], 'PEARSON') from log group by vhost, metric;
```

- **Sample response**

```
| vhost | metric      | score          |
|-----|-----|-----|
| vhost1 | redolog     | -0.3519082537204182 |
| vhost1 | kv_qps      | -0.15922168009772697 |
| vhost1 | file_meta_write | NaN           |
```

30.4.9.13. Kernel density estimation function

Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable.

The Kernel density estimation function uses the smooth peak function to fit the observed data points. In this way, the function simulates the real probability distribution curve.

- **Syntax**

```
select kernel_density_estimation(bigint stamp, double value, varchar kernelType)
```

- **Parameters**

Parameter	Description
stamp	The Unix timestamp of observed data. Unit: second.
value	The observed value.

Parameter	Description
kernelType	<ul style="list-style-type: none"> ◦ box: rectangle window. ◦ epanechniov: Epanechnikov curve. ◦ gausener: Gaussian curve.

• Response

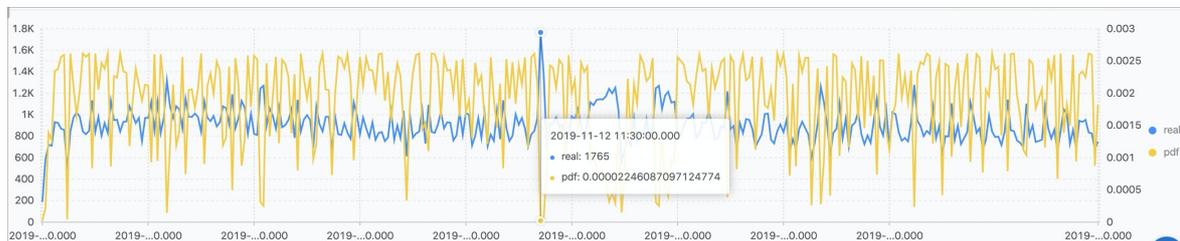
Display item	Description
unixtime	The Unix timestamp of observed data.
real	The observed value.
pdf	The probability of each observed data point.

• Example

◦ Sample statement

```
* |
select
  date_trunc('second', cast(t1[1] as bigint)) as time, t1[2] as real, t1[3] as pdf from (
    select kernel_density_estimation(time, num, 'gaussian') as res from (
      select __time__ - __time__ % 10 as time, COUNT(*) * 1.0 as num from log group by time order by time)
    ), unnest(res) as t(t1) limit 1000
```

◦ Response



30.4.10. Advanced analysis

30.4.10.1. Optimize queries

This topic describes how to optimize queries to improve query efficiency.

You can use the following methods to optimize queries:

- Increase the number of shards.
- Reduce the query time range and data volume.
- Repeat queries multiple times.
- Optimize the SQL statement for queries.

Increase the number of shards

More shards represent more computing resources and faster computing speed. You can increase the number of shards to ensure that the average number of log entries to be scanned in each shard does not exceed 50 million. You can increase the number of shards by splitting shards. For more information, see [Split a shard](#).

 **Note** Splitting shards incurs more fees and only accelerates queries of new data. Existing data is still stored in old shards.

Reduce the query time range and data volume

- The larger the time range, the slower the query. If you query data within a year or a month, data is computed by day. To facilitate computing, you can reduce the query time range.
- The larger the data volume, the slower the query. Reduce the amount of data to be queried as much as possible.

Repeat queries multiple times

If you find that the result of a query is inaccurate, you can repeat the query multiple times. The underlying acceleration mechanism ensures that each query uses the previous query result to analyze data. In this way, multiple queries make the query result more accurate.

Optimize the SQL statement for queries

A time-consuming query statement has the following characteristics:

- Performs the GROUP BY operation on string-type columns.
- Performs the GROUP BY operation on more than five fields.
- Includes operations that generates strings.

You can use the following methods to optimize a query statement:

- Avoid operations that generate strings if possible.
 - If you use the date_format function to generate a formatted timestamp, the query is inefficient.

```
* | select date_format(from_unixtime(__time__), '%H:%i') as t, count(1) group by t
```

- If you use the substr() function, strings are generated. We recommend that you use the date_trunc or time_series function in a query statement.
- Avoid performing the GROUP BY operation on string-formatted columns if possible.

Performing the GROUP BY operation on strings may result in a large number of hash calculations, which account for more than 50% of total calculations. Examples:

```
* | select count(1) as pv, date_trunc('hour', __time__) as time group by time
* | select count(1) as pv, from_unixtime(__time__ - __time__ % 3600) as time group by __time__ - __time__ % 3600
```

Both query 1 and query 2 count the number of log entries per hour. However, query 1 converts the time into a string, for example, 2017-12-12 00:00:00, and then performs the GROUP BY operation on this string. Query 2 calculates the on-the-hour time value, performs the GROUP BY operation on the result, and then converts the value into a string. Query 1 is less efficient than query 2 because query 1 needs to hash strings.

- List fields alphabetically based on the initial letter when performing the GROUP BY operation on multiple columns.

For example, you need to query 100 million users who are from 13 provinces.

```
Fast: * | select province,uid,count(1)groupby province,uid
Slow: * | select province,uid,count(1)groupby uid,province
```

- Use estimating functions.

Estimating functions provide stronger performance than accurate calculation. In estimation, accuracy is compromised to an acceptable extent for fast calculation.

```
Fast: * |select approx_distinct(ip)
Slow: * | select count(distinct(ip))
```

- Specify only required columns in the SQL statement if possible.

You can specify all columns in the search statement. In the SQL statement, specify only required columns if possible. This will speed up calculation.

```
Fast: * |select a,b c
Slow: * |select *
```

- Place columns that do not need to be grouped in an aggregate function if possible.

For example, a user ID is associated with a username. Therefore, you can execute the Group By operation on user IDs to analyze data.

```
Fast: * | select userid, arbitrary(username), count(1)groupby userid
Slow: * | select userid, username, count(1)groupby userid,username
```

- Avoid using the IN operator if possible.

If possible, avoid using the IN clause in SQL statements. Instead, use the OR clause.

```
Fast: key : a or key :b or key:c | select count(1)
Slow: * | select count(1) where key in ('a','b')
```

30.4.10.2. Use cases

This topic provides some use cases of log data analysis.

Trigger an alert when the error rate exceeds 40% over the last 5 minutes

Calculate the percentage of 500 Internal Server Error every minute. An alert is triggered when the error rate exceeds 40% over the last 5 minutes.

```
status:500 | select __topic__, max_by(error_count>window_time)/1.0/sum(error_count) as error_ratio, sum(error_count)
as total_error from (
select __topic__, count(*) as error_count , __time__ - __time__ % 300 as window_time from log group by __topic__, windo
w_time
)
group by __topic__ having max_by(error_count>window_time)/1.0/sum(error_count) > 0.4 and sum(error_count) > 500
order by total_error desc limit 100
```

Calculate the amount of transferred data and configure alerts

Calculate the amount of transferred data every minute. An alert is triggered when transferred data plunges. Transferred data counted in the last minute does not cover a full minute. The `(max(time) - min(time))` clause is used for normalization to count the average traffic per minute.

```
* | SELECT SUM(inflow) / (max(__time__) - min(__time__)) as inflow_per_minute, date_trunc('minute',__time__) as minut
e group by minute
```

Calculate the average latency of traffic data in different sizes

Distribute traffic data to multiple bins based on the data size and calculate the average latency of the data in the bins.

```
* | select avg(latency) as latency , case when originSize < 5000 then 's1' when originSize < 20000 then 's2' when originSi
ze < 500000 then 's3' when originSize < 100000000 then 's4' else 's5' end as os group by os
```

Retrieve the percentages of different results

List the number and the percentage of each result for different departments. This query includes subqueries and window functions. The `sum(c) over()` clause indicates the sum of values in all rows.

```
* | select department, c*1.0/ sum(c) over () from(select count(1) as c, department from log group by department)
```

Count the number of log entries that meet the query condition

To count the number of URLs based on their characteristics, you can use the `CASE WHEN` clause or the `COUNT_IF` clause. The latter clause is simpler.

```
* | select count_if(uri like '%login') as login_num, count_if(uri like '%register') as register_num, date_format(date_trunc('minute', __time__), '%m-%d %H:%i') as time group by time order by time limit 100
```

30.4.10.3. Time field conversion examples

During search and analytics, you often need to process time fields in log data, such as converting a timestamp to another time format. This topic uses some examples to describe how to convert time fields.

A log entry may include multiple time fields, for example:

- `__time__`: the time that you specify when you use the API or SDK to write log data. This field can be used for log data shipping, search, and analytics.
- Original time field in log data: the field that records the time when the log data is generated. This field is in raw logs.

Time fields in different formats are difficult to read. To simplify the read process, you can convert the time format during search and analytics. For example, you can perform the following conversions:

1. [Convert __time__ to a timestamp](#)
2. [Display __time__ in a specified format](#)
3. [Convert a timestamp to a specified format](#)

Convert __time__ to a timestamp

You can use the `from_unixtime` function to convert the `__time__` field to a timestamp.

```
* | select from_unixtime(__time__)
```

Display __time__ in a specified format

To display the `__time__` field in the format of `YYYY-MM-DD HH:MM:SS`, you can use the `date_format` function.

```
* | select date_format(__time__, '%Y-%m-%d %H:%i:%S')
```

Convert the time in a log to a specified format

To convert the time field in a log to the specified format (`YYYY-MM-DD HH:MM:SS`) and perform the `GROUP BY` operation on the `YYYY-MM-DD` part, you can use the `date_format` function.

- Sample log entry

```
__topic__:
body_byte_sent: 307
hostname: www.host1.com
http_user_agent: Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X) AppleWebKit/603.3.8 (KHTML, like Gecko)
Mobile/14G60 QQ/7.1.8.452 V1_IPH_SQ_7.1.8_1_APP_A Pixel/750 Core/UIWebView NetType/WIFI QBWebViewType/1
method: GET
referer: www.host0.com
remote_addr: 36.63.1.23
request_length: 111
request_time: 2.705
status: 200
upstream_response_time: 0.225582883754
url: /? k0=v9&
time:2017-05-17 09:45:00
```

- Example SQL statement

```
* | select date_format (date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d') as day, count(1) as uv group by day or
der by day asc
```

30.4.11. Visual analysis

30.4.11.1. Analysis graph

30.4.11.1.1. Overview

All search and analytics results can be rendered by using visualized charts.

Prerequisites

- The index feature is enabled and configured. The analytics switches are turned on. For more information, see [Enable the index feature and configure indexes for a Logstore](#).
- An analytic statement is included in a query statement. You cannot use charts to show query results if you do not include an analytic statement in your query statement.

Precautions

When multiple search and analytic statements are being executed in sequence, the **Value Column**, **X Axis**, or **Y Axis** information cannot automatically change based on the search and analytic statement. The X and Y axis information may remain the same as the last search and analytic statement. If this happens, the query results of the current search and analytic statement cannot be automatically displayed in a chart. If the following messages are returned, configure parameters on the **Properties** tab based on the current search and analytic statement:

- The currently selected dimensions are not in the queried results. Check and configure the attributes.
- X-Axis or Y-Axis is not available. Check and configure the attributes.

Chart configurations

On the **Graph** tab, various charts are provided to show query results. You can select a type of chart from the chart bar to show results.

- On the **Graph** tab, you can view the **Chart Preview** and **Data Preview** of query results of the current search and analytic statement. **Chart Preview** is the preview of the query results that are displayed in the specified type of chart. **Data Preview** displays the query results in a table.

- On the **Graph** tab on the right, you can configure the following chart properties:
 - **Data Source:** used to set placeholder variables. For example, you configure the drill-down event of Chart A to redirect to the dashboard where Chart B is located. The placeholder variable you configured for Chart B is the same as the variable that you click to trigger the drill-down event. Then the placeholder variable is replaced with the variable you click to trigger the drill-down event and the search and analytic statement of Chart B is executed. For more information, see [Drill-down analysis](#).

This feature is applicable to scenarios where you configure drill-down events to redirect to targeted dashboards.

- **Properties:** used to configure the display properties of a chart, including the X axis, left and right Y axes, margins, font size and other properties. The properties vary with different type of charts.

This feature is applicable to all search and analytics scenarios.

- **Interactive Behavior:** used to configure drill-down events for a chart. After you configure a drill-down event for the chart, you can click the variable value in the chart to trigger the specified drill-down event. For more information, see [Drill-down analysis](#).

This feature is applicable to triggering drill-down events for charts.

30.4.11.1.2. Table

Tables are the most common method to sort and display data for reference and analysis. All search and analytics results in Log Service can be rendered by using visualized charts. By default, the query results are displayed in a table.

Components

- Table header
- Row
- Column

where,

- You can use a `SELECT` statement to specify the number of columns.
- The number of rows is computed based on the number of log entries in the specified time range. The default clause is `LIMIT 100`.

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analysis**.
2. On the **Graph** tab that appears, view the data in a table. You do not need to click .
3. On the **Properties** tab on the right, configure the properties of the table.

Properties

Parameter	Description
Items per Page	The number of entries to return on each page.
Zebra Striping	Specifies whether to display the query results in a zebra-striped table.
Transpose Rows and Columns	Specifies whether to transpose rows and columns.
Hide Reserved Fields	Specifies whether to hide reserved fields.
Disable Sorting	Specifies whether to disable the sorting feature.

Parameter	Description
Disable Search	Specifies whether to disable the search feature.
Highlight Settings	The rules for highlighting rows or columns that conform to rules.

30.4.11.1.3. Line chart

A line chart is used to analyze the value changes of fields based on an ordered data type (a continuous time range in most cases).

You can use a line chart to analyze the following change characteristics of field values over a period:

- Increment or decrement
- Increment or decrement rate
- Increment or decrement pattern, for example, periodicity
- Peak value and valley value

Line charts are suitable for analyzing field value changes over time. You can also use a line chart to analyze the value changes of multiple fields in multiple lines over the same period and reveal the relationship between the fields. For example, the values of multiple fields are positively or negatively associated with each other.

Components

- X-axis
- Left Y-axis
- (Optional) Right Y-axis
- Data point
- Line of changing trend
- Legend

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, click  to select the line chart.
3. On the Properties tab on the right, configure the properties of the line chart.

 **Note** In a line chart, a single line must contain more than two data points. Otherwise, the data trend cannot be analyzed. We recommend that you configure no more than five lines in a line chart.

Properties

Parameter	Description
X Axis	The data on the X-axis, which is usually a time sequence.
Left Y Axis	The numeric data on the left Y-axis. You can configure one or more fields for the left Y-axis.
Right Y Axis	The numeric data on the right Y-axis. You can configure one or more fields for the right Y-axis. The layer of the right Y-axis is higher than that of the left Y-axis.

Parameter	Description
Column Marker	The column on the left Y-axis or right Y-axis that is selected as a histogram.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format Left Y-axis	The format in which data configured on the left Y-axis and right Y-axis is displayed.
Format Right Y-axis	
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

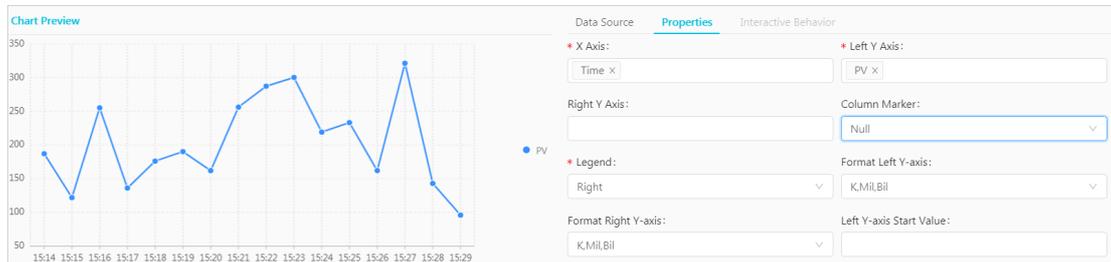
Example of a simple line chart

To query the page views (PVs) of the IP address 10.0.192.0 in the last 24 hours, execute the following statement:

```
remote_addr: 10.0.192.0 | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i')
as time, count(1) as PV group by time order by time limit 1000
```

Select **time** for X Axis, **PV** for Left Y Axis, and **Bottom** for Legend. Adjust the margins based on your needs.

Simple line chart



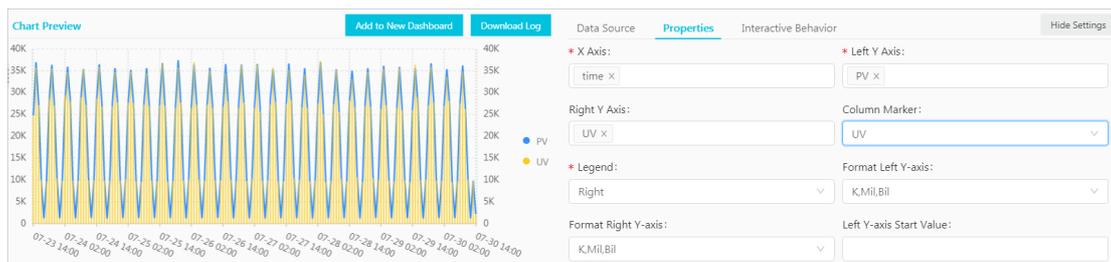
Example of a dual Y-axis line chart

To query the access PVs and unique visitors (UVs) in the last 24 hours, execute the following statement:

```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV, approx_distinct(remote_ad
dr) as UV group by time order by time limit 1000
```

Select **time** for X Axis, **PV** for Left Y Axis, **UV** for Right Y Axis, and **PV** for Column Marker.

Dual Y-axis line chart



30.4.11.1.4. Column chart

A column chart uses vertical or horizontal bars to present categorical values and count the number of values in each category. In contrast, a line chart describes ordered data.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Rectangular block
- Legend

By default, column charts in Log Service use vertical bars. Each rectangular bar has a fixed width and a varying height that indicates a value. You can use a grouped column chart to display the data if multiple columns of data are configured for the Y-axis.

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, click  to select the column chart.
3. On the Properties tab on the right, configure the properties of the column chart.

 **Note** Column charts are suitable to display query results if the number of returned log entries is no greater than 20. You can use a `LIMIT` clause to control the number of categorical rectangular bars. Analysis results may not be clearly displayed if the chart contains excessive rectangular bars. In addition, we recommend that you configure no more than five fields for the Y-axis.

Properties

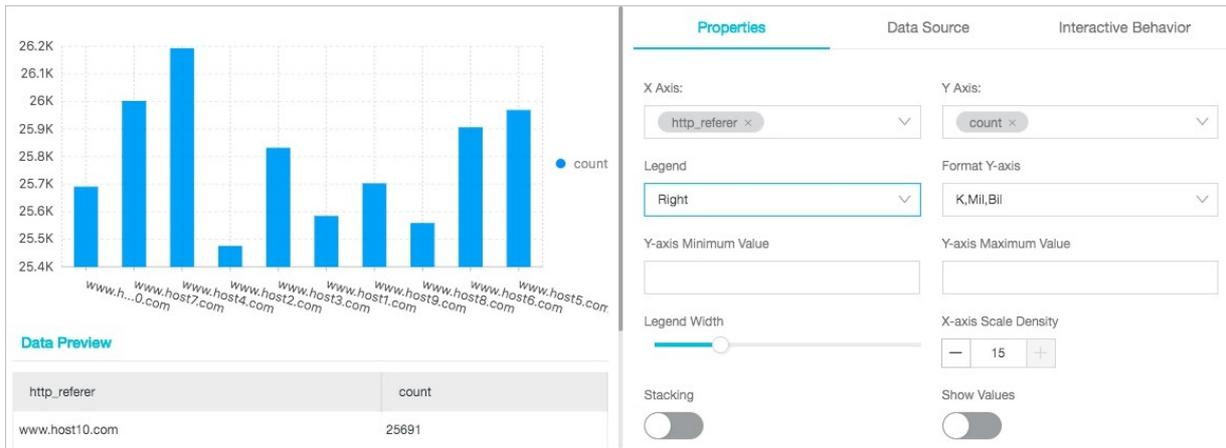
Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can configure one or more fields for the Y-axis.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data configured for the Y-axis is displayed.
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

Example of a simple column chart

To query the number of requests for each `http_referer` in the specified time range, execute the following statement:

```
* | select http_referer, count(1) as count group by http_referer
```

Select `http_referer` for X Axis and `count` for Y Axis.

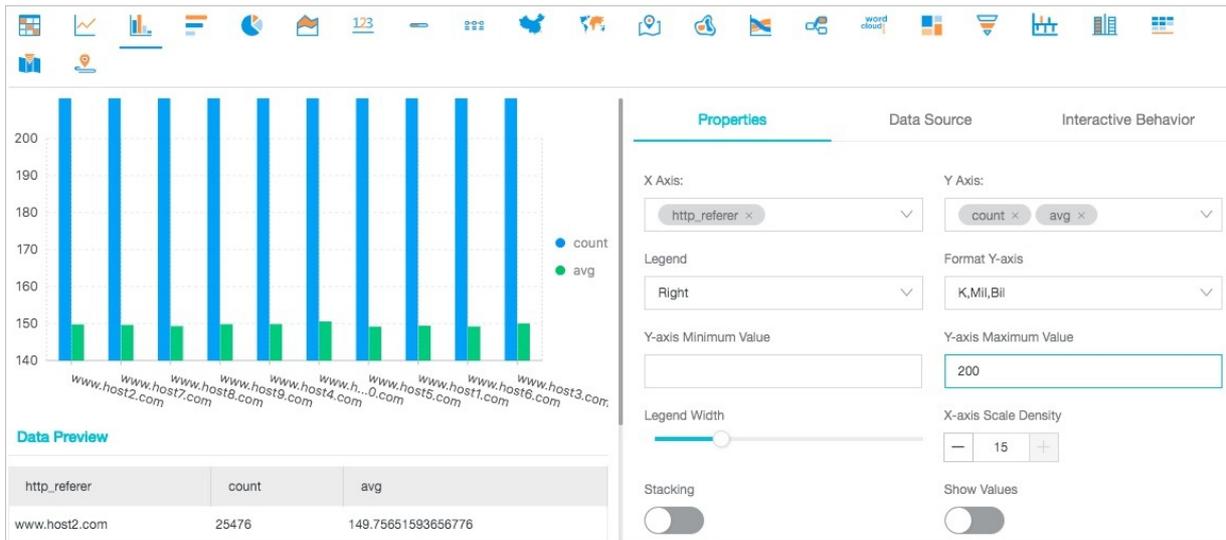


Example of a grouped column chart

To query the number of requests and the average bytes for each `http_referer` in the specified time range, execute the following statement:

```
* | select http_referer, count(1) as count, avg(body_bytes_sent) as avg group by http_referer
```

Select `http_referer` for X Axis. Select `count` and `avg` for Y Axis.



30.4.11.1.5. Bar chart

A bar chart is a horizontal column chart. It is used to analyze the top N values of fields. It is configured in a way similar to a column chart.

Components

- X-axis (vertical)
- Y-axis (horizontal)
- Rectangular block
- Legend

Each rectangular bar has a fixed height and a varying width that indicates a value. You can use a grouped bar chart to display the data if multiple columns of data are configured for the Y-axis.

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, select the bar chart .
3. On the Properties tab on the right, configure the properties of the bar chart.

Note

- Bar charts are suitable to show query results if the number of returned log entries is no greater than 20. You can use a `LIMIT` clause to control the number of categorical rectangular bars. Analysis results may not be clearly displayed if the chart contains excessive rectangular bars. You can use an `ORDER BY` clause to analyze the top N values of fields. In addition, we recommend that you configure no more than five fields for the Y-axis.
- You can use a grouped bar chart to show query results. However, the values represented by each rectangular bar in a group must be positively or negatively associated with each other.

Properties

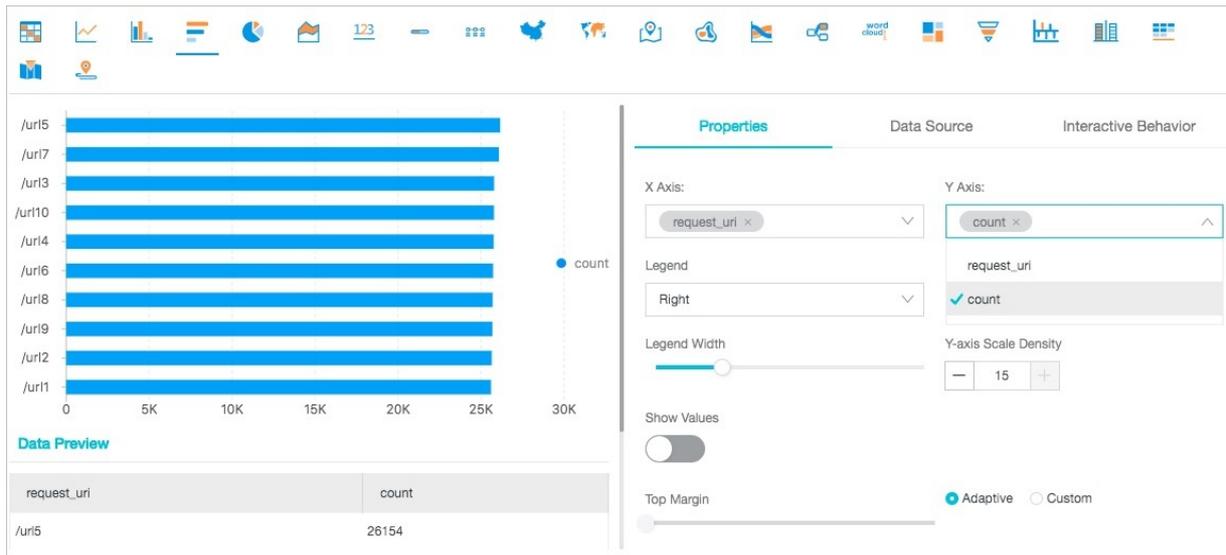
Parameters

Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can configure one or more fields for the Y-axis.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format X-axis	The format in which data configured for the X-axis is displayed.
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

Examples

To analyze the `request_uri` with the top 10 number of visits and display the analysis results in a bar chart, execute the following statement:

```
* | select request_uri, count(1) as count group by request_uri order by count desc limit 10
```



30.4.11.1.6. Pie chart

A pie chart is used to analyze the percentages of the number of occurrences of different field values based on the arc length.

Components

- Sector
- Percentage in the text format
- Legend

Types

Log Service provides three types of pie charts: the default pie chart, the donut chart, and the polar area chart.

- Pie Chart

A pie chart is divided into multiple sectors based on the percentages of various field values. The entire chart indicates all field values. Each arc-shaped sector indicates the percentage of the occurrences of a field value to the total occurrences. The sum of percentages in all sectors is equal to 100%.

- Donut chart

A donut chart is a pie chart with a hollow center. It has the following features:

- A donut chart displays the total number of occurrences in addition to the information that a basic pie chart can display.
- You can obtain the differences between the number of occurrences of the same value in two charts based on the ring length. This is more intuitive than when you compare two pie charts.

- Polar area chart

A polar area chart is not a donut chart, but a column chart in the polar coordinate system. Each category of field values is represented by a sector with the same radian. The radius of a sector indicates the number of occurrences of a field value. Compared with a pie chart, a polar area chart has the following features:

- Pie charts are suitable to display query results if the number of returned log entries is no greater than 10. Polar area charts are suitable to display query results if the number of returned log entries ranges from 10 to 30.
- The area of a sector is proportional to its radius squared. Therefore, the polar area chart can highlight the differences between the number of occurrences of various values. It is especially suitable for comparing the number of occurrences of similar values.

- A circle can be used to display a periodic pattern. Therefore, you can use a polar area chart to analyze value change characteristics in specified periods, such as weeks and months.

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, click  to select the pie chart.
3. On the **Properties** tab on the right, configure the properties of the pie chart.

Note

- Pie charts and donut charts can be used to display query results if the number of returned log entries is no greater than 10. You can use a `LIMIT` clause to control the number of sectors. Analysis results may not be clearly displayed if the chart contains excessive sectors of different colors.
- Use the polar area chart or column chart if the number of log entries exceeds 10.

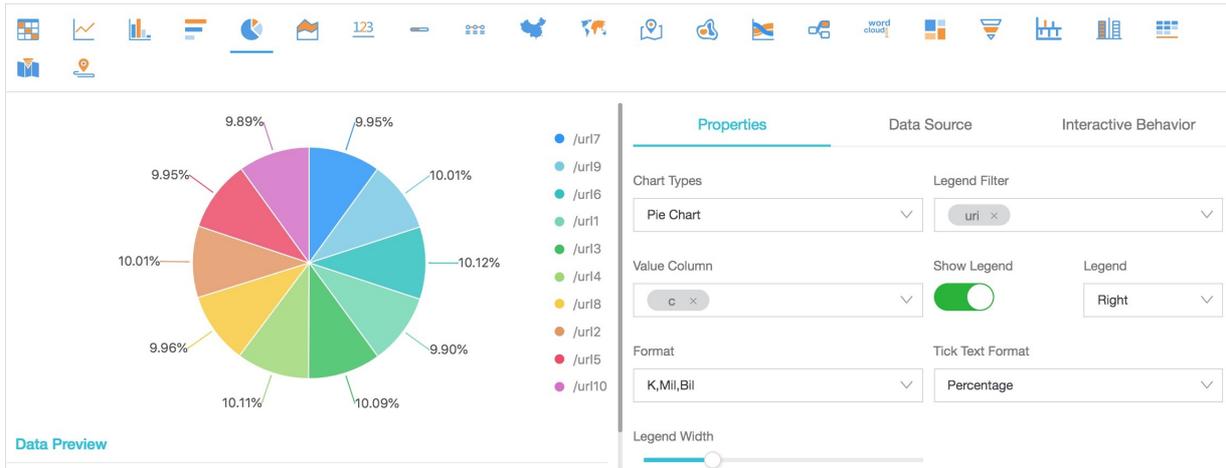
Properties

Parameter	Description
Chart Types	The type of the chart. Valid values: Pie Chart, Donut Chart, and Polar Area Chart. Default value: Pie Chart.
Legend Filter	The categorical data.
Value Column	The values corresponding to different categories of data.
Show Legend	Specifies whether to show the legend.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right. You can configure this parameter only after you turn on the Show Legend switch.
Format	The format in which data is displayed.
Tick Text Format	The format of the tick.
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

Example of a pie chart

To analyze the percentage of the access `requestURI`, execute the following statement:

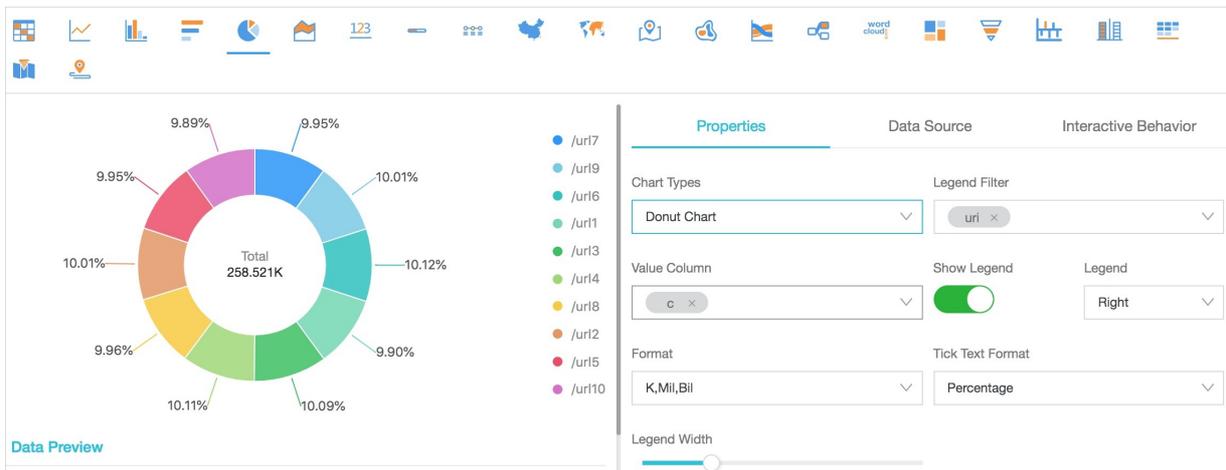
```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



Example of a donut chart

To analyze the percentage of the access `requestURI` , execute the following statement:

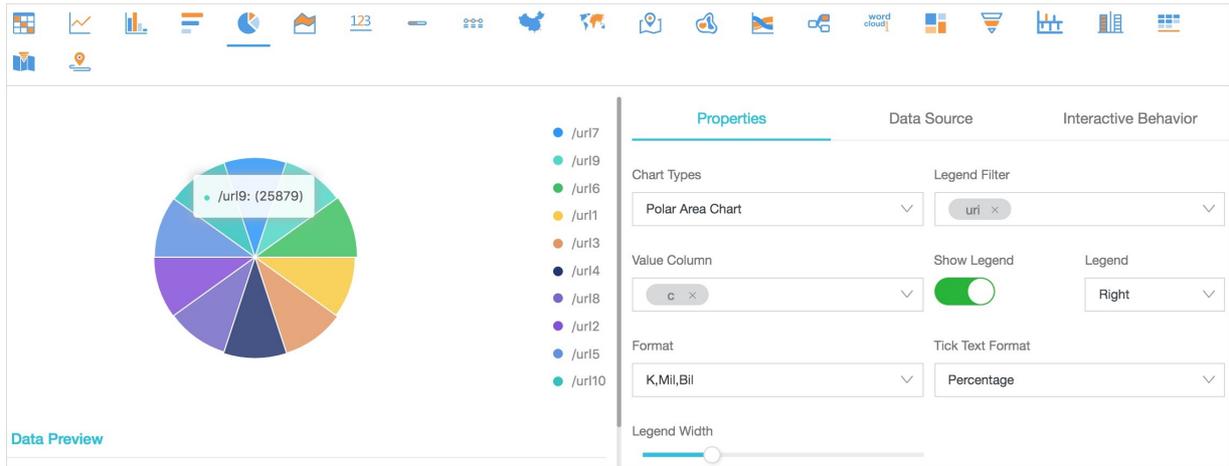
```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



Example of a polar area chart

To analyze the percentage of the access `requestURI` , execute the following statement:

```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



30.4.11.1.7. Area chart

An area chart is constructed based on a line chart. An area chart is the section between a line and the axis that is filled with a color. The color highlights the trend. Similar to a line chart, an area chart emphasizes the numeric value changes over time and is used to highlight the overall data trend. Both the line chart and the area chart indicate the trend and relationship of numeric values. They are not suitable if you want to display specific values.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Area block

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, click  to select the area chart.
3. On the Properties tab on the right, configure the properties of the area chart.

 **Note** In an area chart, a single area block must contain more than two data points. Otherwise, the data trend cannot be analyzed. We recommend that you configure no more than five area blocks in an area chart.

Properties

Parameter	Description
X Axis	The data on the X-axis, which is usually a time sequence.
Y Axis	The numeric data. You can configure one or more fields for the Y-axis.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.

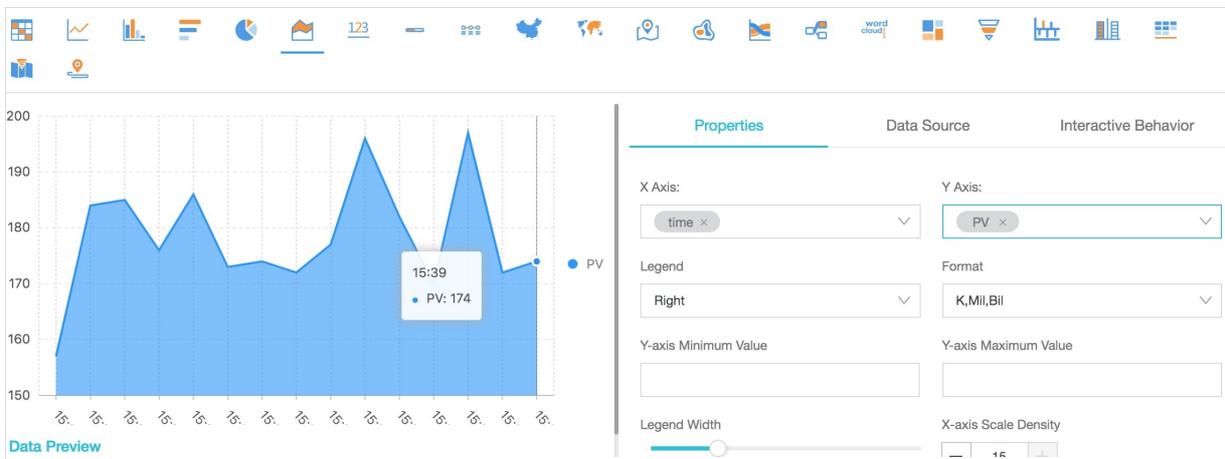
Parameter	Description
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

Example of a simple area chart

To query the page views (PVs) of the IP address `10.0.192.0` in the last 24 hours, execute the following statement:

```
remote_addr: 10.0.192.0 | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV group by time order by time limit 1000
```

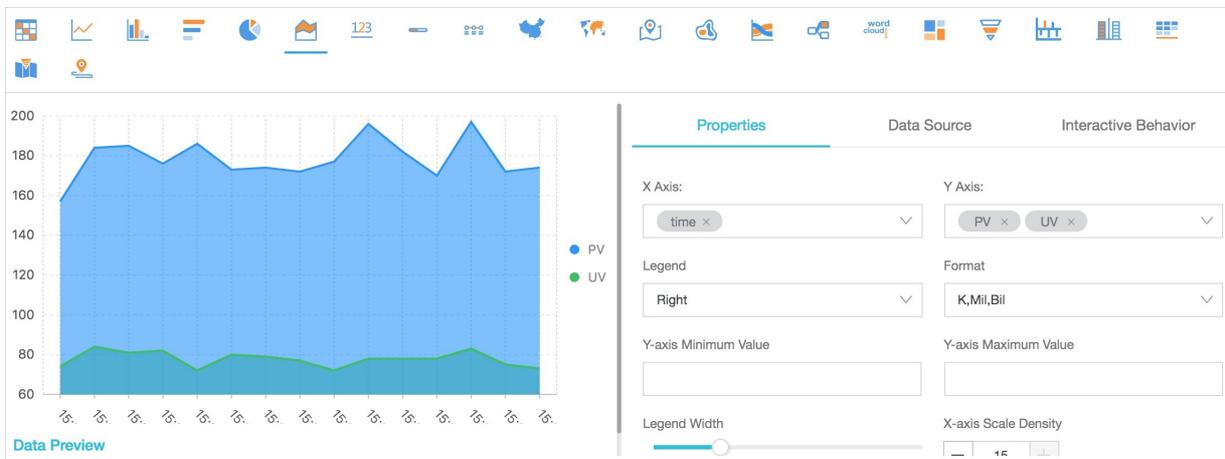
Select `time` for X Axis and `PV` for Y Axis.



Example of a stacked area chart

```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV, approx_distinct(remote_addr) as UV group by time order by time limit 1000
```

Select `time` for X Axis. Select `PV` and `UV` for Y Axis.



30.4.11.1.8. Individual value plot

The individual value plot highlights a single value.

Individual value plots include the following types:

- **Rectangle Frame:** shows a general value.
- **Dial:** shows how close the current value is to the configured threshold.
- **Compare Numb Chart:** shows the SQL query results of interval-valued comparison and periodicity-valued comparison functions. For more information about the analytic syntax, see [Interval-valued comparison and periodicity-valued comparison functions](#).

By default, Rectangle Frame is selected. A rectangle frame is the simplest way to display data at a point. In most cases, it is used to show the key information at a time point. To show a proportional metric, you can use a dial.

Components

- Numeric value
- (Optional) Unit
- (Optional) Description
- Chart type

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. On the Graph tab, click [123](#) to select the individual value plot.
3. On the Properties tab on the right, configure the properties of the individual value plot.

 **Note** Log Service automatically normalizes data in numeric value-based charts. For example, 230000 is processed as 230K. You can include [Mathematical calculation functions](#) in search and analytic statements to customize numeric formats.

Properties

- The following table lists the parameters of a rectangle frame.

Parameter	Description
Chart Types	The type of the chart. Select Rectangle Frame.
Value Column	The value displayed in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of data.
Unit Font Size	The font size of the unit. You can drag the slider to adjust the font size. Valid values: [10, 100]. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the value description. You can drag the slider to adjust the font size. Valid values: [10, 100]. Unit: pixels.
Format	The format in which data is displayed.
Font Size	The font size of the value. You can drag the slider to adjust the font size. Valid values: [10, 100]. Unit: pixels.

Parameter	Description
Font color	The color of the value, unit, and description in the chart. You can select the default color or customize a color.
Background Color	The color of the background. You can select the default color or customize a color.

- The following table lists the parameters of a dial.

Parameter	Description
Chart Types	The type of the chart. Select Dial to display query results in a dial.
Actual Value	The actual value in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of the value in the dial.
Font Size	The font size of the value and unit. Valid values: [10, 100]. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the value description. You can drag the slider to adjust the font size. Valid values: [10, 100]. Unit: pixels.
Dial Maximum	The maximum value of the scale in the dial. Default value: 100.
Maximum Value Column	The maximum value in the specified column. If you turn on the Use Query Results switch, Dial Maximum is replaced by Maximum Value Column. Then you can select the maximum value from query results for this parameter.
Use Query Results	If you turn on the Use Query Results switch, Dial Maximum is replaced by Maximum Value Column. Then you can select the maximum value from query results for this parameter.
Format	The format in which data is displayed.
Colored Regions	The number of segments that the dial is divided into. Each segment is displayed in a different color. Valid values: 2, 3, 4, and 5. Default value: 3.
Region Max Value	The number of segments that the dial is divided into. Each segment is displayed in a different color. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e6f2ff;"> <p> Note By default, a dial is evenly divided into three colored segments. If you change the value of Colored Regions, Region Max Value is not automatically adjusted. You can set the maximum value for each colored segment based on your needs.</p> </div>
Font color	The color of the value in the dial.
Region	The colored segment that the dial is divided into. By default, a dial is evenly divided into three segments, which are displayed in blue, yellow, and red. If you set Colored Regions to a value greater than 3, the added segments are displayed in blue by default. You can change the color of each segment.

- The following table lists the parameters of a comparison chart.

Parameter	Description
Chart Types	The type of the chart. Select Compare Numb Chart to display query results in a comparison chart.
Show Value	The value displayed in the center of the comparison chart. This value is set to the statistical result calculated by the relevant comparison function in the specified time range.
Compare Value	The value used to compare with the threshold. This value is set to the result of comparison between the statistical results calculated by the relevant comparison function in the specified time range and in the previously specified time range.
Font Size	The font size of the show value. Valid values: [10, 100]. Unit: pixels.
Unit	The unit of the show value.
Unit Font Size	The font size of the unit for the show value. Valid values: [10, 100]. Unit: pixels.
Compare Unit	The unit of the compare value.
Compare Font Size	The font size of the compare value and its unit. Valid values: [10, 100]. Unit: pixels.
Description	The description of the show value and its growth trends.
Description Font Size	The font size of the description. Valid values: [10, 100]. Unit: pixels.
Trend Comparison Threshold	<p>The value used to measure the variation trend of the compare value.</p> <p>For example, the compare value is -1:</p> <ul style="list-style-type: none"> ○ If you set Trend Comparison Threshold to 0, a down arrow is displayed on the page, indicating a value decrease. ○ If you set Trend Comparison Threshold to -1, the system determines that the value remains unchanged and does not display the trend on the page. ○ If you set Trend Comparison Threshold to -2, an up arrow is displayed on the page, indicating a value increase.
Format	The format in which data is displayed.
Font color	The color of the show value and its description.
Growth Font Color	The font color of the compare value that is greater than the threshold.
Growth Background Color	The background color displayed when the compare value is greater than the threshold.
Decrease Font Color	The font color displayed when the compare value is less than the threshold.
Decrease Background Color	The background color displayed when the compare value is less than the threshold.
Equal Background Color	The background color displayed when the compare value is equal to the threshold.

Examples

Run the following search and analytic statements to view the number of requests and display the analysis results in charts:

- Rectangle frame

* | select count(1) as pv

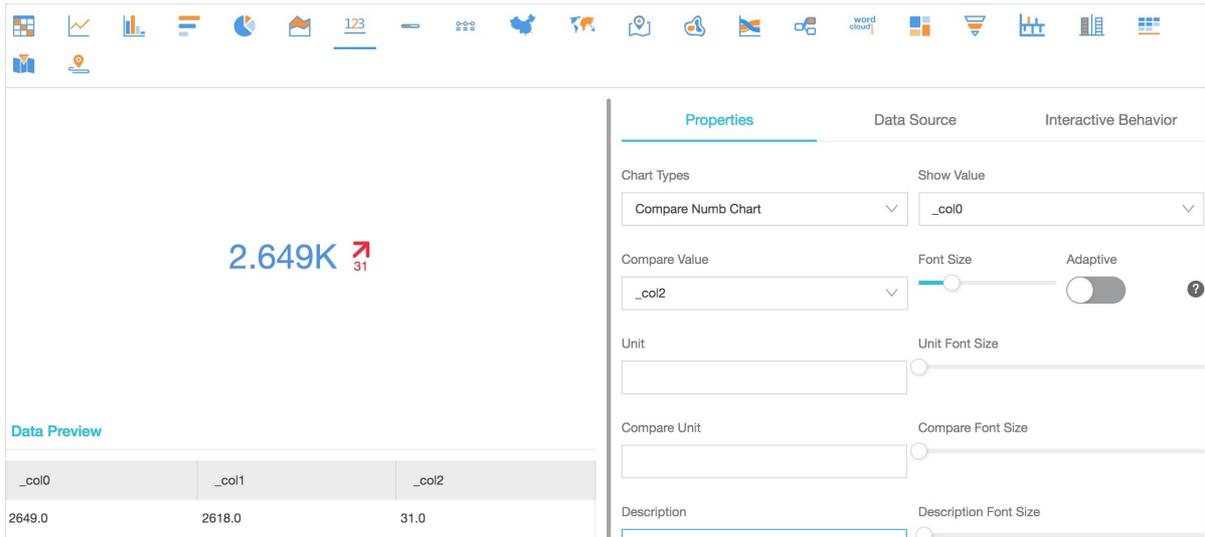
- Dial

* | select count(1) as pv

- Comparison chart

Run the following search and analytic statement to view the comparison of the requests made today and the requests made yesterday:

```
* | select diff[1],diff[2], diff[1]-diff[2] from (select compare( pv , 86400) as diff from (select count(1) as pv from log))
```



30.4.11.1.9. Progress bar

The progress bar shows the percentage of the actual value of a field to the maximum value of the field. You can configure the properties of the progress bar to adjust its style and set display rules.

Components

- Actual value
- (Optional) Unit
- Total value

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, specify the time range, and then click **Search & Analyze**.
2. Click  to select the progress bar.
3. Configure the properties of the progress bar.

Properties

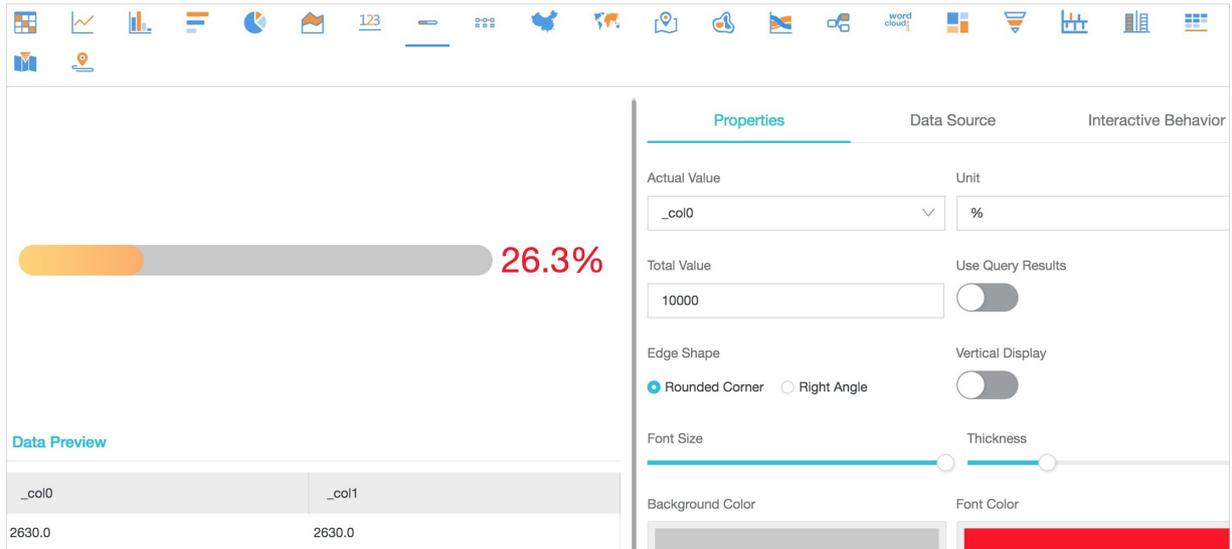
Parameter	Description
Actual Value	The actual value in the chart. By default, the data in the first row of the specified column is displayed.
Unit	The unit of the value in the progress bar.
Total Value	The maximum value indicated by the progress bar. Default value: 100.
Maximum Value Column	The maximum value in the specified column. If you turn on the Use Query Results switch, Total Value is replaced by Maximum Value Column . Then, you can select the maximum value from query results for this parameter.
Use Query Results	Specifies whether to select a value from query results. If you turn on the Use Query Results switch, you can select the maximum value from query results for Maximum Value Column .

Parameter	Description
Edge Shape	The edge shape of the progress bar.
Vertical Display	Specifies whether to display the progress bar in the vertical mode.
Font Size	The font size of the progress bar.
Thickness	The thickness of the progress bar.
Background Color	The background color of the progress bar.
Font Color	The font color of the progress bar.
Default Color	The default color of the progress bar.
Color Display Mode	The color display mode of the progress bar.
Start Color	The start color of the progress bar. This parameter is available when Color Display Mode is set to Gradient .
End Color	The end color of the progress bar. This parameter is available when Color Display Mode is set to Gradient .
Display Color	<p>The display color of the progress bar. This parameter is available when Color Display Mode is set to Display by Rule.</p> <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note The value of Actual Value is compared with that of Threshold based on the condition specified by Operator. If the actual value matches the condition specified by Operator, the progress bar is displayed in the color specified by Display Color. Otherwise, the progress bar is displayed in the default color.</p> </div>
Operator	The condition that is used to determine the color of the progress bar. This parameter is available when Color Display Mode is set to Display by Rule .
Threshold	The threshold that is used to determine the color of the progress bar. This parameter is available when Color Display Mode is set to Display by Rule .

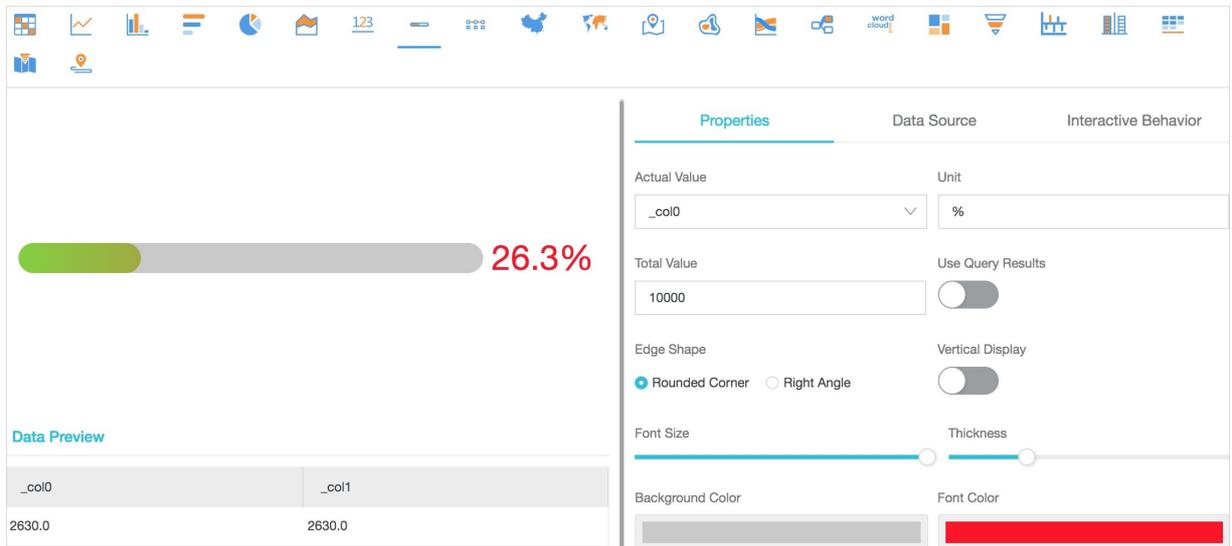
Examples

You can use the progress bar to show the percentage of a metric or the proportion of data.

```
* | select diff[1],diff[2] from (select compare( pv , 86400) as diff from (select count(1) as pv from log))
```



When you select **Display by Rule** as the color display mode, the progress bar changes its color according to the specified condition. If the specified condition is not met, the progress bar is displayed in the default color.



30.4.11.1.10. Map

You can add color blocks and marks to a map to display geographic data. Log Service provides three types of maps: Map of China, World Map, and AMap. Among them, AMap offers the scatter chart and heat map. You can use specific functions in search and analytic statements to display analysis results in different maps.

Components

- Map canvas
- Color block

Properties

Parameter	Description
-----------	-------------

Parameter	Description
Location information	The location information recorded in logs. The dimension varies depending on the type of the map: <ul style="list-style-type: none"> Provinces (Map of China) Country (World Map) Longitude/Latitude (AMap)
Value Column	The data volume of the location information.

Procedure

- On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, set the time range, and then click **Search & Analyze**.
 - Map of China: Use the `ip_to_province` function.
 - World Map: Use the `ip_to_country` function.
 - AMap: Use the `ip_to_geo` function.
- Click  to select the map.
- Configure the properties of the map.

Map of China

You can include the `ip_to_province` function to display query results in a map of China.

- SQL statement

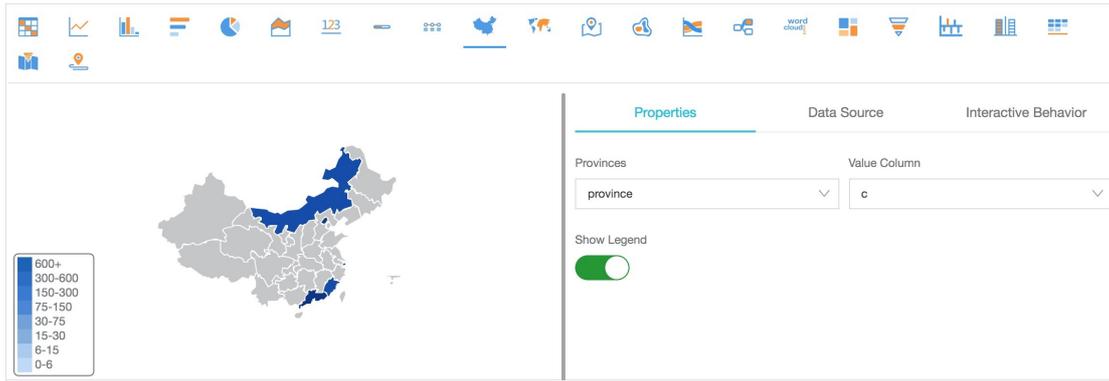
```
* | select ip_to_province(remote_addr) as address, count(1) as count group by address order by count desc limit 10
```

- Dataset

address	count
Guangdong	163
Zhejiang	110
Fujian	107
Beijing	89
Chongqing	28
Heilongjiang	19

Select address for *Provinces* and count for *Value Column*.

Map of China



World Map

You can include the `ip_to_country` function to display query results in a world map.

- SQL statement

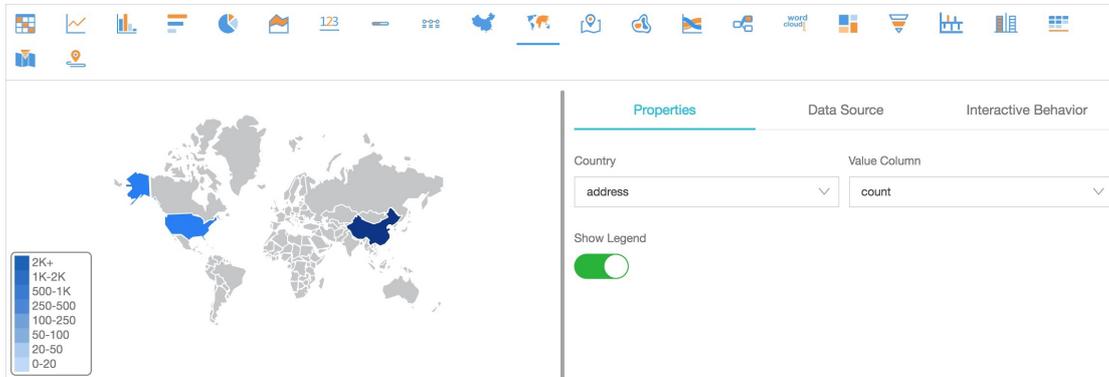
```
* | select ip_to_country(remote_addr) as address, count(1) as count group by address order by count desc limit 10
```

- Dataset

address	count
China	8354
United States	142

Select address for *Country* and count for *Value Column*.

World Map



AMap

You can include the `ip_to_geo` function in a statement to display query results in an AMap. The address column in the dataset contains the latitude and longitude information, which are separated with a comma (,). If the longitude and latitude are indicated by two separate columns named lng and lat, you can use the `concat('lat', ',', lng)` function to integrate the two columns into one column.

- SQL statement

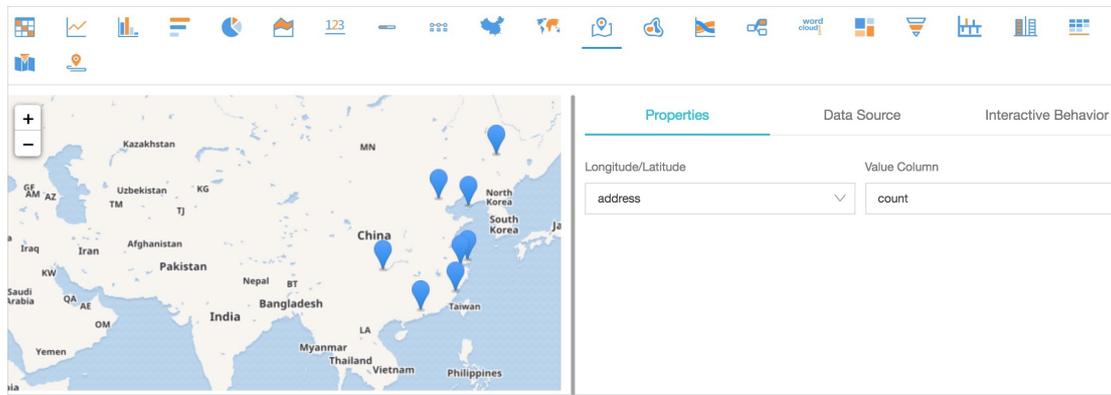
```
* | select ip_to_geo(remote_addr) as address, count(1) as count group by address order by count desc limit 10
```

- Dataset

address	count
39.9289,116.388	771
39.1422,117.177	724
29.5628,106.553	651
30.2936,120.161420	577
26.0614,119.306	545
34.2583,108.929	486

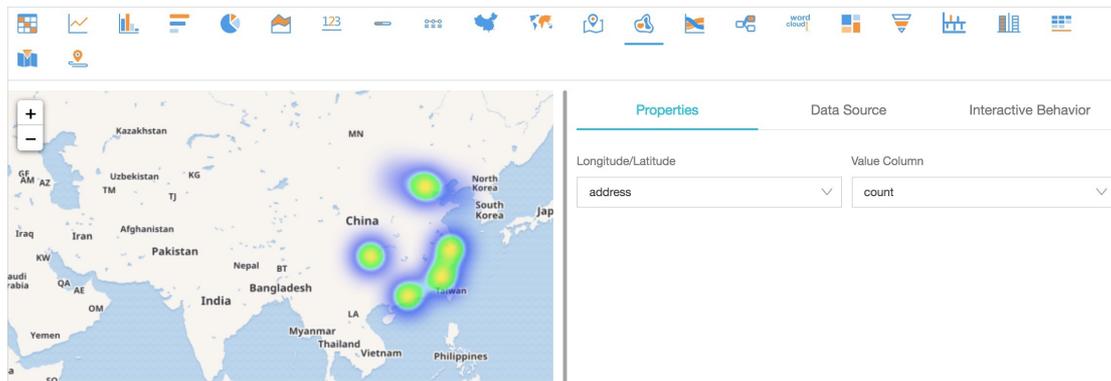
Select address for *Longitude/Latitude* and count for *Value Column*.

AMap: Scatter chart



The scatter chart is used by default. If data points are densely distributed in the map, you can convert it to the heat map.

AMap: Heat map



30.4.11.11. Flow diagram

The flow diagram, also known as ThemeRiver, is a stacked area chart around the central axis. The banded branches with different colors indicate different categorical data. The band width indicates the numeric value. The time information of the data is configured on the X-axis of the chart by default. A flow diagram displays data from three dimensions.

You can convert a flow diagram to a line chart or column chart. The column chart is stacked by default. Each category of data starts from the top of the last categorical data.

Components

- X-axis (horizontal)
- Y-axis (vertical)
- Band

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, set the time range, and then click Search & Analyze.
2. On the Graph tab, click  to select the flow diagram.
3. On the Properties tab on the right, configure the properties of the flow diagram.

Properties

Parameter	Description
Chart Types	The type of the chart. Valid values: Line Chart, Area Chart, and Column Chart. Default value: Line Chart.
X Axis	The data on the X-axis, which is usually a time sequence.
Y Axis	The numeric data. You can configure one or more fields on the Y-axis.
Aggregate Column	The information required to be aggregated in the third dimension.
Legend	The position where the legend is located in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

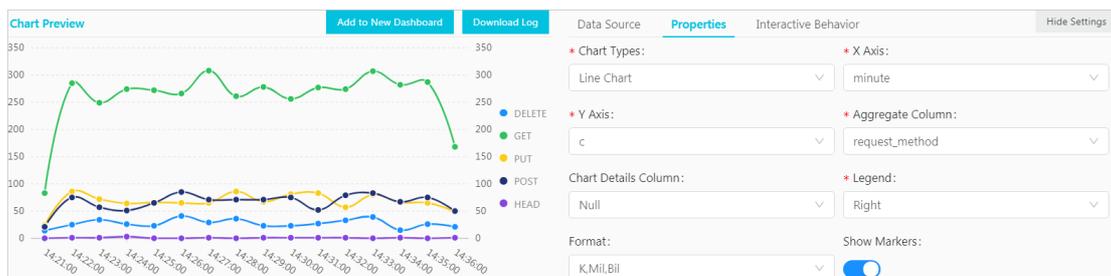
Examples

The flow diagram is suitable to display data from three dimensions, including the time information, categories, and numeric values.

```
* | select date_format(from_unixtime(__time__ - __time__% 60), '%H:%i:%S') as minute, count(1) as c, request_method group by minute, request_method order by minute asc limit 100000
```

Select `minute` for X Axis, `c` for Y Axis, and `request_method` for Aggregate Column.

Flow chart



30.4.11.1.12. Sankey diagram

A sankey diagram is a specific type of flow chart. It is used to describe the flow from one set of values to another.

Sankey diagrams are applicable to scenarios such as network traffic flows. A sankey diagram contains values of three fields: `source`, `target`, and `value`. The `source` and `target` fields describe the source and target nodes and the `value` field describes the flows from the `source` node to the `target` node.

Features

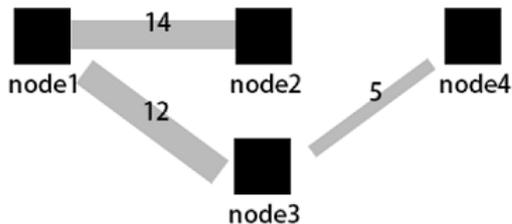
The sankey diagram has the following features:

- The start flow is as large as the end flow. The overall width of all main edges is the same as that of all branch edges. This maintains flow balance.
- In a sankey diagram, different edges indicate the distribution of different flows. The width of an edge is proportional to the flow volume it represents.
- The width of an edge between two nodes represents the flow volume in a state.

For example, the following data can be displayed in a sankey diagram.

source	target	value
node1	node2	14
node1	node3	12
node3	node4	5
...

The sankey diagram in the following figure shows the relationship between the data.



Components

- Node
- Edge

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, set the time range, and then click Search & Analyze.
2. Click  to select the sankey diagram.
3. On the Properties tab on the right, configure the properties of the area chart.

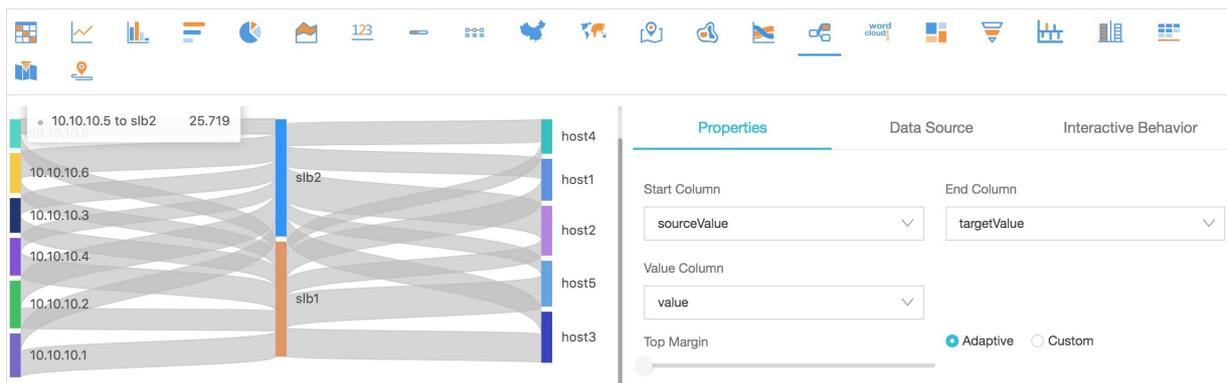
Properties

Parameter	Description
Start Column	The start node.
End Column	The end node.
Value Column	The value that indicates the flow volume from the start node to the end node.
Margin	The distance of the axis to the borders of the chart, including Top Margin, Bottom Margin, Right Margin, and Left Margin.

Example of a simple sankey diagram

If a log entry contains the `source`, `target`, and `value` fields, you can use a **Nested subqueries** statement to obtain the sum of all `streamValue` values.

```
* | select sourceValue, targetValue, sum(streamValue) as streamValue from (select sourceValue, targetValue, streamValue, __time__ from log group by sourceValue, targetValue, streamValue, __time__ order by __time__ desc) group by sourceValue, targetValue
```



30.4.11.1.13. Word cloud

A word cloud visualizes text data. It is a cloud-like and colored image composed of words. It can be used to display a large amount of text data. The font size or color of a word indicates the significance of the word. This allows you to recognize the most significant words in an efficient way.

Components

Sorted words

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, set the time range, and then click **Search & Analyze**.
2. Click  to select the word cloud.
3. On the Properties tab on the right, configure the properties of the word cloud.

Parameters

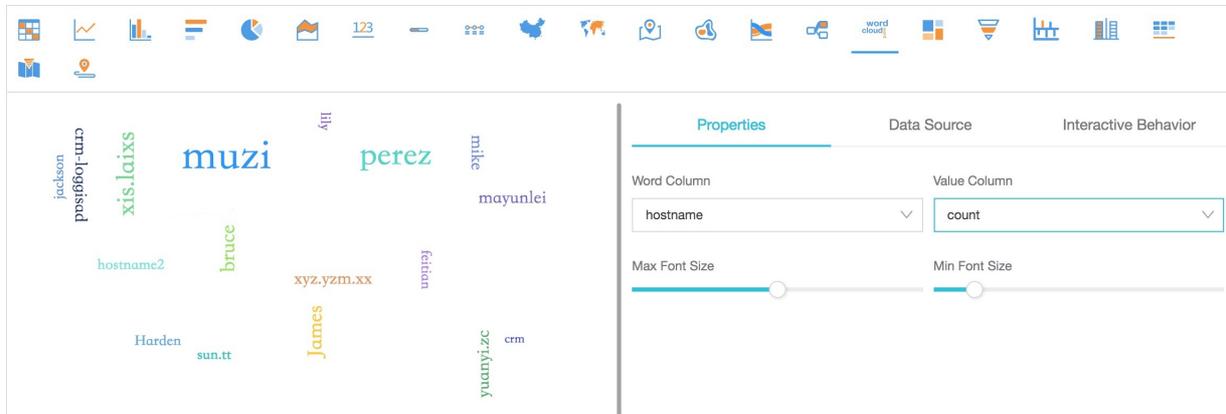
Parameter	Description
Word Column	The words to be displayed.
Value Column	The numeric value corresponds to a word.
Font Size	The font size of a word. <ul style="list-style-type: none"> The minimum font size ranges from 10 pixels to 24 pixels. The maximum font size ranges from 50 pixels to 80 pixels.

Examples

To query the distribution of hostnames in NGINX logs, run the following statement:

```
* | select domain, count(1) as count group by domain order by count desc limit 1000
```

Select **hostname** for Word Column and **count** for Value Column.



30.4.11.1.14. Treemap chart

A treemap chart includes multiple rectangles that represent the data volume. A larger rectangle area represents a larger proportion of the categorical data.

Components

Sorted rectangles

Procedure

1. On the Search & Analysis page of a Logstore, enter a search and analytic statement in the search box, set the time range, and then click **Search & Analyze**.
2. On the Graph tab, click  to select the bar chart.
3. On the Properties tab on the right, configure the properties of the area chart.

Properties

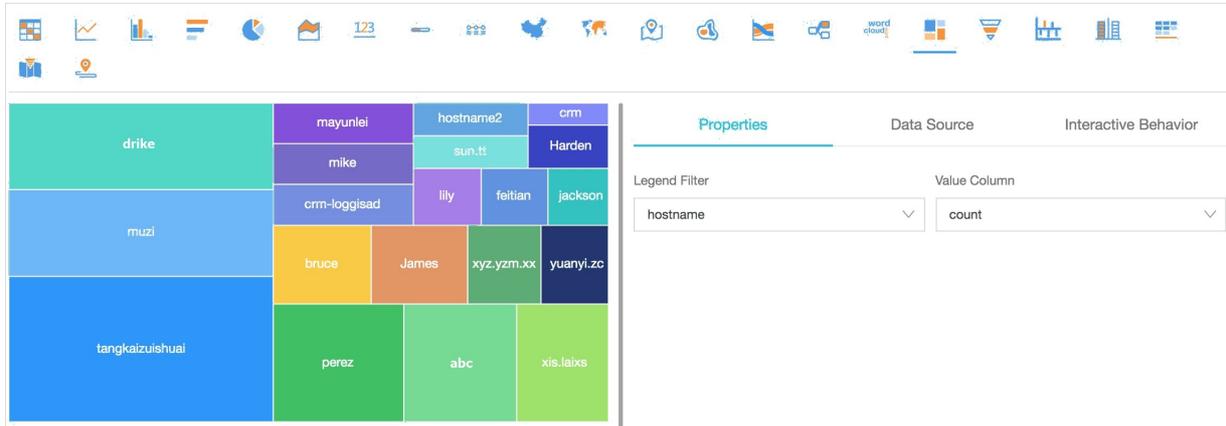
Parameter	MNS logs
Description	The categorical field.
Value Column	The numeric value of a field. A greater field value results in a larger rectangle area.

Example

To query the distribution of hostnames in NGINX logs, run the following statement:

```
* | select hostname, count(1) as count group by hostname order by count desc limit 1000
```

Select **hostname** for **Legend Filter** and **count** for **Value Column** .



30.4.11.2. Dashboard

30.4.11.2.1. Overview

A dashboard provided by Log Service is a platform where you can analyze data in real time. You can add multiple charts to a dashboard for data analysis. Each chart is a visualized search and analytic statement.

A dashboard allows you to view the charts of multiple search and analytic statements at one time. When you open or refresh the dashboard, the statements of the charts run automatically.

After you add a chart to a dashboard, you can configure **Drill-down analysis** for the chart. Then you can click the chart on the dashboard to further analyze data and obtain more fine-grained analysis results.

Limits

- You can create a maximum of 50 dashboards for a project.
- Each dashboard can contain a maximum of 50 analysis charts.

Features

A dashboard has two modes: display mode and edit mode.

- **Configure the display mode of a dashboard**

In the display mode, you can configure multiple display settings on the dashboard page.

- **Dashboard:** You can specify the time range, the automatic refresh interval, full screen, and the display mode of the title for the dashboard, configure alerts for all charts on the dashboard, and filter chart data based on the **Configure and use a filter on a dashboard of a Logstore**.

- **Chart:** You can view the analysis details of a specified chart, specify the time range and configure alerts for the chart, download logs and the chart, and check whether **drill-down** analysis is configured for the chart.

- **Edit mode**

In the edit mode, you can change the configurations of the dashboard and charts.

- **Dashboard:** You can use a dashboard as a canvas and add [Markdown chart](#), custom charts, text, icons, and other chart elements to the dashboard. You can also add lines between chart elements that are self-adaptive to the positions of the charts. You can also add [Configure and use a filter on a dashboard of a Logstore](#), which can be used to filter chart data in the display mode. In addition, you can configure display gridlines to help arrange chart elements such as icons in an orderly manner.
- **Chart:** You can also edit a chart on the dashboard. You can modify the statement, properties, and interactive behavior such as [drill-down analysis](#) of the chart.

30.4.11.2.2. Create and delete a dashboard

This topic describes how to create and delete a dashboard in a Logstore. In the Log Service console, you can run a search and analytic statement and visualize the query result in a chart. After you complete the configurations of the chart, you can add the chart to a dashboard. Each dashboard can display up to 50 charts that support multiple formats and custom settings.

Prerequisites

The index feature of the Logstore is enabled and configured. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Create a dashboard

1. [Log on to the Log Service console](#).
2. Click a project name.
3. Click the  icon next to the name of the Logstore, and then select **Search & Analysis**.
4. Enter a search and analytic statement in the search box, and then click **Search & Analyze**.
5. On the **Graph** tab that appears, configure the chart properties on the **Properties** tab.
6. (Optional) Set a placeholder variable. For example, you have configured a drill-down event for another chart. This drill-down event redirects you to the current dashboard. You have also specified a placeholder variable for the search and analytic statement of the preceding chart. When you click a chart value to trigger the drill-down event, you are redirected to the current dashboard. The placeholder variable is replaced with the chart value and the current dashboard is refreshed by the new search and analytic statement. For more information, see [Drill-down analysis](#).
 - i. Click the **Data Source** tab, and then select a part of the search and analytic statement in the **Query** field.
 - ii. Click **Generate Variable** to generate a placeholder variable.

iii. Set the parameters in the Variable Config section.

Parameter	Description
Variable Name	The name of the placeholder variable. If the name of the placeholder variable is the same as the variable specified in the chart, the placeholder variable will be replaced with the chart value when the drill-down event is triggered.
Default Value	The default value of the placeholder variable in the current dashboard.
Matching mode	You can select Global Match or Exact Match.
Result	The search and analytic statement that contains the specified variable.

Data Source
Properties
Interactive Behavior
Hide Settings

Query: Generate Variable

```
* | SELECT date_format(__time__ - __time__ % 60, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```

Select the query statement to generate a placeholder variable. You can configure a drill-down configuration to replace the variable. For how to use dashboards, please refer to the documentation ([Help](#))

Variable Config:

* Variable Name:

* Default Value:

* Matching Mode:

interval

60

Global Match v X

Result

```
* | SELECT date_format(__time__ - __time__ % $(interval), '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```

...
☰

7. Configure a drill-down event. After you configure a drill-down event, you can click the chart on the dashboard for a deeper analysis. For example, you can be redirected to another dashboard or a saved search. For more information, see [Drill-down analysis](#).

- i. Click the Interactive Behavior tab.
- ii. Select an Event Action.

iii. Set the parameters of the selected event action.

8. Click **Add to New Dashboard** and specify the dashboard name and chart name.

Parameter	Description
Operation	<ul style="list-style-type: none"> ◦ Add to Existing Dashboard: Add the chart to an existing dashboard. ◦ Create Dashboard: Create a dashboard and then add the chart to the dashboard.
Dashboards	Select an existing dashboard name. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ⓘ Note This parameter is required only when you set the Operation parameter to Add to Existing Dashboard. </div>
Dashboard Name	Enter a dashboard name. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> ⓘ Note This parameter is required only when you set the Operation parameter to Create Dashboard. </div>
Chart Name	Enter a name for the current chart. The chart name is displayed as the chart title in the dashboard.

9. Click **OK**. You can add up to 50 analysis charts to the dashboard.

Delete a dashboard

You can delete a dashboard when you no longer need it. You cannot recover a deleted dashboard.

1. Log on to the Log Service console, and then click the target project name.
2. In the left-side navigation pane, click the **Dashboard** icon.
3. Click the  icon next to the dashboard that you want to delete, and then select **Delete**.

30.4.11.2.3. Configure the display mode of a dashboard

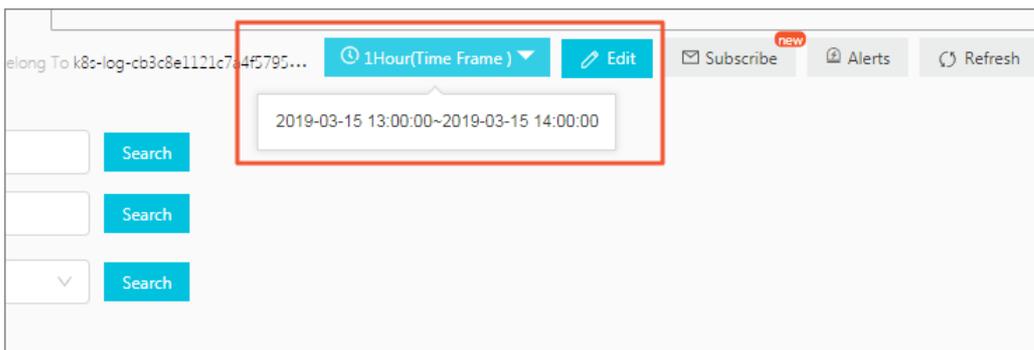
This topic describes how to configure the display mode of a dashboard. By default, you can view all charts in a dashboard in the display mode. When you configure the display mode, you can add chart elements, enable automatic refresh, and set the title display mode.

Set a query time range for a dashboard of a Logstore

By default, all charts in a dashboard use the query time range that is set for the dashboard. For more information about how to set a query time range for a single chart, see [Set a query time range for a chart](#).

Note On the dashboard page, you can click **Time Range** to specify a time range for a query. The specified query time range is used only for the current query. The next time you open the dashboard, the system will display the analysis results in the default query time range.

1. [Log on to the Log Service console](#).
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. Click the  icon next to the dashboard, and then select **Details** from the drop-down list.
5. Click **Time Range** to set a time range. You can set one of the following time ranges:
 - **Relative**: queries log data obtained in a time range of 1 minute, 5 minutes, 15 minutes, or other time ranges that end with the current time, accurate to the second. For example, if the current time is 19:20:31 and you select 1Hour as the relative time, the charts on the dashboard display the analysis results of the log data queried from 18:20:31 to 19:20:31.
 - **Time Frame**: If you select or customize a time range less than one hour (for example, 1 minute, 5 minutes, and 15 minutes), log data obtained in the time range that ends with the current time is queried, accurate to the minute. If you select or customize a time range greater than one hour, log data obtained on the hour before the current time is queried. For example, if the current time is 19:20:31 and you select 1Hour as the time frame, the charts on the dashboard display the analysis results of the log data queried from 18:00:00 to 19:00:00.
 - **Custom**: queries log data obtained in a specified time range.
6. Move the pointer over the **Time Range** button to confirm the specified time range.



Switch to the edit mode

Click **Edit** to switch to the edit mode of the dashboard. In the edit mode, you can add [Markdown chart](#), custom charts, text, icons, and other chart elements to the dashboard. For more information, see [Edit mode](#).

Set alerts

On the dashboard page, choose **Alerts > Create** to create an alert. Choose **Alerts > Modify** to modify an alert. An alert must be associated with one or more charts.

For more information, see [Configure alerts](#).

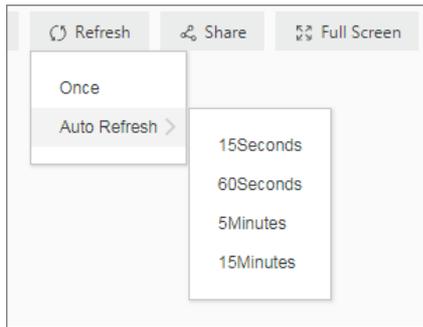
Set a refresh method

You can manually refresh the dashboard, or set an automatic refresh interval for the dashboard.

- To manually refresh the dashboard, choose **Refresh > Once**.
- To set an automatic refresh interval for the dashboard, choose **Refresh > Auto Refresh**, and then select an interval.

The **Auto Refresh** interval can be 15 seconds, 60 seconds, 5 minutes, or 15 minutes.

Note If your browser is inactive, the automatic refresh interval may be inaccurate.



Share a dashboard

To share a dashboard with authorized users, click **Share** to copy the link of the dashboard page and then send the link to the users. The shared dashboard page uses the settings of the dashboard at the time of sharing. The settings include the time range of charts and chart title format.

Note Before you share the dashboard with other users, you must grant relevant permissions to them.

Display a dashboard in full screen

Click **Full Screen**. Then the charts on a dashboard are displayed in full screen.

Set the chart title format

On the dashboard page, click **Title Configuration**. Available title formats include:

- Single-line Title and Time Display
- Title Only
- Time Only

Reset the query time range

To restore the default query time ranges of all charts on the dashboard, click **Reset Time**.

Select chart view

- View analysis details of a chart

To view analysis details of a chart, move the pointer over the **More** icon in the upper-right corner of the chart, and then select **View Analysis Details**. The corresponding **Search & Analysis** page appears, showing the query statement and property settings.

- Set the query time range for a chart

To set the query time range for a chart, move the pointer over the **More** icon in the upper-right corner of the chart, and then select **Select Time Range**. The settings are valid only for the current chart.

- Set an alert for a chart

To set an alert for a chart, move the pointer over the **More** icon in the upper-right corner of the chart, and then select **Create Alert**. For more information, see [Configure alerts](#).

- Download log analysis results of a chart

To download analysis results of a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Download Log**.

- Download a chart

To download a chart, move the pointer over the More icon in the upper-right corner of the chart, and then select **Download Chart**.

- Check whether a drill-down event is configured for a chart

To check whether a drill-down event is configured for a chart, move the pointer over the More icon in the upper-right corner of the chart. Then, check the color of the hand icon at the bottom of the shortcut menu. If the icon is red, a drill-down event is configured for the chart. If the icon is gray, no drill-down event is configured for the chart.

 **Note** Different charts in a dashboard have different shortcut menus. For example, you cannot use the shortcut menu of a custom chart or Markdown chart to view analysis details because they are not analysis charts.

30.4.11.2.4. Edit mode

You can click the Edit button on the dashboard page to enter the edit mode. Then you can change the configurations of the dashboard and charts on the dashboard.

- Dashboard:

- You can modify the dashboard name in the upper-left corner of the page.
- You can use a dashboard as a canvas and add **Markdown chart**, custom charts, text, icons, and other chart elements to the dashboard.
- You can add lines between chart elements. The lines are self-adaptive to the positions of the charts.
- You can add a filter to the dashboard that can filter chart data in the display mode. For more information, see **Configure and use a filter on a dashboard of a Logstore**.
- In addition, you can configure display gridlines to help arrange chart elements such as icons in an orderly manner.
- You can use the menu bar to control the chart property settings in the dashboard. For example, you can perform addition, deletion, and cancellation operations and configure the size and location of charts. You can also move a chart to the top of bottom of the dashboard.

- Chart: You can edit a chart on the dashboard. You can modify the statement, properties, and interactive behavior such as **drill-down analysis** of the chart.

 **Note** You must click Save in the upper-right corner of the page for the changes made to the dashboard to take effect.

Chart elements

In the edit mode of a dashboard, you can add the following chart elements:

- Common icons

Log Service allows you to display common icons on a dashboard page. You can drag an icon from the menu bar to the specified position.

- Text

You can drag the text icon from the menu bar to the specified position to insert text. Double-click the text box to modify the text content.

- **Markdown chart**

You can add a Markdown chart to a dashboard and edit the chart with the Markdown syntax.

Drag the Markdown icon from the menu bar to the specified position and select **Edit** from the **More** icon. Then you can set the Markdown content.

- **Filter**

You can add a filter to a dashboard. Then you can add search conditions or replace placeholder variables in query statements.

Click the filter icon in the dashboard menu bar. On the page that appears, configure the filter based on your needs. The filter is in the upper-left corner of the dashboard page by default. To modify the settings of a filter, you can select **Edit** on the **More** icon in the upper-right corner of the page.

- **Customize SVG**

You can upload a Scalable Vector Graphics (SVG) file to a dashboard. Click the SVG icon in the menu bar. On the page that appears, click the box or drag an SVG file to the box to upload the file.

 **Note** The size of an SVG file cannot exceed 10 KB.

- **Customize image's HTTP link**

You can upload the HTTP link of an image to a dashboard. Click the **Customize image's HTTP link** icon in the menu bar. On the page that appears, enter the HTTP link of an image and click **OK**.

Layout

In the edit mode, all charts and chart elements are placed in a canvas. You can drag and scale each chart, except for the lines. The horizontal width of a chart is up to 1,000 units, with each unit equal to $\frac{\text{current browser width}}{1,000}$. The vertical height is unlimited, with each unit equal to 1 pixel. Before you arrange a chart on the dashboard, you can click **Display gridlines** to help arrange the position of the chart and the spacing with other charts.

You can perform the following operations to arrange a chart on the dashboard:

- **Adjust the position of a chart**
 - You can drag a chart to the specified position.
 - After you specify a chart, you can click **L** and **T** to adjust the chart position.
- **Adjust the width and height of a chart**
 - After you specify a chart, you can drag the chart in the lower-right corner to resize the chart.
 - After you specify a chart, you can also specify the **W** and **H** to resize the chart.

- **Add lines to connect charts**

You can add a directional line between two charts. When you adjust the position and size of the charts, the line automatically moves to display the relative position between the two charts.

After you specify a chart, four box marks appear on the border of the chart. These marks indicate the starting point of the connection line. Press and hold one box mark, and the area where the end point of the connection line is automatically displayed. Move the pointer to this position to connect the two charts.

- You can move a chart to the top or bottom of the dashboard. After you select a chart, you can use the menu bar to move the chart to the top or bottom.

Chart configurations

In the edit mode of a dashboard, you can perform the following operations on chart elements:

- **Edit**: modifies the query statement, properties, and interactive behavior such as **drill-down analysis** of a chart.
 - In the upper-right corner of the dashboard page, click **Edit**.
 - Find the **More** icon in the upper-right corner of the chart, and click **Edit**.
 - Modify the query statement of the chart, **Properties**, **Data Source**, or **Interactive Behavior**.
 - Click **Preview**, and then click **OK**.

- v. In the upper-right corner of the dashboard page, click **Save**.
- **Copy:** creates a copy of the specified chart and retains all configurations.
 - i. In the upper-right corner of the dashboard page, click **Edit**.
 - ii. Find the **More** icon in the upper-right corner of the chart, and click **Copy**.
 - iii. Drag the chart copy to the specified position and set the top and left margins and size of the copy.
 - iv. In the upper-right corner of the dashboard page, click **Save**.
- **Delete:** deletes the specified chart from the dashboard.
 - i. In the upper-right corner of the dashboard page, click **Edit**.
 - ii. Find the **More** icon in the upper-right corner of the chart, and click **Delete**.
 - iii. In the upper-right corner of the dashboard page, click **Save**.

30.4.11.2.5. Drill-down analysis

This topic describes how to configure drill-down analysis for a chart of a Logstore. When you add a chart to a dashboard, you can modify the configurations in the drill-down list to obtain deeper data analysis results.

Prerequisites

- The index feature of the Logstore is enabled and configured. For more information, see [Enable the index feature and configure indexes for a Logstore](#).
- A saved search, a dashboard, and a custom link to which you want to be redirected are configured.
- A placeholder variable is specified in the saved search or the query statement of a chart added to a dashboard to which you are redirected if you want to add a variable. For more information, see [Saved search](#) and [Create and delete a dashboard](#).

Context

Drilling is essential for data analysis. This feature allows you to analyze data in a fine-grained or coarse-grained way. This feature includes drill-up and drill-down analysis. You can implement drill-down analysis to gain a deeper insight into data and make a better decision.

Log Service supports drill-down analysis of the following charts: tables, line charts, column charts, bar charts, pie charts, individual value plots, area charts, and treemap charts.

Procedure

1. [Log on to the Log Service console](#).
2. Click the target project name.
3. Click the  icon next to the name of the Logstore, and then select **Search & Analysis** from the drop-down list.
4. Enter a query statement in the search box, set a time range, and then click **Search & Analyze**.
5. On the **Graph** tab that appears, select a **chart type**, and then configure the parameters on the **Properties** tab of the chart.
6. Click the **Interactive Behavior** tab. On this tab, configure the **Event Action** for drill-down analysis.
 - **Disable:** disables drill-down analysis.
 - **Open Logstore:** sets the drill-down event to open a Logstore.

If you configure a filter statement on the **Interactive Behavior** tab, the filter statement automatically fills the **Search & Analyze** search box of the redirected Logstore page when you click a value on the chart.

Parameter	Description
Select Logstore	The name of the Logstore to which you want to be redirected.

Parameter	Description
Open in New Tab	If you turn on this switch, the specified Logstore is opened on a new tab when the interactive behavior is triggered.
Time Range	<p>The query time range of the Logstore to which you are redirected. Valid values:</p> <ul style="list-style-type: none"> ▪ Default: The default time range (15 minutes, accurate to the second) is used for a query statement of the redirected Logstore. ▪ Inherit table time: The time range that a statement queries in the redirected Logstore is the time range specified for the chart when the event is triggered. ▪ Relative: The time range that a statement queries in the redirected Logstore is accurate to the second. ▪ Time Frame: The time range that a statement queries in the redirected Logstore is accurate to the minute, hour, or day. <p>Default value: Default.</p>
Inherit Filtering Conditions	If you turn on the Inherit Filtering Conditions switch, the filtering conditions added to the dashboard are synchronized to a query statement of the specified Logstore. The filtering conditions are added before the query statement by using the logical AND operator.
Filter	<p>Enter a Filter Statement on the Filter tab. The filter statement can contain the Optional Parameter Fields.</p> <p>If you configure a filter statement on the Filter tab, the filter statement automatically fills the Search & Analyze search box of the redirected Logstore page when you click a chart value on the dashboard.</p>

- **Open Saved Search:** sets the drill-down event to execute a saved search.

You can configure variables and a filter statement for a chart at the same time. When you click a value on the chart:

- If you configure a variable for the chart, the variable value that you click replaces the placeholder variable that you have configured for the saved search and the saved search is executed for drill-down analysis.
- If you configure a filter statement, the filter statement is added to the saved search to which you are redirected.

Parameter	Description
Select Saved Search	The name of the saved search to which you want to be redirected. For more information about how to configure a saved search, see Saved search .
Open in New Tab	If you turn on this switch, the specified saved search is opened on a new tab when the interactive behavior is triggered.
Time Range	<p>The time range of the saved search to which you want to be redirected. Valid values:</p> <ul style="list-style-type: none"> ▪ Default: The default time range (15 minutes, accurate to the second) is used for the saved search to which you are redirected. ▪ Inherit table time: The time range of the saved search is the query time range of the chart that you configure on the dashboard. ▪ Relative: The time range of the saved search is accurate to the second. ▪ Time Frame: The time range of the saved search is accurate to the minute, hour, or day. <p>Default value: Default.</p>

Parameter	Description
Inherit Filtering Conditions	If you turn on the Inherit Filtering Conditions switch, the filtering conditions added on the dashboard are synchronized to the saved search of the specified Logstore. The filtering conditions are added before the saved search by using the logical AND operator.
Filter	Click the Filter tab, and then enter a Filter Statement . The filter statement can contain the Optional Parameter Fields . If you configure a filter statement on the Filter tab, the Filter Statement is added to the saved search when you click a chart value on the dashboard.
Variable	Click the Variable tab, click Add Variable , and then set the following parameters: <ul style="list-style-type: none"> ▪ Replace Variable Name: the variable that triggers the drill-down event. When you click this variable, you are redirected to the specified saved search. ▪ Replace the value in the column: the column where the value that you want to replace data with is located. To process multiple columns, you can specify the current column and other columns. The current column is the column on which you want to perform drill-down analysis. Specify the current column in the Replace the value in the column field. Other columns can be the fields in the chart for which you configure drill-down analysis. If the variable name of the saved search is the same as the added variable, the variable of the saved search is replaced with the chart value that triggers the drill-down event. This helps you use the saved search for analysis in an efficient way. <div style="background-color: #e1f5fe; padding: 10px; margin-top: 10px;"> <p> Note</p> <ul style="list-style-type: none"> ▪ If you add a variable for drill-down analysis, you must first configure a placeholder variable for the saved search to which you want to be redirected. ▪ You can add up to five placeholder variables for a saved search. </div>

- **Open Dashboard**: sets the drill-down event to open a dashboard.

Analysis charts on a dashboard are visualized query results. When you click a chart value on the source dashboard:

- If you configure a variable for the chart and set a placeholder variable for the chart on the destination dashboard to which you want to be redirected, the placeholder variable is replaced with the chart value that you click.
- If you configure a filter statement, the filter statement is added to the query statement of the chart on the destination dashboard.

Parameter	Description
Select Dashboard	The name of the dashboard to which you want to be redirected. For more information, see Create and delete a dashboard .
Open in New Tab	If you turn on this switch, the specified dashboard is opened on a new tab when interactive behavior is triggered.

Parameter	Description
Time Range	<p>Set the time range for the dashboard to which you want to be redirected. Valid values:</p> <ul style="list-style-type: none"> ▪ Default: After you are redirected to the dashboard by clicking a chart value on the current dashboard, the selected dashboard uses its original time range. ▪ Inherit table time: After you are redirected to the selected dashboard, the time range of the chart on the selected dashboard is the time range of the chart specified on the source dashboard when the event is triggered. ▪ Relative: After you are redirected to the selected dashboard, set the time range of the selected dashboard to the specified relative time. ▪ Time Frame: After you are redirected to the selected dashboard set the time range of the selected dashboard to the specified time frame. <p>Default value: Default.</p>
Inherit Filtering Conditions	<p>If you turn on the Inherit Filtering Conditions switch, the filtering conditions added on the current dashboard are synchronized to the dashboard to which you are redirected. The filtering conditions are added before the query statement by using the logical AND operator.</p>
Filter	<p>Click the Filter tab, and then enter a Filter Statement. The filter statement can contain the Optional Parameter Fields.</p> <p>If you have set the Filter, a filtering condition is added to the selected dashboard when you click a chart value on the current dashboard. The filtering condition is the specified Filter Statement.</p>
Variable	<p>Click the Variable tab, click Add Variable, and then set the following parameters:</p> <ul style="list-style-type: none"> ▪ Replace Variable Name: the variable that triggers drill-down analysis. When you click this variable, you are redirected to the selected dashboard. ▪ Replace the value in the column: the column where the value that you want to replace data with is located. To process multiple columns, you can specify the Default Column and other columns. The Default Column is the current column on which you want to perform drill-down analysis. Other columns can be fields in the chart for which you configure drill-down analysis. <p>If the variable in the query statement of the chart on the selected dashboard is the same as the added variable, the variable is replaced with the chart value that triggers the drill-down event. This changes the query statement of the chart on the selected dashboard.</p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p> Note</p> <ul style="list-style-type: none"> ▪ If you add a variable for drill-down analysis, you must first configure a placeholder in the query statement for the selected dashboard to which you want to be redirected. ▪ You can add up to five variables. </div>

- **Custom HTTP Link:** sets the drill-down event to open a custom HTTP link.

The path in the HTTP link is the path of the destination file in the server. When you add optional parameter fields to the path and click the chart value on the dashboard, the added parameter fields are replaced with the chart value. At the same time, you are redirected to the new HTTP link.

Parameter	Description
Enter Link	The destination address to which you want to be redirected.
Optional Parameter Fields	By clicking an optional parameter variable, you can replace a part of the HTTP link with the chart value that triggers a drill-down event.

7. Click **Add to New Dashboard** and set the dashboard name and chart name.

Example

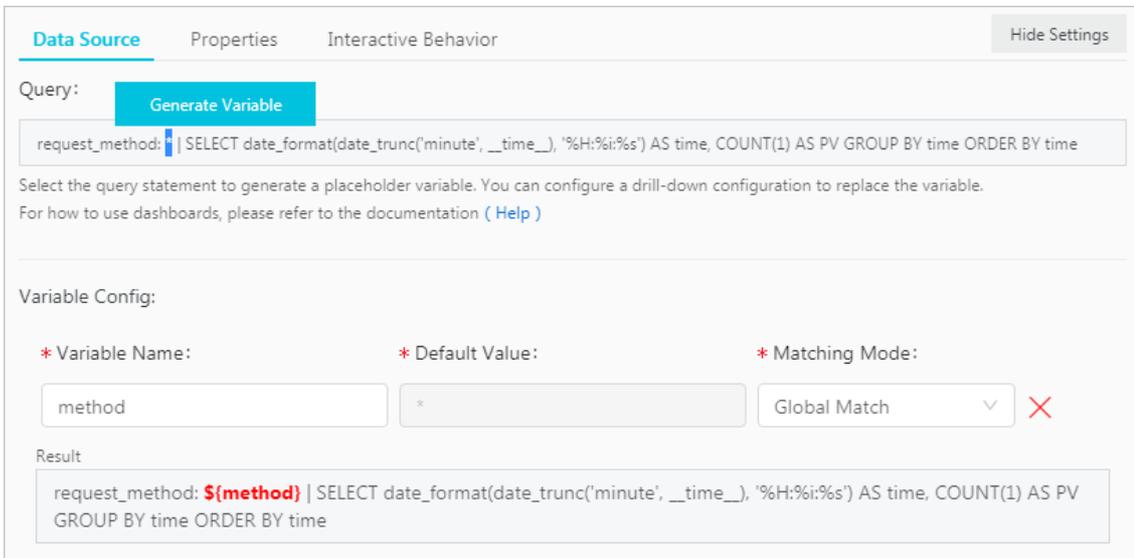
You want to store collected NGINX access logs in a Logstore named `accesslog`, display the relevant analysis results on a dashboard named `RequestMethod`, and display the page view (PV) changes over time on a dashboard named `destination_drilldown`. You can configure drill-down analysis for the table of request methods, add the table to the `RequestMethod` dashboard, and then configure the drill-down event that redirects you to the `destination_drilldown` dashboard. When you click a request method on the table on the `RequestMethod` dashboard, you are redirected to the `destination_drilldown` dashboard. Then you can view the PV changes on the dashboard.

1. Set the dashboard to which you want to be redirected.

- i. Filter log data by request method and analyze how the PV of each request method changes over time. The query statement is shown as follows:

```
request_method: * | SELECT date_format(date_trunc('minute', __time__), '%H:%i:%s') AS time, COUNT(1) AS PV GROUP BY time ORDER BY time
```

- ii. Use a line chart to display the query result and add the line chart to the dashboard named `destination_drilldown`. Before you add the line chart to the dashboard, specify the asterisk (`*`) to generate a placeholder variable and set the variable name to `method`. If the variable name of another chart for which you configure drill-down analysis is `method`, when you click the variable value on the chart to trigger drill-down analysis, the asterisk (`*`) is replaced with the value that you click and the query statement of the line chart is performed.



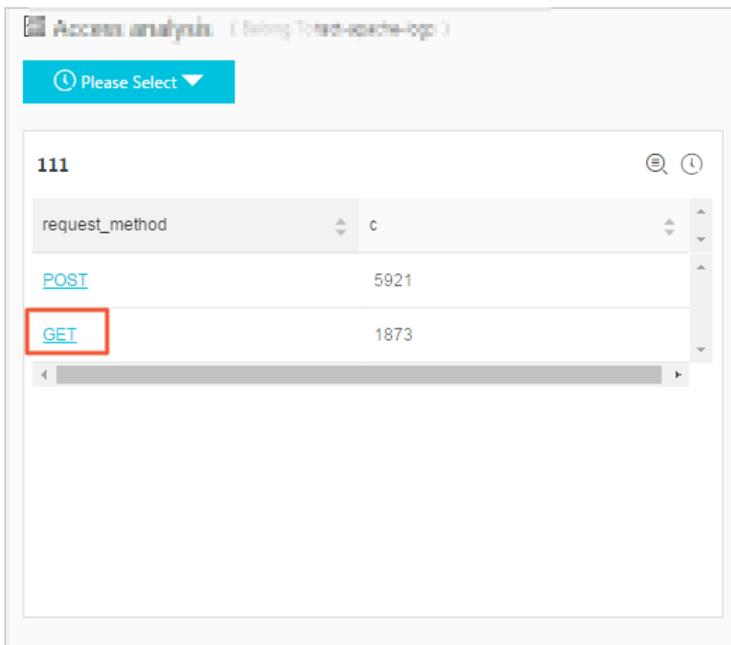
2. Configure drill-down analysis for a chart and add the chart to the dashboard.

- i. On the **Search & Analysis** page, enter a SQL statement to query the number of NGINX access log entries of each request method, and display the result in a table. The query statement is shown as follows:

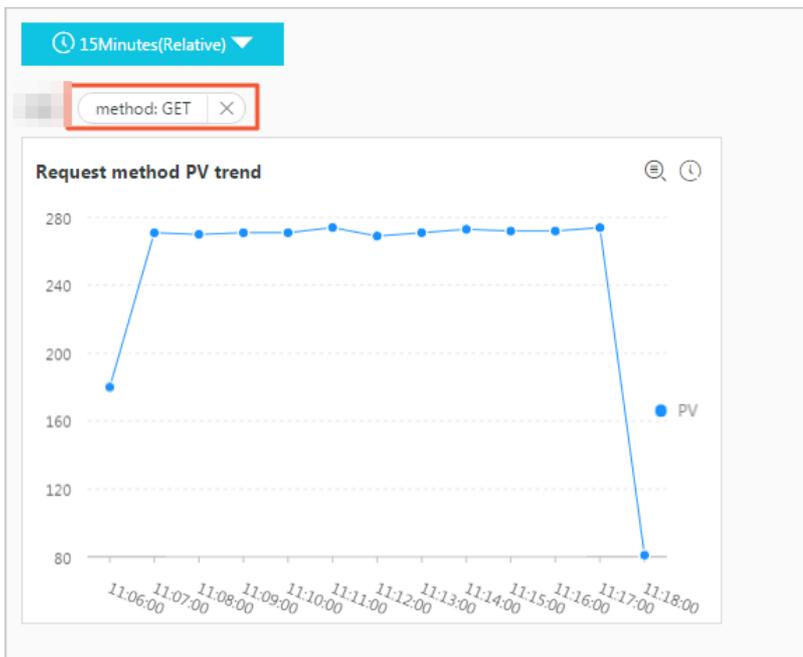
```
*|SELECT request_method, COUNT(1) AS c GROUP BY request_method ORDER BY c DESC LIMIT 10
```

- ii. Configure drill-down analysis for the `request_method` column in the table.

iii. Click the GET request on the RequestMethod dashboard.



iv. Redirected to the destination_drilldown dashboard. You are redirected to the dashboard configured in step . The asterisk (*) in the query statement is replaced with GET . The dashboard shows how the PV of the GET request changes over time.



30.4.11.2.6. Configure and use a filter on a dashboard of a Logstore

This topic describes how to configure and use a filter on a dashboard of a Logstore. Filters help you refine search results or replace placeholder variables with specified values.

Prerequisites

- The index feature of the Logstore is enabled and configured. For more information, see [Enable the index feature and configure indexes for a Logstore](#).
- A dashboard is created. For more information, see [Create and delete a dashboard](#). A placeholder variable is

specified for charts on the dashboard if the filter type is set to Replace Variable.

Context

Each chart on a dashboard is a visualized query statement. When you configure a filter on a dashboard, the specified filtering condition or variables apply to all charts on the dashboard. You can configure the following two types of filters:

- **Filter:** Add key-value pairs as a filtering condition before the query statement [search query]. The new query statement is key:value AND [search query], which means to search the result of the original query statement for log entries that contain key:value. For the Filter type, you can select or enter multiple key-value pairs. When you select multiple key-value pairs as filtering conditions, the logical OR operator is used between the pairs.
- **Replace Variable:** Specify a variable. If the dashboard contains a chart configured with the same placeholder variable, the placeholder variable in the query statement of the chart is replaced with the selected value.

Components

Each filter chart has one or more filters. Each filter consists of the following two components:

- The key, which indicates a filter operation.
- The values of the key.

Procedure

1. [Log on to the Log Service console.](#)
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. In the dashboard list, click the name of the target dashboard.
5. On the dashboard page, click **Edit** to enter the edit mode.
6. Click the  icon, and then set the filter parameters. Click **OK**.

Parameters of a filter

Parameter	Description
Filter Name	The name of the filter.
Display Settings	Valid values: <ul style="list-style-type: none"> ◦ Title: specifies to add a title for a filter. You can turn on the Title switch to add a title for a filter. ◦ Border: specifies to add borders for a filter. You can turn on the Border switch to add borders for a filter. ◦ Background: specifies to add a white background for a filter. You can turn on the Background switch to add a white background for a filter.
Type	The filter type. Valid values: <ul style="list-style-type: none"> ◦ If you select Filter, a List Item is a value of a key that is used to filter the results of a query statement. You can set multiple values for a key. After the filter takes effect, you can select one or more values on the dashboard to filter query results based on your needs. ◦ If you select Replace Variable, a List Item is the value that replaces a specified variable. You can set multiple values for a variable. After the filter takes effect, you can select one or more values on the dashboard to filter query results based on your needs.

Parameter	Description
Key	<ul style="list-style-type: none"> For the Filter type, the Key parameter specifies the key in the filtering condition. For the Replace Variable type, the Key parameter specifies the variable. <p> Note The variable must be the same as the placeholder variable that you specified for charts. Otherwise, the placeholder variable cannot be replaced.</p>
Alias	The alias of the key. This parameter is available only when you select Filter . After you set an alias for a key, the alias is displayed in the filter on the dashboard.
Global filter	This switch indicates whether to filter the specified values in all fields. This switch is turned off by default. The switch is available only when you select the Filter type. <ul style="list-style-type: none"> To filter the specified values in all fields, turn on the Global filter switch. To filter the specified values in specified keys, turn off the Global filter switch.
Static List Items	The values of the key that is used as a filtering condition. Click the plus sign and enter a value for the key in the text box.
Add Dynamic List Item	The dynamic value of the key retrieved by running the specified query statement. Turn on the Add Dynamic List Item switch, select a Logstore, and turn on the Inherit Filtering Conditions switch (specifies whether to include the filtering condition in the query statement). Enter a query statement in the search box, specify a time range, and then click Search to preview the dynamic values.

Examples

You can use a filter to modify the query statements of charts on a dashboard and replace placeholder variables in the charts on the dashboard. Each chart represents a query statement in the format of `[search query] | [sql query]`. The filter query is appended to the original query statement to filter data.

- If the filter type is **Filter**, the key-value pairs to be filtered are added before `[search query]` to form a new query statement by using the logical **AND** operator. The new query statement is `key:value AND [search query]`.
- If the filter type is **Replace Variable**, the filter queries all charts that contain the specified placeholder variables on the dashboard and replaces the placeholder variables with the selected `values`.

Example 1: Use different time granularities to analyze logs

After you collect NGINX access logs, you need to query and analyze these logs in real time.

You can use a query statement to view the number of page views (PVs) per minute. However, if you want to view the number of PVs per second, you must modify the value of `__time__ - __time__ % 60` in the query statement. To simplify this operation, you can use a filter to replace variables in the query statement.

- Use the following query statement to view the number of PVs per minute:

```
* | SELECT date_format(__time__ - __time__ % 60, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```

- Add the chart to a dashboard, and select `60` generate a variable with the name `interval`.

The screenshot shows a dashboard interface with a 'Chart Preview' on the left and a 'Data Source' configuration panel on the right. The chart preview displays a table with columns 'time' and 'count'. The data points are:

time	count
17:04:00	4
17:09:00	4
17:13:00	8

The 'Data Source' panel includes a 'Query' section with a 'Generate Variable' button. Below it, there is a 'Variable Config' section with fields for 'Variable Name' (set to 'interval'), 'Default Value' (set to '60'), and 'Matching Mode' (set to 'Global Match'). A 'Result' section shows the query statement with the variable replaced: `* | SELECT date_format(__time__ - __time__ % $interval, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time`.

3. Add a filter on the dashboard page.

- **Type:** Select **Replace Variable**.
- **Key:** Enter `interval`.
- **Static List Items:** Add `1` and `120` as values of the key. The default unit is seconds.

4. In the Filter section of the dashboard, select `1` from the Interval drop-down list to view statistics by second.

The following statement shows the query statement after the placeholder variable is replaced:

```
* | SELECT date_format(__time__ - __time__ % 1, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```

The screenshot shows a 'Variable' configuration dialog with 'interval: 1' entered. Below the dialog, a table titled 'PV-01 15Minutes(Relative)' displays the following data:

time	count
17:15:00	103
17:16:00	112
17:17:00	68
17:18:00	157

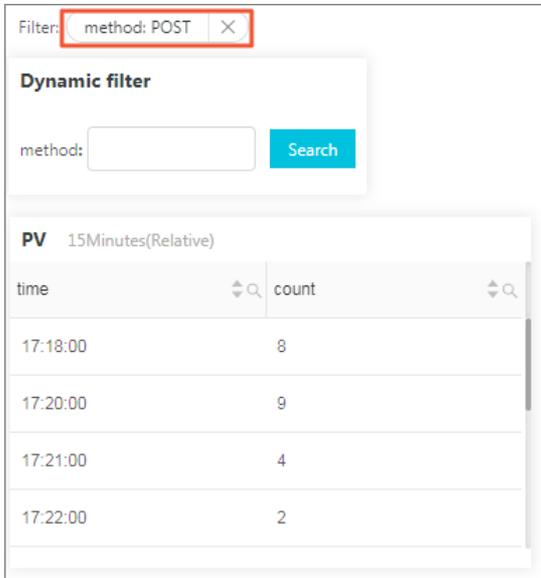
Example 2: Switch between request methods

You can add dynamic values to a filter to dynamically switch between request methods. In example 1, the query statement starts with an asterisk (*), which means no condition is set to filter the query results and all logs are queried. You can add another filter to view the PV data of different request `methods`.

1. Add a filter on the dashboard and turn on the **Add Dynamic List Item** switch. Set the parameters as follows:
 - **Type:** Select **Filter**.
 - **Key:** Enter `method`.
 - **Select Logstore:** Select the Logstore to which the dashboard belongs.
 - **Add Dynamic List Item:** Enter a query statement to obtain the relevant dynamic values, and then click **OK**.

- In the Filter section of the dashboard, select `POST` from the method drop-down list. Only the PV data whose `method` is `POST` is displayed in the chart. The query statement is changed as follows:

```
(* and (method: POST) | SELECT date_format(__time__ - __time__ % 60, '%H:%i:%s') as time, count(1) as count GROUP BY time ORDER BY time
```



30.4.11.2.7. Markdown chart

Log Service allows you to add a Markdown chart to a dashboard. In the Markdown chart, you can insert images, links, videos, and other elements to make your dashboard page more intuitive.

Context

You can create different Markdown charts based on your business needs. Markdown charts can make a dashboard more intuitive. You can insert text such as background information, chart description, notes, extension information, and custom images in a Markdown chart. You can also insert saved searches or dashboard links of other projects to redirect to other query pages.

Scenarios

You can insert links in a Markdown chart to redirect to other dashboard pages of the current project. You can also insert an image corresponding to each link for better recognition. In addition, you can use a Markdown chart to describe parameters of analysis charts.

Procedure

- Log on to the [Log Service console](#).
- Click a project name.
- In the left-side navigation pane, click the **Dashboard** icon.
- In the dashboard list, click the name of the target dashboard.
- On the dashboard page, click **Edit** to enter the edit mode.
- In the edit mode, drag the Markdown icon  from the menu bar to the specified location to create a Markdown chart.
- Click the created Markdown chart, find the More icon in the upper-right corner of the chart, and click **edit**.

Parameter	Description
Chart Name	The name of the Markdown chart.
Show Border	Specifies whether to show the borders of a Markdown chart. You can turn on the Show Border switch to show the borders of a Markdown chart.
Show Title	Specifies whether to show the title of a Markdown chart. You can turn on the Show Title switch to show the title of a Markdown chart.
Show Background	Specifies whether to show the background of a Markdown chart. You can turn on the Show Background switch to show the background of a Markdown chart.
Query Binding	Specifies whether to bind a query statement to a Markdown chart. You can turn on the Query Binding switch and bind a query statement to a Markdown chart. Then, dynamic query results are displayed in the Markdown chart.

8. (Optional)Bind a query statement

- i. Select the target Logstore and enter a query statement in the search box. A query statement consists of a search statement and an analytic statement in the format of `search statement|analytic statement`.
- ii. On the Search & Analysis page, click **15 Minutes(Relative)** to set the time range for the query.
- iii. Click **Search** to display the first values of the returned query result.
- iv. Click the ⊕ icon next to a field to insert the corresponding query result to **Markdown Content**.

9. Edit **Markdown Content**. Enter Markdown content in the **Markdown content** column on the right. Then data preview is displayed in real time in the **Show Chart** column on the right. You can modify the Markdown content based on the data preview on the right.

Modify a Markdown chart

- **Modify the location and size of a Markdown chart**
 - i. Click **Edit** in the upper-right corner of the **Dashboard** page.
 - ii. Drag the Markdown icon to the specified location on the dashboard and drag the lower-right corner of the chart to adjust its size.
 - iii. Click **Save** in the upper-right corner of the dashboard page to save the modification.
- **Modify the title of a Markdown chart**
 - i. Click **Edit** in the upper-right corner of the **Dashboard** page.
 - ii. Click the specified Markdown chart, find the **More** icon in the upper-right corner of the chart, and click **Edit**.
 - iii. Enter a new title in the **Chart name** field and then click **OK**.
 - iv. Click **Save** in the upper-right corner of the dashboard page. On the dialog box that appears, click **OK**.
- **Modify the content of a Markdown chart**
 - i. Click **Edit** in the upper-right corner of the **Dashboard** page.
 - ii. Click the specified Markdown chart, find the **More** icon in the upper-right corner of chart, and click **Edit**.
 - iii. Modify the chart content, and then click **OK**.
 - iv. Click **Save** in the upper-right corner of the dashboard page. On the dialog box that appears, click **OK**.
- **Delete a Markdown chart**
 - i. Click **Edit** in the upper-right corner of the **Dashboard** page.
 - ii. Click the specified Markdown chart, find the **More** icon in the upper-right corner of chart, and click **Delete**.
 - iii. Click **Save** in the upper-right corner of the dashboard page. On the dialog box that appears, click **OK**.

Common Markdown syntax

- Title

Markdown syntax:

```
# Level 1 heading
## Level 2 heading
### Level 3 heading
```

- Link

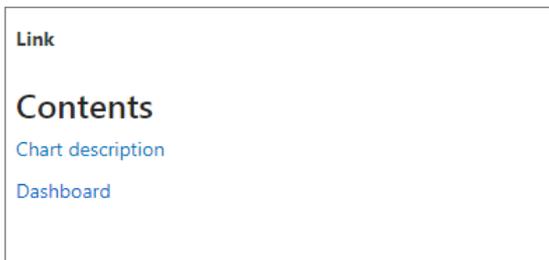
Markdown syntax:

```
### Contents

[Chart description](https:// xxx)

[Dashboard](https:// xxx)
```

Link preview



- Image

Markdown syntax:

```
<div align=center>

! [ Alt txt][id]

With a reference later in the document defining the URL location

[id]: https://octodex.github.com/images/dojocat.jpg "The Dojocat"
```

- Special tag

Markdown syntax:

```

---

__Advertisement :)__

==some mark== `some code`
> Classic markup: :wink: :crush: :cry: :tear: :laughing: :yum:
>> Shortcuts (emoticons): :-) 8-) ;)

__This is bold text__

*This is italic text*

---

```

30.5. Alerts

30.5.1. Overview

Log Service enables you to configure alerts for charts on a dashboard to monitor the service status in real time.

You can configure alerts on the **Search & Analysis** page of a Logstore or on a **Dashboard** page. When you configure an alert, you must configure the alert name, trigger condition, notification method, and other parameters. After you **Configure alerts**, Log Service checks the query results on the dashboard at an interval and sends an alert notification if the check results meet the specified conditions. In this way, Log Service facilitates real-time monitoring of the service status.

Limits

Item	Description
Associated charts	The number of charts that can be associated with an alert ranges from 1 to 3.
String	If the length of a string exceeds 1,024 characters, only the first 1,024 characters are computed during a query.
Conditional expression	<ul style="list-style-type: none"> The conditional expression must be 1 to 128 characters in length. The conditional expression is evaluated based on the first 100 log entries returned for a query. The conditional expression can be evaluated up to 1,000 times. If the conditional expression is not matched, the alert is not triggered.
Search Period	The time range of each query statement cannot exceed 24 hours.

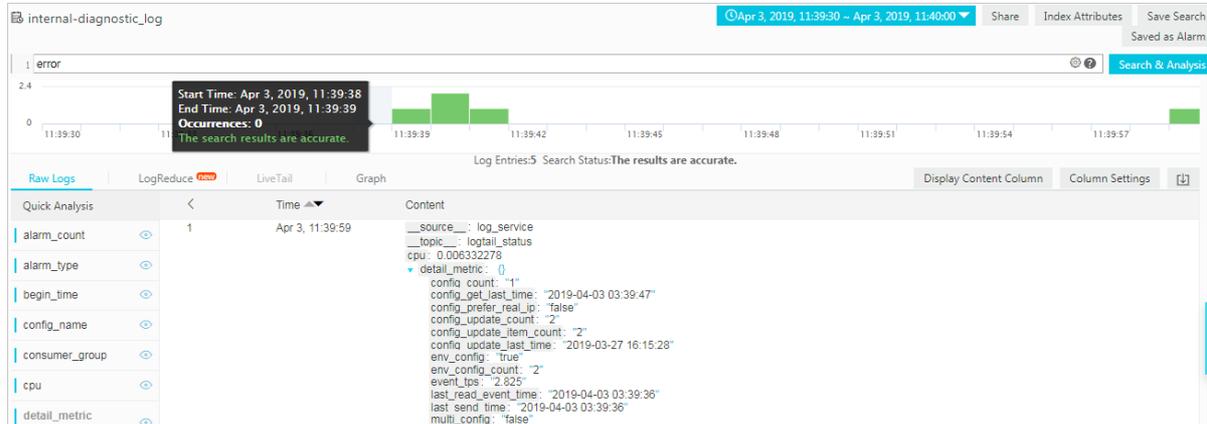
Query statement in an alert

An alert is configured based on analysis charts on a dashboard. An analysis chart is a visualized query result of a query statement. A query statement can include a search statement and an analytic statement.

- If you use only a search statement for a query, log data that matches the search condition is returned.
- If you include search and analytic statements for a query, log data that matches the search condition is analyzed before being returned.
- Search statement

For example, you want to query the data that contains "error" information in the last 15 minutes. The search statement is error. A total of 154 log entries are retrieved. Each log entry consists of key-value pairs. You can set an alert rule for the value of a key.

Note If over 100 log entries are returned for a query, only the first 100 log entries are used for evaluating the conditions set in an alert. An alert is triggered when any of the first 100 log entries returned meets the conditions.

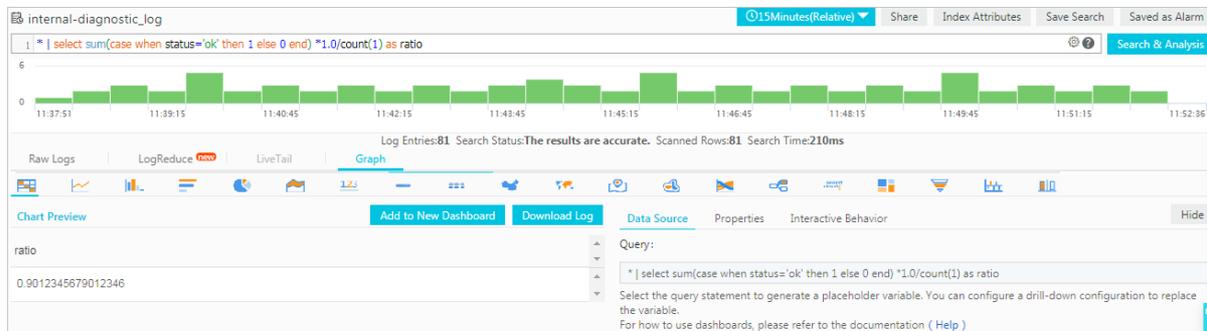


• Search and analytic statement

For example, the following statement queries the ratio of log entries whose status field value is ok to all log entries. For more information about query syntax, see [Query syntax](#).

```
* | select sum(case when status='ok' then 1 else 0 end) *1.0/count(1) as ratio
```

If you set the trigger condition of an alert to `ratio < 0.9`, the alert is triggered when the ratio of log entries whose status field value is ok to all log entries is less than 90%.



30.5.2. Configure an alarm

30.5.2.1. Configure alerts

Log Service allows you to configure alerts on the Search & Analysis page of a Logstore or on a dashboard page. If the trigger condition of an alert is met, the alert is triggered and a notification is sent to specified recipients.

Prerequisites

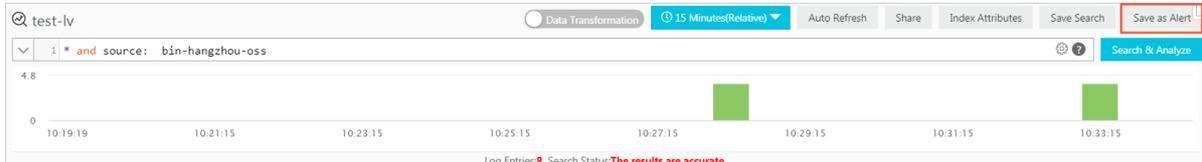
- Log data is collected.
- The indexing feature is enabled and indexes are configured. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Context

Alerts are configured based on analysis charts. When you view an analysis chart, you can add the chart to a dashboard and configure an alert for the chart. You can also configure an alert for the existing charts on a dashboard.

- Create a chart and configure an alert for the chart

You can save the results of a query statement as a chart on a dashboard, and configure an alert for the chart. When you configure an alert on the Search & Analysis page, you must specify the name of the dashboard on which the chart is saved and the chart name.



- Configure an alert for existing charts on a dashboard.

You can configure an alert for one or more charts on a dashboard at a time. When you configure an alert for multiple charts, you can specify a conditional expression for each chart and combine the conditional expressions into the trigger condition for the alert.

This topic describes how to configure an alert for existing charts on a dashboard.

Note If an alert is configured for a chart on a dashboard and you update the search and analytic statement of the chart, you must update the search and analytic statement in the alert configuration. For more information, see [Modify an alert](#).

For information about example alert configurations, see [FAQ about alerts](#).

Procedure

1. [Log on to the Log Service console](#).
2. In the **Projects** section, click the name of a project.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. Click the target dashboard name.
5. In the upper-right corner of the dashboard, choose **Alerts > Create**.
6. In the **Create Alert** wizard, configure an alert and click **Next**.The following table describes the configuration parameters of an alert.

Parameter	Description
Alert Name	The name of the alert. The name must be 1 to 64 characters in length.
Associated Chart	<p>The chart that is associated with the alert.</p> <p>The Search Period parameter specifies the time range of log data that Log Service reads when you query data. You can select a relative time or a time frame. For example, if you set Search Period to 15 minutes (relative) and start the query at 14:30:06, Log Service reads the log data that was written from 14:15:06 to 14:30:06. If you set Search Period to 15 minutes (time frame) and start the query at 14:30:06, Log Service reads the log data that was written from 14:15:00 to 14:30:00.</p> <p>To associate the alert with multiple charts, you must separately add and configure the charts. The number before the chart name indicates the sequence number of the chart in the alert configuration. You can use the sequence number to associate a chart in the trigger condition.</p>
Frequency	The time interval at which Log Service executes the alert.

Parameter	Description
Trigger Condition	<p>The conditional expression that determines whether to trigger the alert. If the condition is met, Log Service sends an alert notification based on the specified Check Frequency and Notification Interval.</p> <p>For example, you can enter <code>pv%100 > 0 && uv > 0</code> in the Trigger Condition field.</p> <p>Note In the conditional expression, you can use <code>\$sequence number</code> to differentiate charts. For example, you can use <code>\$0</code> to identify the chart whose sequence number is 0.</p>
Advanced	
Notification Trigger Threshold	<p>If the number of times that the trigger condition is met exceeds this threshold and the specified Notification Interval elapses, Log Service sends an alert notification to the specified recipients.</p> <p>The default value of Notification Trigger Threshold is 1. This value indicates that each time the specified Trigger Condition is met, Log Service checks Notification Interval to determine whether to send notifications.</p> <p>You can set a custom value. This way, Log Service sends an alert notification to the specified recipient only after the trigger condition is met multiple times. For example, if you set the value to 100, Log Service checks Notification Interval only after the trigger condition is met 100 times. If the Notification Trigger Threshold and Notification Interval are reached, Log Service sends an alert notifications to the specified recipients. The overall count is then reset to zero. If Log Service fails to check log data due to exceptions such as network failures, the overall count does not change.</p>
Notification Interval	<p>The time interval at which Log Service sends an alert notification.</p> <p>If the number of times that the trigger condition is met exceeds the specified Notification Trigger Threshold and the specified notification interval elapses, Log Service sends an alert notification to the specified recipients. If you set this parameter to 5 minutes, you can receive up to one alert notification every 5 minutes. The default value is No Interval.</p> <p>Note You can use the Notification Trigger Threshold and Notification Interval parameters to control the number of alert notifications that you receive.</p>

7. Set the notification method. Notifications can be sent to a custom webhook address in a specified format. To use this notification method, you must set the Request URL, Request Method, and Request Content parameters. For more information, see [Notification methods](#).

- Request URL: a custom webhook address, for example, `https://webhook.com/notify`.
- Request Method: the request method. Request methods include GET, PUT, POST, DELETE, and OPTIONS.
- Request Content: the content of the notification. The content must be 1 to 500 characters in length. Template variables are supported.

8. Click OK.

30.5.2.2. Grant permissions on alerts to a RAM user

This topic describes how to grant a RAM user the permissions to enable the alerting feature.

Context

Grant a RAM user the permissions only to create and modify alerts. Create a custom authorization policy, and apply the policy to the RAM user. For more information, see Procedure in this topic.

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. [Create a RAM role](#).
3. [Create a permission policy](#). Use the following policy and replace the `<Project name>` .

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateLogStore",
        "log:CreateIndex",
        "log:UpdateIndex"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/logstore/internal-alert-history"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateDashboard",
        "log:CreateChart",
        "log:UpdateDashboard"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/dashboard/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:*"
      ],
      "Resource": "acs:log:*:*:project/<Project name>/job/*"
    }
  ]
}
```

4. [Create a RAM user](#).
5. [Create a RAM user group](#).
6. [Add a RAM user to a RAM user group](#)
7. [Grant permissions to a RAM role](#).

30.5.2.3. Notification methods

This topic describes the available notification methods that you can configure for an alert in Log Service.

WebHook-Custom

You can set the notification method to WebHook-Custom. When an alert is triggered, Log Service sends an alert notification to a specified webhook URL by using a specified method.

Note The timeout period of the WebHook-Custom notification method is five seconds. If no response is received within five seconds after a notification request is sent, the notification request is deemed to be failed.

1. Configure an alert in the Log Service console. For more information, see [Configure alerts](#). Select **WebHook-Custom** from the Notifications drop-down list.
2. Enter your custom webhook URL in the **Request URL** field. Select a **Request Method**.
3. (Optional) Click **Add Request Headers** to add request header fields. By default, the request header contains the field `Content-Type: application/json;charset=utf-8`. You can add request header fields based on your business needs.
4. Enter the notification content in the **Request Content** field. When an alert is triggered, Log Service sends the specified notification content to the custom webhook URL by using the specified method.
5. Click **Submit**.

Template variables

You must set **Content** for each notification method. In the notification content, you can reference some template variables in the `${fieldName}` format for the alert. When sending an alert notification, Log Service replaces the template variables referenced in the **Content** field with real values. For example, it replaces `${Project}` with the name of the project to which the alert belongs.

Note You must reference valid variables. If a referenced variable does not exist or is invalid, Log Service processes the variable as a null string. If the value of a referenced variable is of the object type, the value is converted and displayed as a JSON string.

The following table lists all the supported variables and reference methods.

Variable	Description	Example	Reference example
Aliuid	The ID of the Apsara Stack tenant account to which the project belongs.	1234567890	The alert configured by the user <code>\${Aliuid}</code> is triggered.
Project	The project to which the alert belongs.	my-project	The alert configured in the project <code>\${Project}</code> is triggered.
AlertID	The unique ID of the alert.	0fdd88063a611aa114938f9371dae6b6-1671a52eb23	The ID of the alert is <code>\${AlertID}</code> .
AlertName	The name of the alert, which must be unique in a project.	alert-1542111415-153472	The alert <code>\${AlertName}</code> is triggered.
AlertDisplayName	The display name of the alert.	My alert	The alert <code>\${AlertDisplayName}</code> is triggered.
Condition	The conditional expression that triggers the alert. Each variable in the conditional expression is replaced with the value that triggers the alert. The value is enclosed in brackets [].	[5] > 1	The conditional expression that triggers the alert is <code>\${Condition}</code> .

Variable	Description	Example	Reference example
RawCondition	The original conditional expression that triggers the alert. Variables in the conditional expression are not replaced.	count > 1	The original conditional expression that triggers the alert is <code>\${RawCondition}</code> .
Dashboard	The name of the dashboard with which the alert is associated.	mydashboard	The alert is associated with the dashboard <code>\${Dashboard}</code> .
DashboardUrl	The URL of the dashboard with which the alert is associated.	https://sls.console.aliyun.com/next/project/myproject/dashboard/mydashboard	The URL of the dashboard associated with the alert is <code>\${DashboardUrl}</code> .
FireTime	The time when the alert is triggered.	2018-01-02 15:04:05	The alert is triggered at <code>\${FireTime}</code> .
FullResultUrl	The URL used to query the history records that an alert rule was executed.	https://sls.console.aliyun.com/next/project/my-project/logsearch/internal-alert-history?endTime=1544083998&queryString=AlertID%3A9155ea1ec10167985519fccede4d5fc7-1678293caad&queryTimeType=99&startTime=1544083968	Click <code>\${FullResultUrl}</code> to view details.

Variable	Description	Example	Reference example
Results	<p>The parameters and results of each log data query. The value is of the array type. For information about parameters in the Results field, see Fields in alert log entries.</p> <p>Note A maximum of 100 alert notifications can be sent.</p>	<pre>[{ "EndTime": 1542507580, "FireResult": { "__time__": "1542453580", "count": "0" }, "LogStore": "test-logstore", "Query": "* SELECT COUNT(*) as count", "RawResultCount": 1, "RawResults": [{ "__time__": "1542453580", "count": "0" }], "StartTime": 1542453580 }]</pre>	<p>The first query starts at <code>#{Results[0].StartTime}</code> and ends at <code>#{Results[0].EndTime}</code>. The alert has been triggered <code>#{Results[0].FireResult.count}</code> times.</p> <p>Note In this example, the number 0 indicates the sequence number of the chart or the search and analytic statement. For more information, see How can I check the sequence number of a chart?</p>

30.5.3. Modify and view an alarm

30.5.3.1. Modify an alert

This topic describes how to modify an alert. After you create an alert, you can modify the alert and then update the alert. To modify an alert associated with a search statement, you can directly modify the search statement.

Precautions

- You can modify only search statements with which alerts are associated. You cannot modify search statements to search and analytic statements, which are in the format of `search statement|analytic statement`.
For example, after you associate the `request_method: GET` statement with an alert, you can modify the statement to `error`, but you cannot modify the statement to `error| select count(1) as c`.
- To modify an alert, you can click **Modify Settings** on the **Alert Overview** page, or choose **Alerts > Modify** on the associated dashboard.

Modify the search statement associated with an alert

If you associate a search statement with an alert, you can modify the search statement to modify the alert.

1. [Log on to the Log Service console.](#)
2. Click a project name.
3. In the left-side navigation pane, click the **Dashboard** icon.
4. In the dashboard list, click the name of the target dashboard.
5. On the dashboard, choose **Alerts > Modify**.
6. Find the search statement, and then click



. You can modify only search statements with which alerts are associated. You cannot modify search statements to search and analytic statements, which are in the format of `search statement|analytic statement`.

7. On the dialog box that appears, enter a new search statement, click **Preview**, and then click **OK** after the search statement is verified.
8. Modify other parameters specific to your environment, such as **Frequency** and **Trigger Condition**, and then click **Next**.
9. Set the notification method, and then click **Submit**.

Modify the chart associated with an alert

After you create an alert, you can modify the chart associated with the alert to modify the alert.

1. In the dashboard list, click the name of the target dashboard.
2. On the dashboard, choose **Alerts > Modify**.
3. Find the **Associated Chart**, and then click



next to **Query**.

4. On the dialog box that appears, enter a new query statement, click **Preview**, and then click **OK** after the query statement is verified.
5. Modify other parameters specific to your environment, such as **Frequency** and **Trigger Condition**, and then click **Next**.
6. Set the notification method.
7. Click **Submit**. The new settings take effect immediately.

30.5.3.2. View history alerts

This topic describes how to view history alerts in the Log Service console. Log Service records alerts as log data and creates a dashboard to display alert details.

View history alerts in the Logstore

When you create an alert, Log Service creates a Logstore named `internal-alert-history` for the project to which the alert belongs. A log entry is generated and written to the Logstore each time the alert rule is executed, regardless of whether the alert is triggered. For more information about the fields in the log entry, see [Fields in alert log entries](#).

Note The Logstore does not incur fees and cannot be deleted or modified. Each alert log entry is retained in the Logstore for seven days.

1. [Log on to the Log Service console](#).
2. Click a project name.
3. Click the  icon next to the `internal-alert-history` Logstore, and then select **Search & Analysis**.
4. On the page that appears, query alert log entries based on your needs.

View history alerts on the dashboard

After you create an alert, Log Service creates a dashboard named `internal-alert-analysis` for the project to which the alert belongs. The dashboard displays the statistics of all previous alerts, including the number of triggered alerts, percentage of successful alerts and notifications, and top 10 alerts whose alert rules are executed.

Note The dashboard cannot be deleted or modified.

1. In the left-side navigation pane, click the **Dashboard** icon.
2. Click **Alert History Statistics** to open the dashboard page. On the **Alert History Statistics** dashboard, the details of history alerts are displayed, including whether the alerts are triggered, why the alerts are triggered, error information, and other information.

30.5.3.3. Manage an alert

This topic describes how to manage an alert after you create the alert.

Context

You can disable, enable, modify, and delete the alert, or view the details of the alert such as the update time.

View the details of an alert

1. [Log on to the Log Service console](#).
2. Click a project name.
3. In the left-side navigation pane, click the **Alerts** icon.
4. In the alert list, click the name of the target alert. On the **Alert Overview** page, you can view the details of the alert, such as the dashboard, creation time, last update time, check frequency, alert status, and notification status.

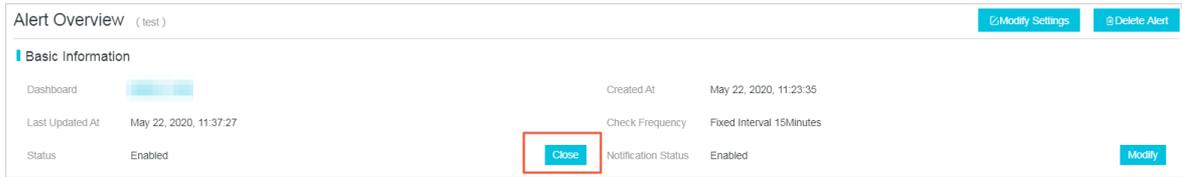
Basic Information			
Dashboard		Created At	Jan 19, 2020, 14:57:16
Last Updated At	Jun 23, 2020, 12:49:45	Check Frequency	Cron Expression:0/5 * * * *
Status	Enabled	Notification Status	Enabled

Disable and enable an alert

After you create an alert, you can disable or enable the alert. If you disable an alert, Log Service no longer checks the alert or send alert notifications.

1. In the left-side navigation pane, click the **Alerts** icon.
2. In the alert list, click the name of the target alert. On the **Alert Overview** page, click **Enable** or **Close** in the

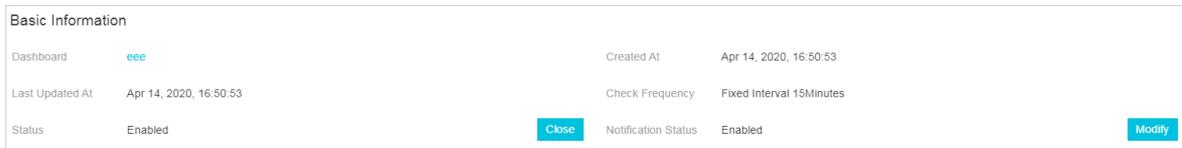
Alert Status field.



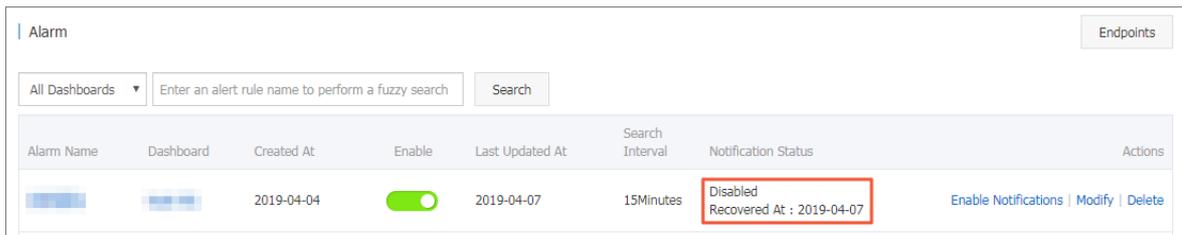
Disable and enable alert notifications

You can disable notifications of enabled alerts. After you disable notifications of an alert, alert notifications are not sent even if the trigger condition is met.

1. In the left-side navigation pane, click the Alerts icon.
2. In the alert list, click the name of the target alert. On the Alert Overview page, click Modify in the Notification Status field.



3. Set the time range for which notifications are disabled, and then click Confirm. After you disable alert notifications, you can view the time when alert notifications resume in the Notification Status field. You can click Modify in the Notification Status field to manually resume alert notifications.



Delete an alert

You cannot recover a deleted alert. Proceed with caution when you delete an alert.

1. In the left-side navigation pane, click the Alerts icon.
2. In the alert list, click the name of the target alert.
3. On the Alert Overview page, click Delete Alert.
4. In the dialog box that appears, click OK.

30.5.4. Relevant syntax and fields for reference

30.5.4.1. Conditional expression syntax of an alert

This topic describes how to configure a conditional expression for an alert in Log Service. An alert is triggered if the conditional expression configured for the alert is met.

In determining whether the conditional expression of an alert is met, the results of query statements configured for the alert are used as the inputs and the log fields in the conditional expression are used as the variables. If the conditional expression is met, the alert is triggered.

Limits

- Negative numbers must be enclosed with parentheses (), for example, $x + (-100) < 100$.
- Numeric data is treated as 64-bit floating-point numbers. If a comparison is performed, errors may occur.
- A variable can contain only letters and digits and must start with a letter.

- A conditional expression can be up to 128 characters in length.
- A conditional expression can be evaluated up to 1,000 times. If an alert is configured for multiple charts and the conditional expression of the alert is not met after 1,000 times of evaluation, the alert is not triggered.
- A maximum of three charts can be associated with an alert.
- An alert is triggered if and only if the Boolean value of its conditional expression is true. For example, the result of the expression `100+100 is 200`, which cannot trigger the alert.
- `true` , `false` , `$` , and `.` are reserved and cannot be used as variables.

Basic syntax

The following table lists the syntax supported in a conditional expression.

Syntax	Description	Examples
Basic operators	Supports the addition operator (+), subtraction operator (-), multiplication operator (*), division operator (/), and modulus operator (%).	<code>x*100+y>200</code> <code>x%10>5</code>
Comparison operators	Supports eight comparison operators, including the greater than operator (>), greater than or equal to operator (>=), less than operator (<), less than or equal to operator (<=), equal to operator (==), not equal to operator (!=), match operator (=~), and mismatch operator (!~). <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #d9e1f2;"> <p> Note</p> <ul style="list-style-type: none"> • Backslashes (/) must be escaped. • Regular expressions support the RE2 syntax. </div>	<code>x >= 0</code> <code>x < 100</code> <code>x <= 100</code> <code>x == 100</code> <code>x == "foo"</code> Regular expression: <code>x =~ "\\w +"</code>
Logical operators	Supports the logical operators AND (&&) and OR ().	<code>x >=0&& y <=100</code> <code>x > 0 y > 0</code>
Logical negation	Supports the logical negation operator (!).	<code>!(a < 1 && a > 100)</code>
Numeric constants	Numeric constants are processed as 64-bit floating-point numbers.	<code>x > 100</code>
String constants	String constants are formatted in a string enclosed in single quotation marks (').	<code>foo == 'string'</code>
Boolean constants	Supports true and false.	<code>(x > 100) == true</code>
Parentheses	Parentheses () can be used to enforce precedence order.	<code>x*(y+100)>100</code>
contains function	Determines whether a substring is included in a string. For example, if the result of the <code>contains(field, 'xxxx')</code> function is true, the field string includes the <code>xxxx</code> substring.	<code>contains(foo, 'hello')</code>

Conditional expression for results of multiple query statements

- Syntax

If you configure an alert for multiple charts, the variables in a conditional expression must be prefixed. In this way, you can specify from which query result to obtain the corresponding values of the variables when it evaluates your expression. The format is `$N.fieldname`, where N is the sequence number of the query statement. You can configure up to three query statements in an alert. Therefore the value range of N is 0 to 2. For example, `$0.foo` indicates to obtain the value of the foo field returned from the first query statement. If you configure one query statement in an alert, you do not need to specify the prefix.

 **Note** How can I view the sequence number of a query statement?

In the Alert Configuration step, the Associated Chart parameter specifies the sequence number of each query statement (chart). The first query statement is numbered 0, the second query statement is numbered 1, and the third statement is numbered 2.

• Evaluate a conditional expression

If multiple query results are returned, the variables specified in the conditional expression determine how to use the results to evaluate the conditional expression. For example, if you configure three query statements in an alert, x, y, and z log entries are returned when you execute each of the three statements. The conditional expression that you configure for the alert is `$0.foo > 100 && $1.bar < 100`. Then the first two query results are used to evaluate the conditional expression. A maximum of $x \times y$ times of evaluation (or 1,000 if $x \times y$ is greater than 1,000) is performed. If the conditional expression is met within the maximum times of evaluation, true is returned. Otherwise, false is returned.

Operations

 **Note**

- 64-bit floating-point numbers are used in a conditional expression.
- Each string constant must be enclosed in single quotation marks (') or double quotation marks (""), for example, 'string', and "string".
- Boolean values include true and false.

Operator	Operation		
	Operation between variables	Operation between non-string constants and variables	Operation between string constants and variables
Basic operators, including the addition operator (+), subtraction operator (-), multiplication operator (*), division operator (/), and modulus operator (%)	The left and right operands are converted to numbers before being operated.		Unsupported.

Operator	Operation		
	Operation between variables	Operation between non-string constants and variables	Operation between string constants and variables
Comparison operators, including the greater than operator (>), greater than or equal to operator (>=), less than operator (<), less than or equal to operator (<=), equal to operator (==), not equal to operator (!=)	<p>Operations are performed based on the following priorities:</p> <ol style="list-style-type: none"> 1. The left and right operands are converted to numbers before being operated based on the numerical order. If the conversion fails, 2. operands are operated based on the alphabetical order of strings. 	The left and right operands are converted to numbers before being operated based on the numerical order.	The left and right operands are operated based on the alphabetical order of strings.
Regular expression match operator (=~) and regular expression mismatch operator (!~)	The left and right operands are operated based on the alphabetical order of strings.	Unsupported.	The left and right operands are operated based on the alphabetical order of strings.
Logical operators, including AND (&&) and OR ()	These two operators cannot be directly used on the fields in query results. The left and right operands must be sub-expressions, and the values of the sub-expressions are of the Boolean type.		
Logical negation (!)	This operator cannot be directly used on the fields in query results. The left and right operands must be sub-expressions, and the values of the sub-expressions are of the Boolean type.		
contains function	The left and right operands are operated based on the alphabetical order of strings.	Unsupported.	The left and right operands are operated based on the alphabetical order of strings.
Parentheses ()	Parentheses () enforce precedence order.		

30.5.4.2. Fields in alert log entries

After you configure an alert, Log Service automatically creates a Logstore to store log entries that are generated when alert rules are executed and notifications are sent. This topic describes fields in alert log entries.

Fields

Field	Description	Example
AlertDisplayName	The display name of an alert.	Test alert rules
AlertID	The unique ID of an alert.	0fdd88063a611aa114938f9371daeeb6-1671a52eb23
AlertName	The unique name of an alert in a project.	alert-1542111415-153472
Condition	The conditional expression configured for an alert.	\$0.count > 1
Dashboard	The dashboard associated with an alert.	my-dashboard
FireCount	The cumulative times that an alert has been triggered since the last notification was sent.	1
Fired	Indicates whether an alert was triggered. Valid values: true and false.	true
LastNotifiedAt	The time when the last notification was sent. The time is displayed in a Unix timestamp.	1542164541
NotifyStatus	The status of a notification. Valid values: <ul style="list-style-type: none"> Success: indicates that a notification was successfully sent. Failed: indicates that a notification failed to be sent. NotNotified: indicates that no notification was sent. PartialSuccess: indicates that the notification sending partially succeeded. 	Success
Reason	The reason that a notification failed to be sent or no notification was sent.	result type is not bool

Field	Description	Example
Results	The parameters and results of each log data query. The value is of the array type. For information about parameters in the Results field, see Parameters in the Result field .	<pre>[{ "EndTime": 1542334900, "FireResult": null, "LogStore": "test-logstore", "Query": "* select count(1) as count", "RawResultCount": 1, "RawResults": [{ "__time__": "1542334840", "count": "0" }], "StartTime": 1542334840 }]</pre>
Status	The execution result of an alert. Valid values: Success and Failed.	Success

Parameters in the Result field

Parameter	Description	Example
Query	The query statement that is configured in an alert.	* select count(1) as count
LogStore	The target Logstore of a query.	my-logstore
StartTime	The time when a query starts.	2019-01-02 15:04:05
StartTimeTs	The time when a query starts. The time is in the Unix timestamp format.	1542334840
EndTime	The time when a query ends.	2019-01-02 15:19:05
EndTimeTs	The time when a query ends. The time is in the Unix timestamp format. The actual query time range is [StartTime, EndTime) .	1542334900

Parameter	Description	Example
RawResults	The raw query result. The parameter value is formatted in an array where each element is a log entry. The maximum length of the array is 100.	<pre>[{ "__time__": "1542334840", "count": "0" }]</pre>
RawResultsAsKv	The query result that is formatted in key-value pairs. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> ? Note This parameter can only be used as a template variable. It is not saved to a Logstore. </div>	[foo:0]
RawResultCount	The number of log entries that are returned in the RawResults parameter.	1
FireResult	The log entry that records the triggering of an alert. If an alert is not triggered, the parameter value is null.	<pre>{ "__time__": "1542334840", "count": "0" }</pre>
FireResultAsKv	The log entry that records the triggering of an alert, formatted in key-value pairs. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> ? Note This parameter can only be used as a template variable. It is not saved to a Logstore. </div>	[foo:0]

30.6. Real-time consumption

30.6.1. Overview

Log Service allows you to consume log data by using multiple methods.

After data is collected to LogHub, you can consume the log data by using two methods. The following table describes the methods.

Method	Scenarios	Timeliness	Retention period
Real-time consumption (LogHub)	Real-time computing	Real time	Custom

Method	Scenarios	Timeliness	Retention period
Indexing and query (LogSearch)	Online query of hot data	Real time	Custom

Real-time consumption

LogHub allows you to pull log data and consume the data in real time. The following procedure describes how log data in a shard is consumed:

1. Obtain a cursor based on the start time and end time of data consumption.
 2. Read log data based on the cursor and step parameters and return the position of the next cursor.
 3. Move the cursor to continuously consume log data.
- Consume log data by using an SDK
You can use Log Service SDKs in multiple programming languages such as Java, Python, and Go to consume log data.
 - Consume log data by using consumer groups
Log Service provides an advanced method that allows you to consume logs by using consumer groups. A consumer group is a lightweight computing framework that allows multiple consumers to consume data from a Logstore at the same time. The consumers in a consumer group are automatically allocated shards. Data is consumed in order based on the time when it is written to the Logstore. In addition, the consumers can use checkpoints to resume consumption from a breakpoint. You can use consumer group SDKs in multiple programming languages such as Go, Python, and Java to consume log data.
 - Log consumption by using real-time stream processing systems
 - Use [Spark Streaming clients](#) to consume log data.
 - Use [Storm spouts](#) to consume log data.
 - Use [Flink Connector](#) to consume log data.
 - Log consumption by using open-source services
Use [Flume](#) to consume log data and import log data to Hadoop file system (HDFS) instances.

Log search and analytics

- You can query log data in the Log Service console.
- You can also query log data by using an SDK or the API of Log Service. Log Service provides an HTTP-based RESTful API. You can call all log query API operations that are provided by Log Service.

30.6.2. Consume log data

Log Service provides SDKs in various programming languages, such as Java, Python, and Go. You can use an SDK to consume log data.

Use an SDK to consume log data

The following example shows how to use the SDK for Java to consume log data in a shard:

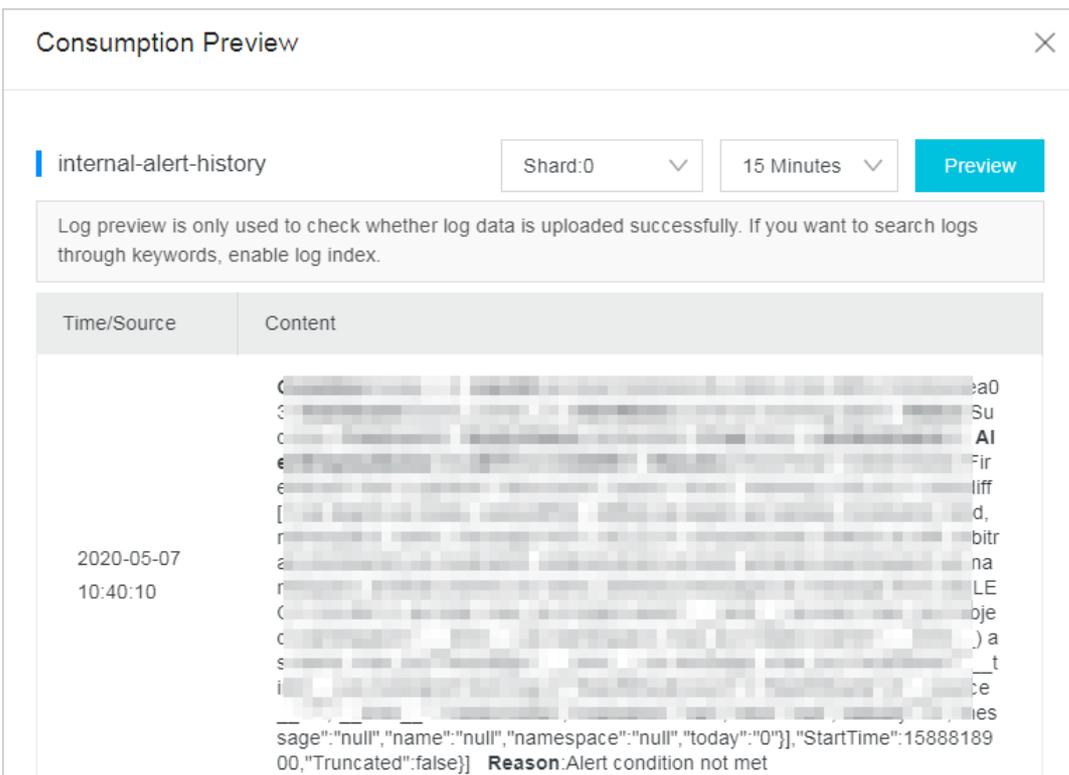
```
Client client = new Client(host, accessId, accessKey);

String cursor = client.GetCursor(project, logStore, shardId, CursorMode.END).GetCursor();
System.out.println("cursor = " + cursor);
try {
    while (true) {
        PullLogsRequest request = new PullLogsRequest(project, logStore, shardId, 1000, cursor);
        PullLogsResponse response = client.pullLogs(request);
        System.out.println(response.getCount());
        System.out.println("cursor = " + cursor + " next_cursor = " + response.getNextCursor());
        if (cursor.equals(response.getNextCursor())) {
            break;
        }
        cursor = response.getNextCursor();
        Thread.sleep(200);
    }
}
catch(LogException e) {
    System.out.println(e.GetRequestId() + e.GetErrorMessage());
}
```

Preview log data in the Log Service console

Log preview is a way of log data consumption. To preview log data that is stored in a Logstore in the Log Service console, perform the following steps:

1. [Log on to the Log Service console](#).
2. In the Projects section, click the target project.
3. In the Logstore list, find the target Logstore, click the  icon next to the Logstore, and then select **Consumption Preview**.
4. In the Consumption Preview dialog box, select a shard, set a time range, and then click **Preview**. The log preview page displays the log data of the first 10 packets in the specified time range.



30.6.3. Consumption by consumer groups

30.6.3.1. Use consumer groups to consume log data

Log consumption by consumer groups

Consumer groups allow you to focus on the business logic during log data consumption. You do not need to consider factors such as Log Service implementation, load balancing among consumers, and failovers that may be introduced when you use SDKs to consume log data.

Terms

The following table describes the terms of consumer groups and consumers.

Term	Description
consumer group	A consumer group consists of multiple consumers. Each consumer in a consumer group consumes different data in a Logstore.
consumer	In a consumer group, a consumer consumes data. Each consumer name in a consumer group must be unique.

A Logstore has multiple shards. A consumer library allocates shards to consumers in a consumer group based on the following principles:

- Each shard can be allocated to one consumer.
- One consumer can consume data in multiple shards.

After a new consumer joins a consumer group, shards allocated to the consumer group are reallocated to each consumer for load balancing. The reallocation is based on the preceding principles and cannot be viewed by users.

A consumer library can also store checkpoints. This allows consumers to resume consumption from a breakpoint and avoid repetitive consumption after a program fault is resolved.

Procedure

Log consumption by consumer groups is implemented in Java or Python. The following section takes Java as an example to describe how consumer groups consume log data.

1. Add Maven dependencies.

```
<dependency>
  <groupId>com.google.protobuf</groupId>
  <artifactId>protobuf-java</artifactId>
  <version>2.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-client-lib</artifactId>
  <version>0.6.16</version>
</dependency>
```

2. Create a main.java file.

```

import com.aliyun.openservices.loghub.client.ClientWorker;
import com.aliyun.openservices.loghub.client.config.LogHubConfig;
import com.aliyun.openservices.loghub.client.exceptions.LogHubClientWorkerException;

public class Main {
    // Specify the endpoint of Log Service.
    private static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
    // Specify a Log Service project name.
    private static String sProject = "ali-cn-hangzhou-sls-admin";
    // Specify a Logstore name of Log Service.
    private static String sLogstore = "sls_operation_log";
    // Specify a consumer group name.
    private static String sConsumerGroup = "consumerGroupX";
    // Specify an AccessKey pair for data consumption.
    private static String sAccessKeyId = "";
    private static String sAccessKey = "";

    public static void main(String[] args) throws LogHubClientWorkerException, InterruptedException {
        // The second parameter is the consumer name. Each consumer name in a consumer group must be unique. However, the names of consumer groups can be the same. When different consumers start multiple processes on multiple servers to consume the data of a Logstore, you can use a server IP address to identify a consumer. The ninth parameter maxFetchLogGroupSize indicates the maximum number of log groups retrieved from the server at a time. Valid values: 1 to 1000. You can use the default value or specify a value based on your needs.
        LogHubConfig config = new LogHubConfig(sConsumerGroup, "consumer_1", sEndpoint, sProject, sLogstore, sAccessKeyId, sAccessKey, LogHubConfig.ConsumePosition.BEGIN_CURSOR);
        ClientWorker worker = new ClientWorker(new SampleLogHubProcessorFactory(), config);
        Thread thread = new Thread(worker);
        // The ClientWorker instance runs automatically after the thread is executed and extends the Runnable interface.
        thread.start();
        Thread.sleep(60 * 60 * 1000);
        // The shutdown function of the ClientWorker instance is called to exit the consumption instance. The associated thread is stopped automatically.
        worker.shutdown();
        // Multiple asynchronous tasks are generated when the ClientWorker instance is running. We recommend that you wait 30 seconds to ensure that all running tasks exit after shutdown.
        Thread.sleep(30 * 1000);
    }
}

```

3. Create a SampleLogHubProcessor.java file.

```

import com.aliyun.openservices.log.common.FastLog;
import com.aliyun.openservices.log.common.FastLogContent;
import com.aliyun.openservices.log.common.FastLogGroup;
import com.aliyun.openservices.log.common.FastLogTag;
import com.aliyun.openservices.log.common.LogGroupData;
import com.aliyun.openservices.loghub.client.LogHubCheckpointTracker;

```

```

import com.aliyun.openservices.loghub.client.ILogHubCheckpointTracker;
import com.aliyun.openservices.loghub.client.exceptions.LogHubCheckpointException;
import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessor;
import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessorFactory;

import java.util.List;

public class SampleLogHubProcessor implements ILogHubProcessor {
    private int shardId;
    // Record the last persistent checkpoint time.
    private long mLastCheckTime = 0;

    public void initialize(int shardId) {
        this.shardId = shardId;
    }

    // The main logic of data consumption. All exceptions must be captured and cannot be thrown.
    public String process(List<LogGroupData> logGroups,
        ILogHubCheckpointTracker checkPointTracker) {
        // Display the retrieved data.
        for (LogGroupData logGroup : logGroups) {
            FastLogGroup flg = logGroup.GetFastLogGroup();
            System.out.println(String.format("\tcategory\t:\t%s\n\tsource\t:\t%s\n\ttopic\t:\t%s\n\tmachineUUID\t:\t%s",
                flg.getCategory(), flg.getSource(), flg.getTopic(), flg.getMachineUUID()));
            System.out.println("Tags");
            for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++tagIdx) {
                FastLogTag logtag = flg.getLogTags(tagIdx);
                System.out.println(String.format("\t%s\t:\t%s", logtag.getKey(), logtag.getValue()));
            }
            for (int lIdx = 0; lIdx < flg.getLogsCount(); ++lIdx) {
                FastLog log = flg.getLogs(lIdx);
                System.out.println("-----\nLog: " + lIdx + ", time: " + log.getTime() + ", GetContentCount: " + log.getContentsCount());
                for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {
                    FastLogContent content = log.getContents(cIdx);
                    System.out.println(content.getKey() + "\t:\t" + content.getValue());
                }
            }
        }
        long curTime = System.currentTimeMillis();
        // Write checkpoints to the server every 30 seconds. If a ClientWorker instance crashes within 30 seconds,
        // a new ClientWorker instance consumes data starting from the last checkpoint. Duplicate data may exist.
        if (curTime - mLastCheckTime > 30 * 1000) {
            try {
                // If the parameter is set to true, checkpoints are updated to the server immediately. If the parameter is
                // set to false, checkpoints are cached locally. The default update interval of checkpoints is 60 seconds.
            }
        }
    }
}

```

```

        checkPointTracker.saveCheckpoint(true);
    } catch (LogHubCheckpointException e) {
        e.printStackTrace();
    }
    mLastCheckTime = curTime;
}
return null;
}

// The ClientWorker instance calls this function upon exit. You can perform a cleanup.
public void shutdown(ILogHubCheckpointTracker checkPointTracker) {
    // Save consumption breakpoints to the server.
    try {
        checkPointTracker.saveCheckpoint(true);
    } catch (LogHubCheckpointException e) {
        e.printStackTrace();
    }
}

class SampleLogHubProcessorFactory implements ILogHubProcessorFactory {
    public ILogHubProcessor generatorProcessor() {
        // Generate a consumption instance.
        return new SampleLogHubProcessor();
    }
}

```

 **Note** Run the preceding code to print all data in a Logstore. If you want multiple consumers to consume a Logstore, you can modify the code based on the comments. You can use the same consumer group name and different consumer names to start a new consumption process.

Limits and troubleshooting

A maximum of 10 consumer groups can be created for each Logstore. The `ConsumerGroupQuotaExceed` error is reported when the number of consumer groups exceeds 10.

We recommend that you configure Log4j for the consumer program to throw error messages within consumer groups for troubleshooting. If you save the `log4j.properties` file to the resources directory and execute the program, you can see the following exception:

```

[WARN ] 2018-03-14 12:01:52,747 method:com.aliyun.openservices.loghub.client.LogHubConsumer.sampleLogError(Log
HubConsumer.java:159)
com.aliyun.openservices.log.exception.LogException: Invalid loggroup count, (0,1000]

```

A typical `log4j.properties` configuration file is described as follows:

```
log4j.rootLogger = info,stdout
log4j.appender.stdout = org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target = System.out
log4j.appender.stdout.layout = org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern = [%-5p] %d{yyyy-MM-dd HH:mm:ss,SSS} method:%l%n%m%n
```

Advanced operations

The preceding code can help you consume data in common scenarios. The following section describes how to perform advanced operations in other scenarios.

- Consume data that is logged from a certain time point

LoghubConfig in the preceding code has two constructors:

```
// The value of the consumerStartTimeInSeconds parameter is a Unix timestamp representing the number of second
s that have elapsed since 00:00:00 on January 1, 1970, 00:00:00 UTC.
public LogHubConfig(String consumerGroupName,
    String consumerName,
    String loghubEndPoint,
    String project, String logStore,
    String accessId, String accessKey,
    int consumerStartTimeInSeconds);

// The position parameter is an enumeration variable. LogHubConfig.ConsumePosition.BEGIN_CURSOR indicates that
the consumption starts from the earliest data.LogHubConfig.ConsumePosition.END_CURSOR indicates that the cons
umption starts from the latest data.
public LogHubConfig(String consumerGroupName,
    String consumerName,
    String loghubEndPoint,
    String project, String logStore,
    String accessId, String accessKey,
    ConsumePosition position);
```

You can use different constructors based on your needs. However, if a checkpoint is stored on the server, you need to start data consumption from the checkpoint.

- Reset a checkpoint

In scenarios such as data padding or repeated computing, you may need to set the consumption position to a time point for a consumer group to start data consumption. To set the consumption position, you can use either of the following two methods:

- Delete the consumer group.
 - a. Stop the consumption processes.
 - b. Delete the consumer group from the console.
 - c. Modify the code to specify the start time point for data consumption.
 - d. Restart the consumption processes.
- Use the SDK to reset the start time point of data consumption for the consumer group.
 - a. Stop the consumption processes.
 - b. Use the SDK to modify the checkpoint.
 - c. Restart the consumption processes.

```

public static void updateCheckpoint() throws Exception {
    Client client = new Client(host, accessId, accessKey);
    long timestamp = Timestamp.valueOf("2017-11-15 00:00:00").getTime() / 1000;
    ListShardResponse response = client.ListShard(new ListShardRequest(project, logStore));
    for (Shard shard : response.GetShards()) {
        int shardId = shard.GetShardId();
        String cursor = client.GetCursor(project, logStore, shardId, timestamp).GetCursor();
        client.UpdateCheckPoint(project, logStore, consumerGroup, shardId, cursor);
    }
}

```

Access consumer groups as a RAM user

Before a RAM user can access consumer groups, relevant permissions must be granted to the user. For more information about how to grant permissions to a RAM user, see [Overview](#).

The following table lists the actions you can take as a RAM user.

Action	Resource
log:GetCursorOrData	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:CreateConsumerGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*
log:ListConsumerGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*
log:ConsumerGroupUpdateCheckPoint	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:ConsumerGroupHeartBeat	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:UpdateConsumerGroup	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:GetConsumerGroupCheckPoint	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}

For example, a project named project-test resides in the China (Hangzhou) region. The ID of the Apsara Stack tenant account to which the project belongs is 1234567. The name of the Logstore to be consumed is logstore-test and the consumer group name is consumergroup-test. To allow a RAM user to access the consumer group, you must grant the following permissions to the user.

```

{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "log:GetCursorOrData"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:CreateConsumerGroup",
        "log:ListConsumerGroup"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test/consumergroup/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ConsumerGroupHeartBeat",
        "log:UpdateConsumerGroup",
        "log:GetConsumerGroupCheckPoint"
      ],
      "Resource": "acs:log:cn-hangzhou:1234567:project/project-test/logstore/logstore-test/consumergroup/consumer
group-test"
    }
  ]
}

```

30.6.3.2. View the status of a consumer group

This topic describes how to use the console, API, and SDK to view the status of a consumer group. The consumer group is an advanced mode of real-time data consumption. It implements automatic load balancing among multiple consumption instances for log data consumption. Spark Streaming and Storm use consumer groups as the basic mode for log data consumption.

View consumption progress in the console

1. [Log on to the Log Service console.](#)
2. Click the  icon. Choose **Logstore > Data Consumption**.
3. Click the consumer group whose data consumption progress you want to view. The data consumption progress of each shard in the Logstore is displayed.

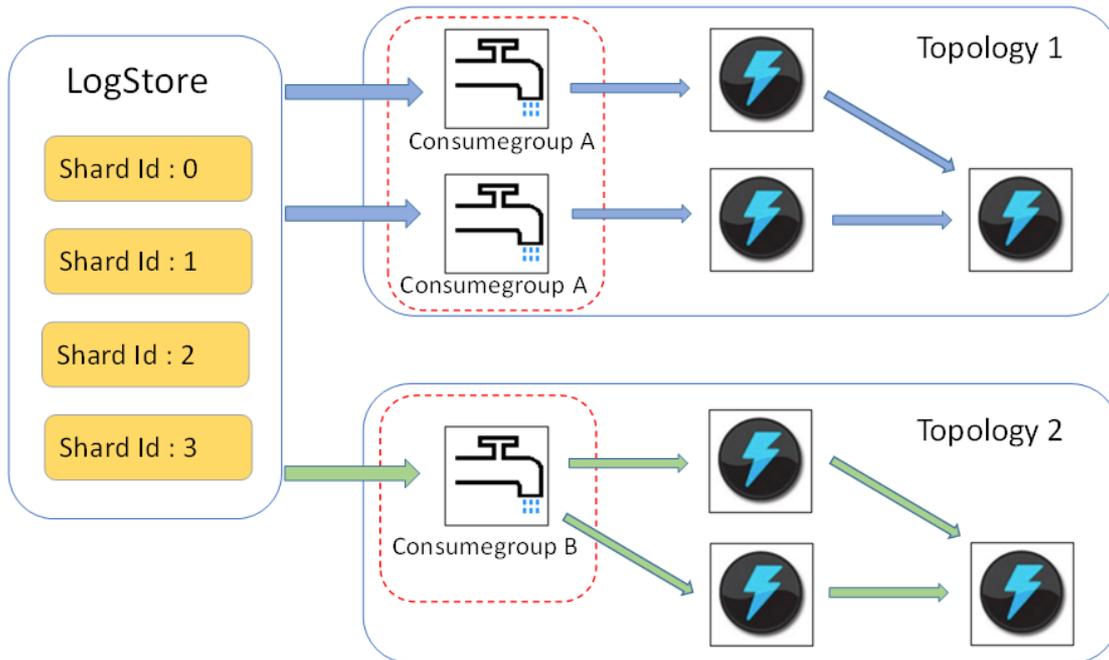
Use the API or SDK to view the data consumption progress

The following example uses the SDK for Java to describe how to call API operations to view the data consumption progress.

```

package test;
import java.util.ArrayList;
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.Consts.CursorMode;
import com.aliyun.openservices.log.common.ConsumerGroup;
import com.aliyun.openservices.log.common.ConsumerGroupShardCheckPoint;
import com.aliyun.openservices.log.exception.LogException;
public class ConsumerGroupTest {
    static String endpoint = "";
    static String project = "";
    static String logstore = "";
    static String accessKeyId = "";
    static String accessKey = "";
    public static void main(String[] args) throws LogException {
        Client client = new Client(endpoint, accessKeyId, accessKey);
        //Query all consumer groups of this Logstore. If no consumer group exists, the length of consumerGroups is 0.
        ArrayList<ConsumerGroup> consumerGroups;
        try{
            consumerGroups = client.ListConsumerGroup(project, logstore).GetConsumerGroups();
        }
        catch(LogException e){
            if(e.GetErrorCode() == "LogStoreNotExist")
                System.out.println("this logstore does not have any consumer group");
            else{
                //internal server error branch
            }
            return;
        }
        for(ConsumerGroup c: consumerGroups){
            //Print consumer group properties, including the name, heartbeat timeout, and whether data is consumed in order.
            System.out.println("Name: " + c.getConsumerGroupName());
            System.out.println("Heartbeat timeout: " + c.getTimeout());
            System.out.println("Consumption in order: " + c.isInOrder());
            for(ConsumerGroupShardCheckPoint cp: client.GetCheckPoint(project, logstore, c.getConsumerGroupName()).GetCheckPoints()){
                System.out.println("shard: " + cp.getShard());
                //Format the returned time. The time is a long integer that is accurate to milliseconds.
                System.out.println("The last time when data is consumed: " + cp.getUpdateTime());
                System.out.println("Consumer name: " + cp.getConsumer());
                String consumerPrg = "";
                if(cp.getCheckPoint().isEmpty())
                    consumerPrg = "Consumption not started";
                else{
                    //The Unix timestamp is seconds. Format the output value of the timestamp

```

Limits

- You can create up to 10 consumer groups to consume log data from a Logstore. If a consumer group is no longer in use, you can call the `DeleteConsumerGroup` operation of the SDK for Java to delete the consumer group.
- We recommend that the number of spouts be equal to the number of shards in a Logstore. This is because a single spout may be unable to process a large amount of data in multiple shards.
- If the data volume in a shard exceeds the processing capacity of a single spout, you can split the shard to reduce its data volume.
- LogHub spouts and bolts must use the `ack` method to check whether log data is successfully sent from spouts to bolts and whether the data is successfully processed by the bolts.

Examples

- Use the following code to create spouts and construct a topology:

```
public static void main( String[] args )
{
    String mode = "Local"; // Specify to use the local test mode.
    String consumer_group_name = ""; // Specify a unique consumer group name for each topology. The name can
    not be an empty string. It must be 3 to 63 characters in length and can contain lowercase letters, digits, hyphens (-),
    and underscores (_). It must start and end with a lowercase letter or digit.
    String project = ""; // Specify the project in Log Service.
    String logstore = ""; // Specify the Logstore in Log Service.
    String endpoint = ""; // Specify the endpoint of Log Service.
    String access_id = ""; // Specify the AccessKey ID of the user.
    String access_key = "";
    // Configure a LogHub Storm spout.
    LogHubSpoutConfig config = new LogHubSpoutConfig(consumer_group_name,
        endpoint, project, logstore, access_id,
        access_key, LogHubCursorPosition.END_CURSOR);
    TopologyBuilder builder = new TopologyBuilder();
    // Create a LogHub Storm spout.
```

```

LogHubSpout spout = new LogHubSpout(config);
// You can create the same number of spouts as that of shards in a Logstore in actual business scenarios.
builder.setSpout("spout", spout, 1);
builder.setBolt("exclaim", new SampleBolt()).shuffleGrouping("spout");
Config conf = new Config();
conf.setDebug(false);
conf.setMaxSpoutPending(1);
// Configure the serialization method of LogGroupData by using the LogGroupDataSerializSerializer class if Kry
o is used to serialize and deserialize data.
Config.registerSerialization(conf, LogGroupData.class, LogGroupDataSerializSerializer.class);
if (mode.equals("Local")) {
    logger.info("Local mode...");
    LocalCluster cluster = new LocalCluster();
    cluster.submitTopology("test-jstorm-spout", conf, builder.createTopology());
    try {
        Thread.sleep(6000 * 1000); //waiting for several minutes
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    cluster.killTopology("test-jstorm-spout");
    cluster.shutdown();
} else if (mode.equals("Remote")) {
    logger.info("Remote mode...");
    conf.setNumWorkers(2);
    try {
        StormSubmitter.submitTopology("stt-jstorm-spout-4", conf, builder.createTopology());
    } catch (AlreadyAliveException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (InvalidTopologyException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
} else {
    logger.error("invalid mode: " + mode);
}
}
}

```

- Use the following example code of bolts to consume log data and display the content of each log entry:

```

public class SampleBolt extends BaseRichBolt {
    private static final long serialVersionUID = 4752656887774402264L;
    private static final Logger logger = Logger.getLogger(BaseBasicBolt.class);
    private OutputCollector mCollector;
    @Override
    public void prepare(@SuppressWarnings("rawtypes") Map stormConf, TopologyContext context,
        OutputCollector collector) {
        mCollector = collector;
    }
    @Override
    public void execute(Tuple tuple) {
        String shardId = (String) tuple
            .getValueByField(LogHubSpout.FIELD_SHARD_ID);
        @SuppressWarnings("unchecked")
        List<LogGroupData> logGroupDatas = (ArrayList<LogGroupData>) tuple.getValueByField(LogHubSpout.FIELD_L
OGGROUPS);
        for (LogGroupData groupData : logGroupDatas) {
            // Each log group consists of one or more log entries.
            LogGroup logGroup = groupData.getLogGroup();
            for (Log log : logGroup.getLogsList()) {
                StringBuilder sb = new StringBuilder();
                // Each log entry has a time field and other key-value pairs.
                int log_time = log.getTime();
                sb.append("LogTime:").append(log_time);
                for (Content content : log.getContentsList()) {
                    sb.append("\t").append(content.getKey()).append(":")
                        .append(content.getValue());
                }
                logger.info(sb.toString());
            }
        }
        // Spouts must use the ack method to indicate whether data has been successfully sent to bolts.
        // In addition, bolts must use the ack method to indicate whether data is successfully processed by the bolts.
        mCollector.ack(tuple);
    }
    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        //do nothing
    }
}

```

Maven

Use the following code to add Maven dependencies for Storm versions earlier than Storm 1.0 (for example, Storm 0.9.6):

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-storm-spout</artifactId>
  <version>0.6.6</version>
</dependency>
```

Use the following code to add Maven dependencies for Storm 1.0 and later versions:

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>loghub-storm-1.0-spout</artifactId>
  <version>0.1.3</version>
</dependency>
```

30.6.5. Use Flume to consume log data

This topic describes how to use Flume to consume log data. You can use the aliyun-log-flume plug-in to connect LogHub of Log Service to Flume, and then write and consume log data.

The aliyun-log-flume plug-in connects LogHub to Flume. When LogHub is connected to Flume, you can connect Log Service to other systems such as HDFS and Kafka through Flume. The aliyun-log-flume plug-in provides the Sink and Source methods to connect Log Service to Flume.

- Sink: uses Flume to read data from other data sources and then writes data to LogHub.
- Source: uses Flume to consume LogHub data and then writes data to other systems.

LogHub Sink

You can use the Sink method to transmit data from other data sources to LogHub through Flume. Data can be parsed into the following two formats:

- SIMPLE: writes a Flume event to LogHub as a field.
- DELIMITED: delimits a Flume event with delimiters, parses an event into fields based on the configured column names, and then writes the fields to LogHub.

The following table lists the parameters you can configure when you use the Sink method to read data.

Parameter	Required	Description
type	Yes	Valid value: com.aliyun.loghub.flume.sink.LoghubSink.
endpoint	Yes	The endpoint of Log Service.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID of your Apsara Stack tenant account.
accessKey	Yes	The AccessKey secret of your Apsara Stack tenant account.
batchSize	No	The number of log entries that are written to LogHub at a time. Default value: 1000.
maxBufferSize	No	The maximum size of the queue in the buffer. Default value: 1000.

Parameter	Required	Description
serializer	No	The serialization format of log data. Valid values: <ul style="list-style-type: none"> DELIMITED: Data is parsed into the DELIMITED format. If you set this parameter to DELIMITED, you must set the columns parameter. SIMPLE: Data is parsed into the SIMPLE format. This is the default value. Custom serializer: Data is parsed into a custom serialization format. If you set this parameter to a custom serializer, you must specify the full name of the class.
columns	No	The configured column names. You must set this parameter if you set the serializer parameter to DELIMITED. Separate multiple columns with commas (,). Ensure that the columns are sorted in the same order as those of the log data.
separatorChar	No	The delimiter, which must be a single character. You can set this parameter if you set the serializer parameter to DELIMITED. Default value: , .
quoteChar	No	The quote character. You can set this parameter if you set the serializer parameter to DELIMITED. Default value: " .
escapeChar	No	The escape character. You can set this parameter if you set the serializer parameter to DELIMITED. Default value: " .
useRecordTime	No	Specifies whether to use the value of the timestamp field as the time when log data is written to Log Service. The default value false indicates that the current time is used.

Loghub Source

You can use the Source method to ship data from LogHub to other data systems through Flume. Data can be output in the following two formats:

- **DELIMITED:** writes delimited log data to Flume.
- **JSON:** writes JSON-formatted log data to Flume.

The following table lists the parameters you can configure when you use the Source method to read data.

Parameter	Required	Description
type	Yes	Valid value: com.aliyun.loghub.flume.source.LoghubSource.
endpoint	Yes	The endpoint of Log Service.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID of your Apsara Stack tenant account.
accessKey	Yes	The AccessKey secret of your Apsara Stack tenant account.

Parameter	Required	Description
heartbeatIntervalMs	No	The heartbeat interval between the Flume client and LogHub. Unit: milliseconds. Default value: 30000.
fetchIntervalMs	No	The interval for pulling data from LogHub. Unit: milliseconds. Default value: 100.
fetchInOrder	No	Specifies whether to consume log data in the order that log data was generated. Default value: false.
batchSize	No	The number of log entries that are read at a time. Default value: 100.
consumerGroup	No	The name of the consumer group that reads data. The name is randomly generated.
initialPosition	No	The start point from which data is read. Valid values: begin, end, and timestamp. Default value: begin. <div style="border: 1px solid #add8e6; padding: 5px; background-color: #e0f0ff;"> <p> Note If a checkpoint exists on the server, the checkpoint is used.</p> </div>
timestamp	No	The Unix timestamp. You must set this parameter if you set the initialPosition parameter to timestamp. Unix timestamp.
deserializer	Yes	The deserialization format of log data. Valid values: <ul style="list-style-type: none"> DELIMITED: Data is parsed into the DELIMITED format. This is the default value. If you set this parameter to DELIMITED, you must set the columns parameter. JSON: Data is parsed into the JSON format. Custom deserializer: Data is parsed into a custom deserialization format. If you set this parameter to a custom deserializer, you must specify the full name of the class.
columns	No	The configured column names. You must set this parameter if you set the deserializer parameter to DELIMITED. Separate multiple columns with commas (.). Ensure that the columns are sorted in the same order as those of the log data.
separatorChar	No	The delimiter, which must be a single character. You can set this parameter if you set the deserializer parameter to DELIMITED. Default value: <code>,</code> .
quoteChar	No	The quote character. You can set this parameter if you set the deserializer parameter to DELIMITED. Default value: <code>"</code> .
escapeChar	No	The escape character. You can set this parameter if you set the deserializer parameter to DELIMITED. Default value: <code>"</code> .
appendTimestamp	No	Specifies whether to append the timestamp as a field to the end of each log entry. You can set this parameter if you set the deserializer parameter to DELIMITED. Default value: false.

Parameter	Required	Description
sourceAsField	No	Specifies whether to add the log source as a field named <code>__source__</code> . You can set this parameter if you set the deserializer parameter to <code>JSON</code> . Default value: <code>false</code> .
tagAsField	No	Specifies whether to add the log tags as a field with the field name <code>__tag__: {tag names}</code> . You can set this parameter if you set the deserializer parameter to <code>JSON</code> . Default value: <code>false</code> .
timeAsField	No	Specifies whether to add the log time as a field named <code>__time__</code> . You can set this parameter if you set the deserializer parameter to <code>JSON</code> . Default value: <code>false</code> .
useRecordTime	No	Specifies whether to use the value of the timestamp field as the time when log data is read from Log Service. The default value <code>false</code> indicates that the current time is used. Default value: <code>false</code> .

30.6.6. Use open source Flink to consume log data

Log Service provides the `flink-log-connector` plug-in to connect with Flink. This topic describes how to integrate the `flink-log-connector` plug-in with Flink to consume log data.

Prerequisites

- An `AccessKey` pair, a Log Service project, and a Logstore are created.
- If you log on to Log Service with a RAM user, relevant permissions to access a Logstore are granted to a RAM user. For more information, see [Grant permissions to a RAM role](#).

Context

The `flink-log-collector` plug-in includes `flink-log-consumer` and `flink-log-producer`.

- The `flink-log-consumer` plug-in reads data from Log Service. This plug-in supports the exactly-once semantics and load balancing among shards.
- The `flink-log-producer` plug-in writes data into Log Service. When you use the `flink-log-producer` plug-in, you must add the following Maven dependencies to a project:

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>flink-log-connector</artifactId>
  <version>0.1.13</version>
</dependency>
<dependency>
  <groupId>com.google.protobuf</groupId>
  <artifactId>protobuf-java</artifactId>
  <version>2.5.0</version>
</dependency>
```

Log Consumer

The `flink-log-consumer` plug-in can consume the log data of a Logstore in Log Service. The exactly-once semantics is achieved during log consumption. The `flink-log-consumer` plug-in detects the change of the number of shards in a Logstore. This increases efficiency.

Each Flink subtask consumes data of some shards in a Logstore. If shards in a Logstore are split or merged, the shards consumed by the subtask also change.

When you use the flink-log-consumer plug-in to consume data from Log Service, you can call the following API operations:

- **GetCursorOrData**

You can call this operation to pull log data from a shard. Frequent API requests may exceed the read speed and IOPS limits of the shard. You can specify the `ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS` and `ConfigConstants.LOG_MAX_NUMBER_PER_FETCH` parameters to control the interval of API requests and number of log entries pulled in each request.

```
configProps.put(ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS, "100");
configProps.put(ConfigConstants.LOG_MAX_NUMBER_PER_FETCH, "100");
```

- **ListShards**

You can call this operation to view all shards in a Logstore and the status of each shard. If the shards are frequently split and merged, you can adjust the call interval to detect the changes in the number of shards.

```
// Call the ListShards operation once every 30 seconds.
configProps.put(ConfigConstants.LOG_SHARDS_DISCOVERY_INTERVAL_MILLIS, "30000");
```

- **CreateConsumerGroup**

You can call this operation to create a consumer group to synchronize checkpoints. This operation can be called only when consumption progress monitoring is enabled.

- **ConsumerGroupUpdateCheckPoint**

You can call this operation to synchronize snapshots of Flink to a consumer group.

The following table lists the Apsara Stack resources required for RAM users to call the preceding API operations.

API	Alibaba Resource Name (ARN)
GetCursorOrData	<code>acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}</code>
ListShards	<code>acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}</code>
CreateConsumerGroup	<code>acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*</code>
ConsumerGroupUpdateCheckPoint	<code>acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}</code>

For more information, see [Grant permissions to a RAM role](#).

1. Configure startup parameters. The following example shows how to consume log data. The `java.util.Properties` class is used as the configuration tool. You can find all constants to be configured in the `ConfigConstants` class.

```

Properties configProps = new Properties();
// Specify the endpoint of Log Service.
configProps.put(ConfigConstants.LOG_ENDPOINT, "cn-hangzhou.log.aliyuncs.com");
// Specify the AccessKey pair.
configProps.put(ConfigConstants.LOG_ACCESSKEYID, "");
configProps.put(ConfigConstants.LOG_ACCESSKEY, "");
// Specify the project.
configProps.put(ConfigConstants.LOG_PROJECT, "ali-cn-hangzhou-sls-admin");
// Specify the Logstore.
configProps.put(ConfigConstants.LOG_LOGSTORE, "sls_consumergroup_log");
// Specify the start position to consume logs.
configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR);
// Specify the data deserialization method.
RawLogGroupListDeserializer deserializer = new RawLogGroupListDeserializer();
final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
DataStream<RawLogGroupList> logTestStream = env.addSource(
    new FlinkLogConsumer<RawLogGroupList>(deserializer, configProps));

```

Note The number of subtasks in the Flink Streaming is independent of the number of shards in a Logstore. If the number of shards is greater than that of subtasks, each subtask consumes multiple shards exactly once. If the number of shards is less than that of subtasks, some subtasks are idle until new shards are generated.

- Specify the start consumption position. The `flink-log-consumer` plug-in enables you to specify the start consumption position of log data in a shard. By specifying the `ConfigConstants.LOG_CONSUMER_BEGIN_POSITION` parameter, you can start data consumption from the earliest, latest, or a specific time point. The `flink-log-connector` plug-in also allows a consumer group to resume consumption from a specific position. Valid values:
 - `Consts.LOG_BEGIN_CURSOR`: The consumption starts from the earliest data.
 - `Consts.LOG_END_CURSOR`: The consumption starts from the latest data.
 - `Consts.LOG_FROM_CHECKPOINT`: The consumption starts from a checkpoint that is stored in a specific consumer group. You can use the `ConfigConstants.LOG_CONSUMERGROUP` parameter to specify the consumer group.
 - `UnixTimestamp`: a string of the `INTEGER` data type. The timestamp is the number of seconds that have elapsed since 00:00:00 January 1, 1970. The value indicates that data in a shard is consumed from this time point.

You can use the following code to specify a start consumption position:

```

configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_BEGIN_CURSOR);
configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR);
configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, "1512439000");
configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_FROM_CHECKPOINT);

```

Note If you have configured consumption resumption from a state backend of Flink when you start a Flink job, the `flink-log-connector` plug-in uses checkpoints stored in the state backend.

- (Optional) Configure consumption progress monitoring. The `flink-log-consumer` plug-in enables you to configure consumption progress monitoring. Consumption progress indicates the real-time consumption position of each shard. These positions are indicated by timestamps. For more information, see [View the status of a consumer group](#).

```
configProps.put(ConfigConstants.LOG_CONSUMERGROUP, "your consumer group name");
```

Note This configuration item is optional. If you specify this configuration item and no consumer group exists, the flink-log-consumer plug-in creates a consumer group. Snapshots in the flink-log-consumer plug-in are automatically synchronized to the consumer group of Log Service, and you can view the consumption progress of the flink-log-consumer plug-in in the Log Service console.

4. Configure consumption resumption and the exactly-once semantics. If the checkpointing feature of Flink is enabled, the flink-log-consumer plug-in periodically stores the consumption progress of each shard. When a job fails, Flink restores the flink-log-consumer plug-in and starts to consume data from the latest checkpoint.

While you configure the checkpointing period, the maximum amount of data to be re-consumed when a failure occurs is defined. You can use the following code to configure the checkpointing period:

```
final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
// Configure the exactly-once semantics.
env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
// Store checkpoints every five seconds.
env.enableCheckpointing(5000);
```

For more information about the Flink checkpoints, see [Checkpoints](#) in the Flink documentation.

Log Producer

The flink-log-producer plug-in writes data into Log Service.

Note The flink-log-producer plug-in supports the Flink at-least-once semantics. If a job fails, data written into Log Service may be duplicated but never lost.

When you use the flink-log-producer plug-in to writes data to Log Service, you can call the following API operations:

- PostLogStoreLogs
- ListShards

The following table lists the Apsara Stack resources required for RAM users to call the preceding API operations.

API	ARN
PostLogStoreLogs	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
ListShards	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}

For more information about RAM users and how to authorize RAM users, see [Grant permissions to a RAM role](#).

1. Initialize the flink-log-producer plug-in.

- i. Configure startup parameters for the flink-log-producer plug-in. The initialization process for the flink-log-producer plug-in is similar to that for the flink-log-consumer plug-in. The following code shows the available parameters that you can configure for the flink-log-producer plug-in. You can use the default values of these parameters. You can also specify the parameters based on your needs.

```
// The number of I/O threads used to send data. Default value: 8.
ConfigConstants.LOG_SENDER_IO_THREAD_COUNT
// The time it takes to send the data after log data is cached. Default value: 3000.
ConfigConstants.LOG_PACKAGE_TIMEOUT_MILLIS
// The number of logs in the cached package. Default value: 4096.
ConfigConstants.LOG_LOGS_COUNT_PER_PACKAGE
// The size of the cached package. Default value: 3 Mbits.
ConfigConstants.LOG_LOGS_BYTES_PER_PACKAGE
// The total memory size that the job can use. Default value: 100 Mbits.
ConfigConstants.LOG_MEM_POOL_BYTES
```

 **Note** These parameters are optional. You can use their default values.

- ii. Reload LogSerializationSchema to define the method of serializing data into RawLogGroup. To use the hash key to specify the shard for data writes, you can use the LogPartitioner method to generate the hash key for the data.

Example:

```
FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializer(), configPro
ps);
logProducer.setCustomPartitioner(new LogPartitioner<String>() {
    // Generate a 32-bit hash value.
    public String getHashKey(String element) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(element.getBytes());
            String hash = new BigInteger(1, md.digest()).toString(16);
            while(hash.length() < 32) hash = "0" + hash;
            return hash;
        } catch (NoSuchAlgorithmException e) {
        }
        return "0000000000000000000000000000000000000000000000000000000000000000000000000000";
    }
});
```

 **Note** The LogPartitioner method is optional. If you do not configure this method, data is randomly written into a shard.

2. Run the following code and write the generated strings to Log Service.

```
// Serialize data into the format of raw log groups.
class SimpleLogSerializer implements LogSerializationSchema<String> {
    public RawLogGroup serialize(String element) {
        RawLogGroup rlg = new RawLogGroup();
        RawLog rl = new RawLog();
```

```

        rl.setTime((int)(System.currentTimeMillis() / 1000));
        rl.addContent("message", element);
        rlg.addLog(rl);
        return rlg;
    }
}
public class ProducerSample {
    public static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
    public static String sAccessKeyId = "";
    public static String sAccessKey = "";
    public static String sProject = "ali-cn-hangzhou-sls-admin";
    public static String sLogstore = "test-flink-producer";
    private static final Logger LOG = LoggerFactory.getLogger(ConsumerSample.class);
    public static void main(String[] args) throws Exception {
        final ParameterTool params = ParameterTool.fromArgs(args);
        final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
        env.getConfig().setGlobalJobParameters(params);
        env.setParallelism(3);
        DataStream<String> simpleStringStream = env.addSource(new EventsGenerator());
        Properties configProps = new Properties();
        // Specify the endpoint of Log Service.
        configProps.put(ConfigConstants.LOG_ENDPOINT, sEndpoint);
        // Specify the AccessKey pair to access Log Service.
        configProps.put(ConfigConstants.LOG_ACCESSKEYID, sAccessKeyId);
        configProps.put(ConfigConstants.LOG_ACCESSKEY, sAccessKey);
        // Specify the project to which logs are written.
        configProps.put(ConfigConstants.LOG_PROJECT, sProject);
        // Specify the Logstore to which logs are written.
        configProps.put(ConfigConstants.LOG_LOGSTORE, sLogstore);
        FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializer(), configPro
ps);
        simpleStringStream.addSink(logProducer);
        env.execute("flink log producer");
    }
    // Simulate log generation.
    public static class EventsGenerator implements SourceFunction<String> {
        private boolean running = true;
        @Override
        public void run(SourceContext<String> ctx) throws Exception {
            long seq = 0;
            while (running) {
                Thread.sleep(10);
                ctx.collect((seq++) + "-" + RandomStringUtils.randomAlphabetic(12));
            }
        }
        @Override
        public void cancel() {

```

```

public void cancel() {
    running = false;
}
}
}

```

30.6.7. Use Logstash to consume log data

Log Service allows you to use SDKs developed in various languages and various stream computing systems to consume log data. In addition, Log Service allows you to use Logstash to consume log data. You can configure the Logstash input plug-in to read log data from Log Service and write the data to other systems, such as Kafka and Hadoop Distributed File System (HDFS).

Features

- **Distributed collaborative consumption:** Multiple servers can be configured to consume log data from a Logstore at the same time.
- **High performance:** If you use consumer groups written in Java to consume log data, the consumption speed of a CPU core can reach up to 20 MB/s.
- **High reliability:** Log Service saves the consumption progress. This mechanism enables automatic resumption of log consumption from the last checkpoint after a consumption exception is solved.
- **Automatic load balancing:** Shards are automatically allocated based on the number of consumers in a consumer group. The shards are reallocated if consumers join or leave the consumer group.

30.6.8. Use Spark Streaming to consume log data

This topic describes how to use Spark Streaming to consume log data. After logs are collected to Log Service, you can use the Spark SDK provided by Log Service to process log data in Spark Streaming.

The Spark SDK supports two consumption modes: Receiver and Direct.

The Maven dependency is as follows:

```

<dependency>
<groupId>com.aliyun.emr</groupId>
<artifactId>emr-logservice_2.11</artifactId>
<version>1.7.2</version>
</dependency>

```

Receiver mode

In the Receiver mode, a consumer group consumes data from Log Service and temporarily stores the data in Spark Executor. After a Spark Streaming job is started, it reads and processes data from Spark Executor. For more information, see [Use consumer groups to consume log data](#). Each log entry is returned as a JSON string. The consumer group periodically saves checkpoints to Log Service. You do not need to update checkpoints.

- **Example code**

```

import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.aliyun.logservice.LoghubUtils
import org.apache.spark.streaming.{ Milliseconds, StreamingContext}
import org.apache.spark.SparkConf

object TestLoghub {
    def main(args: Array[String]): Unit = {
        if (args.length < 7) {

```

```

System.err.println(
  """Usage: TestLoghub <project> <logstore> <loghub group name> <endpoint>
  |   <access key id> <access key secret> <batch interval seconds>
  """
  .stripMargin)
System.exit(1)
}

val project = args(0)
val logstore = args(1)
val consumerGroup = args(2)
val endpoint = args(3)
val accessKeyId = args(4)
val accessKeySecret = args(5)
val batchInterval = Milliseconds(args(6).toInt * 1000)

def functionToCreateContext(): StreamingContext = {
  val conf = new SparkConf().setAppName("Test Loghub")
  val ssc = new StreamingContext(conf, batchInterval)
  val loghubStream = LoghubUtils.createStream(
    ssc,
    project,
    logstore,
    consumerGroup,
    endpoint,
    accessKeyId,
    accessKeySecret,
    StorageLevel.MEMORY_AND_DISK)

  loghubStream.checkpoint(batchInterval * 2).foreachRDD(rdd =>
    rdd.map(bytes => new String(bytes)).top(10).foreach(println)
  )
  ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory
  ssc
}

val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext _)

ssc.start()
ssc.awaitTermination()
}
}

```

- Parameter description

Parameter	Type	Description
project	String	The project in Log Service.

Parameter	Type	Description
logstore	String	The Logstore in Log Service.
consumerGroup	String	The name of the consumer group.
endpoint	String	The endpoint of the region to which the project belongs.
accessKeyId	String	The AccessKey ID used to access Log Service.
accessKeySecret	String	The AccessKey secret used to access Log Service.

- **Notes**

In the Receiver mode, data loss may occur in some cases. To avoid data loss, you can turn on the Write-Ahead Logs switch, which is supported in Spark 1.2 and later versions. For more information, visit [Spark Streaming Programming Guide](#).

Direct mode

In the Direct mode, no consumer group is required. API operations are called to request data from Log Service. Compared with the Receiver mode, the Direct mode has the following benefits:

- **Simplified parallelism.** The number of Spark partitions is the same as the number of shards in a Logstore. You can split shards to improve the parallelism of tasks.
- **Increased efficiency.** You no longer need to turn on the Write-Ahead Logs switch to prevent data loss.
- **Exactly-once semantics.** Data is read directly from Log Service. Checkpoints are submitted after the task is successful. In some cases, data may be repeatedly consumed if the task is not ended due to unexpected exit of Spark.

You must configure the ZooKeeper service when you use the Direct mode. You must set a checkpoint directory in ZooKeeper to store intermediate state data. If you want to re-consume data after restarting a task, you must delete the directory from ZooKeeper and change the name of the consumer group.

- **Example code**

```
import com.aliyun.openservices.loghub.client.config.LogHubCursorPosition
import org.apache.spark.SparkConf
import org.apache.spark.streaming.{ Milliseconds, StreamingContext}
import org.apache.spark.streaming.aliyun.logservice.{ CanCommitOffsets, LoghubUtils}

object TestDirectLoghub {
  def main(args: Array[String]): Unit = {
    if (args.length < 7) {
      System.err.println(
        """Usage: TestDirectLoghub <project> <logstore> <loghub group name> <endpoint>
        |   <access key id> <access key secret> <batch interval seconds> <zookeeper host:port=localhost:2181>
        """.stripMargin)
      System.exit(1)
    }

    val project = args(0)
    val logstore = args(1)
    val consumerGroup = args(2)
```

```

val endpoint = args(3)
val accessKeyId = args(4)
val accessKeySecret = args(5)
val batchInterval = Milliseconds(args(6).toInt * 1000)
val zkAddress = if (args.length >= 8) args(7) else "localhost:2181"

def functionToCreateContext(): StreamingContext = {
  val conf = new SparkConf().setAppName("Test Direct Loghub")
  val ssc = new StreamingContext(conf, batchInterval)
  val zkParas = Map("zookeeper.connect" -> zkAddress,
    "enable.auto.commit" -> "false")
  val loghubStream = LoghubUtils.createDirectStream(
    ssc,
    project,
    logStore,
    consumerGroup,
    accessKeyId,
    accessKeySecret,
    endpoint,
    zkParas,
    LogHubCursorPosition.END_CURSOR)

  loghubStream.checkpoint(batchInterval).foreachRDD(rdd => {
    println(s"count by key: ${rdd.map(s => {
      s.sorted
      (s.length, s)
    }).countByKey().size}")
    loghubStream.asInstanceOf[CanCommitOffsets].commitAsync()
  })
  ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory
  ssc
}

val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext _)
ssc.start()
ssc.awaitTermination()
}
}

```

- Parameter description

Parameter	Type	Description
project	String	The project in Log Service.
logstore	String	The Logstore in Log Service.

Parameter	Type	Description
consumerGroup	String	The name of the consumer group (only used to save consumption checkpoints).
endpoint	String	The endpoint of the region to which the project belongs.
accessKeyId	String	The AccessKey ID used to access Log Service.
accessKeySecret	String	The AccessKey secret used to access Log Service.
zkAddress	String	The endpoint of ZooKeeper.

- Parameter settings

In the Direct mode, you must specify the number of log entries that are consumed in each shard in each batch. Otherwise, the data reading process cannot be ended. You can throttle the transmission rate of a single batch by setting the two parameters listed in the following table.

Parameter	Description	Default value
spark.loghub.batchGet.step	The number of log groups returned for a single request.	100
spark.streaming.loghub.maxRatePerShard	The maximum number of log entries that are read in a shard.	10,000

The number of log entries processed in each batch is calculated as follows: $\text{number of shards} \times \max(\text{spark.loghub.batchGet.step} \times n \times \text{number of log entries in a log group}, \text{spark.streaming.loghub.maxRatePerShard} \times \text{duration})$.

- n : the number of requests required to increase the returned rows to $\text{spark.streaming.loghub.maxRatePerShard} \times \text{duration}$.
- duration: the interval between batch processing. Unit: milliseconds.

If you need to combine multiple data streams, the number of shards refers to the total number of shards in all Logstores.

- Example

For example, the number of shards is 100. Each log group contains 50 log entries. Batches are processed at an interval of two seconds. If you want to process 20,000 log entries in each batch, use the following configurations:

- `spark.loghub.batchGet.step: 4`
- `spark.streaming.loghub.maxRatePerShard: 100`

If each log group contains 60 log entries and you want to process 20,000 log entries in each batch, 24,000 log entries will be processed based on the preceding configurations ($60 \times 4 \times 100 = 24,000$).

- Accurate transmission rate throttling

A smaller `spark.loghub.batchGet.step` value increases the accuracy of throttling and the number of requests. We recommend that you count the average number of log entries contained in a log group and then set the preceding two parameters.

30.6.9. Use Realtime Compute to consume log data

You can use Realtime Compute (Blink) to create a schema for Log Service data and consume the data.

Log Service stores streaming data. Therefore, Realtime Compute can use the streaming data as input data. In Log Service, each log entry contains multiple fields that are key-value pairs. The following example is a sample log entry:

```
__source__: 11.85.123.199
__tag__:__receive_time__: 1562125591
__topic__: test-topic
a: 1234
b: 0
c: hello
```

You can use the following data definition language (DDL) statement to create a table in Realtime Compute:

```
create table sls_stream(
  a int,
  b int,
  c varchar
) with (
  type ='sls',
  endPoint ='<your endpoint>',
  accessId ='<your access key id>',
  accessKey ='<your access key>',
  startTime = '2017-07-05 00:00:00',
  project ='ali-cloud-streamtest',
  logStore ='stream-test',
  consumerGroup ='consumerGroupTest1'
);
```

Attribute fields

Realtime Compute can extract fields from the log content. In addition, Realtime Compute can extract three fields and custom fields in tags, such as `__receive_time__`. The following table lists the three fields.

Attribute fields

Field name	Description
<code>__source__</code>	The source of the log entry.
<code>__topic__</code>	The topic of the log entry.
<code>__timestamp__</code>	The time when the log entry is generated.

To extract the preceding fields, you must add HEADERS in the DDL statement. Example:

```

create table sls_stream(
  __timestamp__ bigint HEADER,
  __receive_time__ bigint HEADER
  b int,
  c varchar
) with (
  type ='sls',
  endPoint = '<your endpoint>',
  accessId = '<your access key id>',
  accessKey = '<your access key>',
  startTime = '2017-07-05 00:00:00',
  project = 'ali-cloud-streamtest',
  logStore = 'stream-test',
  consumerGroup = 'consumerGroupTest1'
);

```

Parameters in the WITH clause

The following table describes the parameters in the WITH clause.

Parameter	Required	Description
endPoint	Yes	The endpoint of Log Service.
accessId	Yes	The AccessKey ID of the Apsara Stack tenant account or RAM user that is used to access Log Service.
accessKey	Yes	The AccessKey secret of the Apsara Stack tenant account or RAM user that is used to access Log Service.
project	Yes	The name of the project in Log Service.
logStore	Yes	The name of the Logstore in Log Service.
consumerGroup	No	The name of the consumer group.
startTime	No	The time when Realtime Compute starts to consume data.
heartBeatIntervalMills	No	The heartbeat interval of the client that consumes log data. Unit: seconds. Default value: 10 s.
maxRetryTimes	No	The maximum number of retries to read data. Default value: 5.
batchGetSize	No	The number of log groups that are read at a time. Default value: 10. If the Blink version is 1.4.2 or later, the default value is 100 and the maximum value is 1000.

Parameter	Required	Description
columnErrorDebug	No	Specifies whether to enable debugging. If debugging is enabled, log entries that fail to be parsed are displayed. The default value is false.

Field type mapping

All log fields in Log Service are of the string type. The following table lists the mapping between the type of Log Service fields and the type of Realtime Compute fields. We recommend that you declare the mapping in a data definition language (DDL) statement.

Log Service data type	Realtime Compute data type
STRING	VARCHAR

If you specify another data type to convert Log Service data, an automatic conversion attempt is performed. For example, you can specify BIGINT as the data type to convert the string "1000" and specify timestamp as the data type to convert the string "2018-01-12 12:00:00".

Note

- Blink versions earlier than 2.2.0 do not support shard scaling. If you split or merge shards when a job is reading data from a Logstore, the job fails and cannot continue. In this case, you must restart the job.
- None of the Blink versions allow you to delete or re-create a Logstore whose log data is being consumed.
- For Blink 1.6.0 and earlier versions, if you specify a consumer group to consume log data from a Logstore that contains a large number of shards, the read performance may be affected.
- Log Service data cannot be converted to data of the map type when you create a schema.
- Nonexistent fields are set to null.
- We recommend that you convert the fields in the same order as the fields in the schema. Unordered field conversions are also supported.
- The batchGetSize parameter specifies the number of log groups that are read at a time. If the size of each log entry and the value of the batchGetSize parameter are both large, garbage collection (GC) of data in the memory may frequently occur.

Precautions

- If no new data is written to a shard, the overall latency of a job increases. In this case, you need to adjust the number of concurrent tasks in the job to the same as the number of shards from which data is read and written.
- We recommend that you set the number of concurrent tasks in a job to the same as the number of shards. If they are inconsistent, data may be filtered out when the job reads historical data from two shards at significantly different speeds.
- To extract fields in tags such as `__tag__:__hostname__` and `__tag__:__path__`, you can delete the `__tag__:` prefix and use the method of extracting attribute fields.

 **Note** This type of data cannot be extracted during debugging. We recommend that you use the local debugging method and the print method to display data in logs.

30.7. RAM

30.7.1. Overview

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

You can use RAM to manage users, including employees, systems, and applications. You can also use RAM to grant users permissions to access resources.

RAM provides the following features:

- RAM role

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

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

- User group

You can create multiple RAM users for an organization and grant the users different permissions on the same cloud resources in the organization.

 **Note** For more information about how to create users and user groups, see *Users and User groups in the ASCM User Guide*.

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

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

30.7.2. Create a RAM role

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

Procedure

1. Log on to the ASCM console [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create RAM Role**.
5. On the **Roles - Create RAM Role** page that appears, set **Role Name** and **Description**.
6. Click **Create**.

30.7.3. Create a user

This topic describes how an administrator creates a user and assigns a role to the user. The role varies based on the cloud resources that the user needs to access.

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. On the **Users** page, click **Create**.
5. In the dialog box that appears, set the parameters based on your requirements.

Parameter	Description
Username	The username.

Parameter	Description
Display Name	The display name of the user.
Roles	The role that you want to assign to the user.
Organization	The organization to which the user belongs.
Logon Policy	The logon policy that restricts the logon time and IP address of the user. If you do not specify this parameter, the default policy is attached to the created user.
Mobile Number	The mobile phone number of the user. If you need to send text messages about the usage and requests for resources to the mobile phone number, make sure that the specified mobile phone number is correct.
Landline Number	Optional. The landline number of the user.
Email	The email address of the user. If you need to send emails about the usage and requests for resources to the email address, make sure that the specified email address is correct.
DingTalk Key	Optional. The DingTalk key.
Notify User by SMS	Specifies whether to send text messages about the usage and requests for resources to the specified mobile phone number.
Notify User by Email	Specifies whether to send emails about the usage and requests for resources to the specified email address.
Notify User by DingTalk	Specifies whether to send messages about the usage and requests for resources to the specified DingTalk user.

6. Click OK.

30.7.4. Create a RAM user group

This topic describes how to create a RAM user group in an organization and grant permissions to RAM users in the RAM user group.

Prerequisites

An organization is created.

Context

The relationship between RAM user groups and RAM users:

- A RAM user group can contain zero or more RAM users.
- A RAM user can belong to no RAM user groups.
- You can add a RAM user to multiple RAM user groups.

The relationship between RAM user groups and organizations:

- A RAM user group belongs to only one organization.
- You can create multiple RAM user groups in an organization.

The relationship between RAM user groups and RAM roles:

- Only one RAM role can be assigned to each RAM user group.
- A RAM role can be assigned to multiple RAM user groups.
- When a RAM role is assigned to a RAM user group, the permissions that the RAM role has are automatically granted to RAM users in the RAM user group.

The relationship between RAM user groups and resource sets:

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

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. In the upper-right corner of the page, click **Create User Group**.
5. In the dialog box that appears, specify the **User Group Name** and **Organization** parameters.
6. Click **OK**.

30.7.5. Add a RAM user to a RAM user group

This topic describes how to add a RAM user to a RAM user group.

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Users**.
4. Find the user group to which you want to add users, and click **Add User** in the **Actions** column.
5. In the dialog box that appears, select a username from the left pane, and click the right arrow to move them to the right pane.
6. Click **OK**.

30.7.6. Create a permission policy

If you want to use Log Service to access other cloud resources, you must create a permission policy for a RAM role. Then, the policy is automatically attached to the RAM user group to which the RAM role is assigned.

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the target RAM role, click **More** in the **Actions** column, and choose **Modify** from the drop-down list to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Create Policy**.
7. In the dialog box that appears, enter the policy information.

Add Permission Policy [X]

*Policy Name:
 0/15

Description:
 0/100

*Policy Details:

```
1 | The details of the specified policy must be 2,048 characters in length, and follow the JSON format.
```

[OK] [Cancel]

For more information about how to specify the policy information, see [Use custom policies to grant RAM user the required permissions](#).

30.7.7. Grant permissions to a RAM role

This topic describes how to grant permissions to a RAM role. After a RAM role is granted permissions, the RAM users in the associated RAM user group inherit the permissions.

Procedure

1. Log on to the [ASCM console](#) as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Roles**.
4. In the role name list, find the target RAM role, click **More** in the **Actions** column, and choose **Modify** from the drop-down list to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Select Existing Permission Policy**.
7. In the dialog box that appears, select a permission policy and click **OK**. If no policies are available, create a policy. For more information, see [Create a permission policy](#).

30.7.8. Use custom policies to grant RAM user the required permissions

This topic describes how to use custom Resource Access Management (RAM) policies to grant RAM users the required permissions. You can grant permissions to the RAM users under your Apsara Stack tenant account.

Context

For data security, we recommend that you follow the principle of least privilege (PoLP) when you grant permissions to RAM users. You must grant the read-only permission on the project list to RAM users. Otherwise, the RAM users cannot view the projects in the project list.

Use the RAM console to grant permissions to a RAM user

- The read-only permission on projects

For example, you can use your Apsara Stack tenant account to grant RAM users the following permissions:

- The permission to view the list of projects that belong to the Apsara Stack tenant account
- The permission to read specific projects

Use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": ["log:ListProject"],
      "Resource": ["acs:log:*:*:project/*"],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/*",
      "Effect": "Allow"
    }
  ]
}
```

- The permission to read a Logstore, save searches, and use saved searches.

For example, you can use your Apsara Stack tenant account to grant RAM users the following permissions:

- The permission to view the project list of the Apsara Stack tenant account
- The permission to read a Logstore, save searches, and use saved searches

Use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListProject"
      ],
      "Resource": "acs:log:*:*:project/*",
      "Effect": "Allow"
    },
    {
      "Action": [
```

```

    "log:List*",
  ],
  "Resource": "acs:log:*:*:project/<The name of the project>/logstore/*",
  "Effect": "Allow"
},
{
  "Action": [
    "log:Get*",
    "log:List*"
  ],
  "Resource": [
    "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "log:List*"
  ],
  "Resource": [
    "acs:log:*:*:project/<The name of the project>/dashboard",
    "acs:log:*:*:project/<The name of the project>/dashboard/*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "log:Get*",
    "log:List*",
    "log:Create*"
  ],
  "Resource": [
    "acs:log:*:*:project/<The name of the project>/savedsearch",
    "acs:log:*:*:project/<The name of the project>/savedsearch/*"
  ],
  "Effect": "Allow"
}
]
}

```

 **Note** In the policy, a value of the Resource attribute that does not end with an asterisk (*) indicates the exact resource. A value that ends with an asterisk (*) indicates all resources that match the value.

- The permission to read a Logstore and view all saved searches and dashboards in a project

For example, you can use your Apsara Stack tenant account to grant a RAM user the following permissions:

- The permission to view the project list of the Apsara Stack tenant account

- The permission to read a Logstore and view all saved searches and dashboards in a project

Use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListProject"
      ],
      "Resource": "acs:log:*:*:project/**",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/logstore/**",
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/dashboard",
        "acs:log:*:*:project/<The name of the project>/dashboard/**"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:Get*",
        "log:List*"
      ],
      "Resource": [
```

```

    "acs:log:*:*:project/<The name of the project>/savedsearch",
    "acs:log:*:*:project/<The name of the project>/savedsearch/*"
  ],
  "Effect": "Allow"
}
]
}

```

Grant RAM users the permissions that are required to call Log Service operations

- The permission to write data to a project

To grant RAM users only the permission to write data to a project, use the following policy:

```

{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:Post*"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/*",
      "Effect": "Allow"
    }
  ]
}

```

- The permission to consume data of a project

To grant RAM users only the permission to consume data of a project, use the following policy:

```

{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListShards",
        "log:GetCursorOrData",
        "log:GetConsumerGroupCheckPoint",
        "log:UpdateConsumerGroup",
        "log:ConsumerGroupHeartBeat",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ListConsumerGroup",
        "log:CreateConsumerGroup"
      ],
      "Resource": "acs:log:*:*:project/<The name of the project>/*",
      "Effect": "Allow"
    }
  ]
}

```

- The permission to consume data of a Logstore

To grant RAM users only the permission to consume data of a Logstore, use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListShards",
        "log:GetCursorOrData",
        "log:GetConsumerGroupCheckPoint",
        "log:UpdateConsumerGroup",
        "log:ConsumerGroupHeartBeat",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ListConsumerGroup",
        "log:CreateConsumerGroup"
      ],
      "Resource": [
        "acs:log:*:*:project/<The name of the project>/logstore/<The name of the Logstore>",
        "acs:log:*:*:project/<the name of the project>/logstore/<the name of the Logstore>/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

30.8. FAQ

30.8.1. Log collection

30.8.1.1. How do I troubleshoot Logtail collection errors?

If the Logtail preview page is blank or "No Data" is displayed on the query page, perform the following steps:

Procedure

1. Check whether Log Service receives heartbeats from the server group.

You can view the Logtail heartbeat status in the Log Service console. For more information, see [View the status of a server group](#).

If the heartbeat status is OK, go to the next step. If the heartbeat status is FAIL, proceed with further troubleshooting. For more information, see [What can I do if no heartbeat packet is received from a Logtail client?](#).

2. Check whether the Logtail configuration is created. If the heartbeat status of Logtail is OK, check whether the Logtail configuration is created. Make sure that the path and name of monitored logs match the files that are stored on the server. The path can be a full path or a path that includes wildcards.
3. Make sure that the Logtail configuration is applied to the server group. Check whether the target Logtail configuration is applied to the server group. For more information, see [Manage server group configurations](#) and [Manage server group](#).
4. Check collection errors. If Logtail is configured correctly, check whether new logs are generated in real

time. Logtail only collects incremental log data. Logtail does not read log files in which no log is generated. If a log file is updated but the updates cannot be queried in Log Service, you can diagnose the problem as follows:

- View logs of the Logtail client

The client logs include key INFO logs, all WARNING logs, and all ERROR logs. To view complete error information in real time, check the following client logs:

- Linux: `/usr/local/ilogtail/ilogtail.LOG`.
- Linux: `/usr/local/ilogtail/logtail_plugin.LOG`. The file contains the logs such as HTTP logs, MySQL binary logs, and MySQL query results.
- 64-bit Windows: `C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log`.
- 32-bit Windows: `C:\Program Files\Alibaba\Logtail\logtail_*.log`.

- Check whether the log volume exceeds the limit.

To collect large volumes of logs, you may need to modify the Logtail startup parameters for higher log collection throughput. For more information, see [Set Logtail startup parameters](#).

30.8.1.2. What can I do if Log Service does not receive heartbeats from a Logtail client?

If Log Service does not receive heartbeats from a Logtail client, perform the steps that are described in the topic to troubleshoot the problem.

Context

After Logtail is installed on a server, the Logtail client sends heartbeats to Log Service. If the status page of the machine group shows that Log Service does not receive heartbeats from a Logtail client, it indicates that the Logtail client is not installed or disconnected from the server.

Step 1: Check whether Logtail is installed

Use the following method to check whether Logtail is installed:

- On a Linux server, run the following command:

```
sudo /etc/init.d/ilogtailed status
```

If the command returns `ilogtail is running`, it indicates that Logtail is installed. The following script shows an example command and response:

```
[root@*****~]# sudo /etc/init.d/ilogtailed status
ilogtail is running
```

- On a Windows server:
 - i. Press Win+R. In the Run dialog box, enter `services.msc` and click **OK**.
 - ii. In the **Services** window, check the status of the `LogtailDaemon` and `LogtailWorker` services. If the services are in the **Running** state, it indicates that Logtail is installed.

If Logtail is not installed, install it. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#). Ensure that the Log Service endpoint in the Logtail installation command corresponds to the region to which the Log Service project belongs. If Logtail is running, go to the next step.

Step 2: Check the Log Service endpoint in the Logtail installation command

When you install Logtail, you must specify a [Log Service endpoint](#) based on the region to which the Log Service project belongs. If the endpoint is incorrect or the Logtail installation command is invalid, Log Service cannot receive heartbeats from the Logtail client.

You can view the Log Service endpoint and installation method in the Logtail configuration file named `ilogtail_config.json`. The file is stored in the following path:

- Linux: `/usr/local/ilogtail/ilogtail_config.json`
- 64-bit Windows: `C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json`
- 32-bit Windows: `C:\Program Files\Alibaba\Logtail\ilogtail_config.json`

In the Logtail configuration file, check the value of the `config_server_address` parameter. This parameter specifies the Log Service endpoint. Then, check whether the Logtail client can connect to Log Service based on the endpoint. For example, if the endpoint that is recorded in the Logtail configuration file is `logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com`, you can run the following command to check the connection:

- Linux:

```
curl logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com
```

- Windows:

```
telnet logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com 80
```

If the Log Service endpoint in the Logtail installation command is incorrect, re-install Logtail. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

If the Log Service endpoint in the Logtail installation command is correct, go to the next step.

Step 3: Check the server IP addresses in the machine group

The server IP address that is obtained by a Logtail client must be configured in the machine group. Otherwise, Log Service cannot receive heartbeats or collect logs from the Logtail client. Logtail uses the following methods to obtain the IP address of a server:

- If the server is not bound with a hostname, Logtail obtains the IP address of the first network interface controller (NIC) card of the server.
- If the server is bound with a hostname, Logtail obtains the IP address that corresponds to the hostname. You can view the hostname and IP address in the `/etc/hosts` file.

 **Note** You can run the `hostname` command to query the hostname.

Perform the following steps to check whether the server IP address that is obtained by the Logtail client is configured in the machine group.

1. Check the server IP address that is obtained by Logtail.

The `ip` field in the `app_info.json` file indicates the server IP address that is obtained by Logtail. The file is stored in the following path:

- Linux: `/usr/local/ilogtail/app_info.json`
- 64-bit Windows: `C:\Program Files (x86)\Alibaba\Logtail\app_info.json`
- 32-bit Windows: `C:\Program Files\Alibaba\Logtail\app_info.json`

 **Note**

- If the `ip` field in the `app_info.json` file is empty, Logtail cannot work. In this case, you must configure an IP address for the server and restart Logtail.
- The `app_info.json` file is used only to record information. If you modify the IP address in the file, the server IP address obtained by Logtail is not updated.

2. Check the server IP addresses in the machine group. Log on to the Log Service console. In the **Projects** section, click the project to which the machine group belongs. In the left-side navigation pane, click the **Machine Groups** icon. In the **Machine Groups** pane, click the machine group. In the **Machine Group Status**

section of the **Machine Group Settings** page, check the server IP addresses.

If no server IP address in the machine group is the same as the IP address that is obtained by Logtail, perform the following step to modify the IP address configurations in the Log Service console:

- If a server IP address in the machine group is incorrect, change the IP address to the IP address that is obtained by Logtail. Then, check the heartbeat status 1 minute after you save the change.
- If you have modified the IP address of the server where Logtail is installed (for example, the `/etc/hosts` file is modified), restart Logtail. After Logtail obtains the new server IP address, set a server IP address in the machine group to the value of the `ip` field in the `app_info.json` file.

You can use the following method to restart Logtail:

- On a Linux server, run the following commands:

```
sudo /etc/init.d/ilogtaild stop
sudo /etc/init.d/ilogtaild start
```

- On a Windows server:

Press Win+R. In the Run dialog box, enter `services.msc` and click OK. In the **Services** window, find and restart the **LogtailWorker** service.

30.8.1.3. How do I query the local log collection statuses?

You can use Logtail to query the health status of Logtail and log collection statuses. The statuses help you troubleshoot log collection issues and customize status monitoring for log collection.

Instructions

After a Logtail client that supports the status query feature is installed, you can query the local log collection statuses by running commands on the client. For more information about how to install Logtail, see [Install Logtail in Linux](#).

You can run the `/etc/init.d/ilogtaild -h` command on the client to check whether a client supports the feature of querying the local log collection status. If the command output includes the `logtail insight, version` keyword, it indicates that the client supports the status query feature.

```
/etc/init.d/ilogtaild -h
Usage: ./ilogtaild { start | stop (graceful, flush data and save checkpoints) | force-stop | status | -h for help}$
logtail insight, version : 0.1.0
command list :
  status all [index]
    get logtail running status
  status active [--logstore | --logfile] index [project] [logstore]
    list all active logstore | logfile. if use --logfile, please add project and logstore. default --logstore
  status logstore [--format=line | json] index project logstore
    get logstore status with line or json style. default --format=line
  status logfile [--format=line | json] index project logstore fileFullPath
    get log file status with line or json style. default --format=line
  status history beginIndex endIndex project logstore [fileFullPath]
    query logstore | logfile history status.
index : from 1 to 60. in all, it means last $(index) minutes; in active/logstore/logfile/history, it means last $(index)*10
minutes
```

The following table describes the commands that are supported by Logtail:

Command	Function	Maximum time range that can be queried	Time window
all	Queries the status of Logtail.	Last 60 minutes	1 minute
active	Queries the active Logstores that are collecting logs and the active log files from which logs are being collected.	Last 600 minutes	10 minutes
logstore	Queries the collection status of a Logstore.	Last 600 minutes	10 minutes
logfile	Queries the collection status of a log file.	Last 600 minutes	10 minutes
history	Queries the collection status of a Logstore or log file in the query time window.	Last 600 minutes	10 minutes

 Note

- The `index` parameter in the preceding commands indicates the index of the time window. Valid values: 1 to 60. The index of the latest time window is 1 and the time window ends at the current system time. If you specify a 1-minute time window, the status in the past interval of `(index, index-1]` minutes is returned. If you specify a 10-minute time window, the status in the past interval of `(10*index, 10*(index-1)]` minutes is returned.
- All commands in the preceding table is the subcommands of the status command.

Command all

Command syntax

```
/etc/init.d/ilogtaild status all [ index ]
```

 Note The all command is used to query the status of Logtail. The index parameter is optional. Default value: 1.

Examples

```
/etc/init.d/ilogtaild status all 1
ok
/etc/init.d/ilogtaild status all 10
busy
```

Response

Status	Description	Priority	Troubleshooting
ok	Logtail is running as expected.	N/A	No action is required.

Status	Description	Priority	Troubleshooting
busy	The collection speed is high, and Logtail is running as expected.	N/A	No action is required.
many_log_files	A large number of log files are being collected by Logtail.	Low	You can check the Logtail configuration for log files that do not need to be collected.
process_block	The process of log parsing is blocked.	Low	You can check whether a large number of logs are generated in a short time. If you use the all command for multiple times and the returned value is always process_block, you can modify the limit of CPU usage or the limit of concurrent packet sending .
send_block	The process of sending log packets is blocked.	High	You can check whether a large number of logs are generated in a short time and the network connection is stable. If you use the all command for multiple times and the returned value is always send_block, you can modify the limit of CPU usage or the limit of concurrent packet sending .

Command active

Command syntax

```
/etc/init.d/ilogtaild status active [--logstore] index
/etc/init.d/ilogtaild status active --logfile index project-name logstore-name
```

Note

- You can use the `active [--logstore] index` command to query all active Logstores. The `--logstore` parameter is optional.
- The command `active --logfile index project-name logstore-name` is used to query all active log files in the Logstore of a project.
- The active command is used to query log files. We recommend that you query active Logstores before querying active log files in the Logstores.

Examples

```
/etc/init.d/ilogtaild status active 1
sls-zc-test : release-test
sls-zc-test : release-test-ant-rpc-3
sls-zc-test : release-test-same-regex-3

/etc/init.d/ilogtaild status active --logfile 1 sls-zc-test release-test
/disk2/test/normal/access.log
```

Response

- If you run the `active --logstore index` command, the names of the active Logstores are returned in the

following format: `project-name : logstore-name` . If you run the command `active --logfile index project-name logstore-name` , the paths of active log files are returned.

- The status of the inactive Logstores or inactive log files in the query time window is not returned.

Command logstore

Command syntax

```
/etc/init.d/ilogtaild status logstore [--format={line|json}] index project-name logstore-name
```

Note

- The logstore command is used to query the collection status of the specified project and Logstore in the LINE or JSON format.
- The default value of the `--format=` parameter is `--format=line` , which indicates that the status is returned in the LINE format. Noted that the `--format=` parameter is placed after the `logstore` parameter.
- If the Logstore specified in the preceding command does not exist or is not active in the query time window, an empty response in LINE format or the `null` value in the JSON format is returned.

Examples

```
/etc/init.d/ilogtaild status logstore 1 sls-zc-test release-test-same
time_begin_readable : 17-08-29 10:56:11
time_end_readable : 17-08-29 11:06:11
time_begin : 1503975371
time_end : 1503975971
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503975970
read_count : 687
avg_delay_bytes : 0
max_unsend_time : 0
min_unsend_time : 0
max_send_success_time : 1503975968
send_queue_size : 0
send_network_error_count : 0
send_network_quota_count : 0
send_network_discard_count : 0
send_success_count : 302
send_block_flag : false
sender_valid_flag : true
/etc/init.d/ilogtaild status logstore --format=json 1 sls-zc-test release-test-same
```

```
{
  "avg_delay_bytes" : 0,
  "config" : "##1.0##sls-zc-test$same",
  "last_read_time" : 1503975970,
  "logstore" : "release-test-same",
  "max_send_success_time" : 1503975968,
  "max_unsend_time" : 0,
  "min_unsend_time" : 0,
  "parse_fail_lines" : 0,
  "parse_success_lines" : 230615,
  "project" : "sls-zc-test",
  "read_bytes" : 65033430,
  "read_count" : 687,
  "send_block_flag" : false,
  "send_network_discard_count" : 0,
  "send_network_error_count" : 0,
  "send_network_quota_count" : 0,
  "send_queue_size" : 0,
  "send_success_count" : 302,
  "sender_valid_flag" : true,
  "status" : "ok",
  "time_begin" : 1503975371,
  "time_begin_readable" : "17-08-29 10:56:11",
  "time_end" : 1503975971,
  "time_end_readable" : "17-08-29 11:06:11"
}
```

Response

Parameter	Description	Unit
status	The status of the Logstore. For information about Logstore statuses and actions that are required to deal with each status, see the following table.	N/A
time_begin_readable	The time when logs become readable.	N/A
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	Unix timestamp in seconds
time_end	The time when statistics collection ends.	Unix timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A

Parameter	Description	Unit
config	The name of the Logtail configuration, which is globally unique. The format of the name is <code>##1.0## + project + \$ + config</code> .	N/A
read_bytes	The amount of the log data that is read in the query time window.	Byte
parse_success_lines	The number of the log lines that are parsed in the query time window.	Line
parse_fail_lines	The number of the log lines that fail to be parsed in the query time window.	Line
last_read_time	The last time when logs are read in the query time window.	Unix timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	Times
avg_delay_bytes	The average of difference between the actual file size and the offset generated when reading log data each time in the query time window.	Byte
max_unsend_time	The maximum period of time for which an unsend packet waits in the sending queue. An unsend packet refers to a packet that has not been sent at the end of the query time window. If no packets exist in the queue, the value is 0.	Unix timestamp in seconds
min_unsend_time	The minimum period of time for which an unsend packet waits in the sending queue. Unsend packets refer to packets that have not been sent at the end of the query time window. If no packets exist in the queue, the value is 0.	Unix timestamp in seconds
max_send_success_time	The maximum period of time when a packet waited in the sending queue.	Unix timestamp in seconds
send_queue_size	The number of the unsend packets in the sending queue at the end of the query time window.	Number of packets
send_network_error_count	The number of the packets that cannot be sent due to network errors in the query time window.	Number of packets
send_network_quota_count	The number of the packets that cannot be sent due to quota limit in the query time window.	Number of packets
send_network_discard_count	The number of the packets that are discarded due to data errors or lack of permissions.	Number of packets

Parameter	Description	Unit
send_success_count	The number of the packets that are sent in the query time window.	Number of packets
send_block_flag	Indicates whether the sending queue is blocked at the end of the query time window.	N/A
sender_valid_flag	Indicates whether the sender flag of the Logstore is valid. The value true indicates that the sender flag is valid. The value false indicates that the sender flag is invalid and disabled because of a network error or quota error.	N/A

Logstore statuses

Status	Description	Troubleshooting
ok	Logtail is running as expected.	No action is required.
process_block	The process of log parsing is blocked.	You can check whether a large number of logs are generated in a short time. If you use the all command for multiple times and the returned value is always process_block, you can Set Logtail startup parameters modify the limit of CPU usage or of concurrent packet sending.
parse_fail	Logtail fails to parse logs.	You can check whether the format of logs is consistent with that you set in the Logtail configuration.
send_block	The process of sending log packets is blocked.	You can check whether a large number of logs are generated in a short time and the network connection is stable. If you use the all command for multiple times and the returned value is always send_block, you can Set Logtail startup parameters modify the limit of CPU usage or of concurrent packet sending.

Command logfile

Command syntax

```
/etc/init.d/ilogtaild status logfile [--format={line|json}] index project-name logstore-name fileFullPath
```

 Note

- The logfile command is used to query the collection status of the specified log files in the LINE or JSON format.
- The default value of the `--format=` parameter is `--format=line`, which indicates that the status is returned in the LINE format.
- If the log file specified in the command does not exist or is not active in the query time window, an empty response in the LINE format or the `null` value in the JSON format is returned.
- The `--format` parameter is placed after the `logfile` parameter.
- The value of the `filefullpath` parameter must be the full path of the log file.

Examples

```
/etc/init.d/ilogtailed status logfile 1 sls-zc-test release-test-same /disk2/test/normal/access.log
time_begin_readable : 17-08-29 11:16:11
time_end_readable : 17-08-29 11:26:11
time_begin : 1503976571
time_end : 1503977171
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
file_path : /disk2/test/normal/access.log
file_dev : 64800
file_inode : 22544456
file_size_bytes : 17154060
file_offset_bytes : 17154060
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503977170
read_count : 667
avg_delay_bytes : 0
/etc/init.d/ilogtailed status logfile --format=json 1 sls-zc-test release-test-same /disk2/test/normal/access.log
{
  "avg_delay_bytes" : 0,
  "config" : "##1.0##sls-zc-test$same",
  "file_dev" : 64800,
  "file_inode" : 22544456,
  "file_path" : "/disk2/test/normal/access.log",
  "file_size_bytes" : 17154060,
  "last_read_time" : 1503977170,
  "logstore" : "release-test-same",
  "parse_fail_lines" : 0,
  "parse_success_lines" : 230615,
  "project" : "sls-zc-test",
  "read_bytes" : 65033430,
  "read_count" : 667,
  "read_offset_bytes" : 17154060,
  "status" : "ok",
  "time_begin" : 1503976571,
  "time_begin_readable" : "17-08-29 11:16:11",
  "time_end" : 1503977171,
  "time_end_readable" : "17-08-29 11:26:11"
}
```

Response

Parameter	Description	Unit
status	The collection status of the log file in the query time window. For more information, see the status parameter in the Command logstore section.	N/A
time_begin_readable	The time when logs become readable.	N/A
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	Unix timestamp in seconds
time_end	The time when statistics collection ends.	Unix timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A
file_path	The path of the log file.	N/A
file_dev	The ID of the device from which the log file is collected.	N/A
file_inode	The inode of the log file.	N/A
file_size_bytes	The size of the log file that is last scanned in the query time window.	Byte
read_offset_bytes	The parsing offset of the log file.	Byte
config	The name of the Logtail configuration, which is globally unique. The format of the name is <code>##1.0## + project + \$ + config</code> .	N/A
read_bytes	The amount of the log data that is read in the query time window.	Byte
parse_success_lines	The number of the log lines that are parsed in the query time window.	Line
parse_fail_lines	The number of the log lines that fail to be parsed in the query time window.	Line
last_read_time	The last time when logs are read in the query time window.	Unix timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	Times
avg_delay_bytes	The average of difference between the actual file size and the offset generated when reading log data each time in the query time window.	Byte

Command history

Command syntax

```
/etc/init.d/ilogtaild status history beginIndex endIndex project-name logstore-name [fileFullPath]
```

Note

- The history command is used to query the collection status of a Logstore or log file in the query time window.
- The `beginIndex` and `endIndex` parameters specify the start and end indexes of the range of time windows that you want to query. You must ensure that `beginIndex <= endIndex`.
- The `fileFullPath` parameter is optional. If you specify the path of a log file, the collection status of the log file is queried. If the path is not specified, the collection status of the Logstore is queried.

Examples

```
/etc/init.d/ilogtaild status history 1 3 sls-zc-test release-test-same /disk2/test/normal/access.log
  begin_time      status   read parse_success parse_fail  last_read_time read_count avg_delay device  ino
de file_size read_offset
17-08-29 11:26:11    ok 62.12MB   231000    0 17-08-29 11:36:11    671    0B 64800 22544459 18.22
MB 18.22MB
17-08-29 11:16:11    ok 62.02MB   230615    0 17-08-29 11:26:10    667    0B 64800 22544456 16.36
MB 16.36MB
17-08-29 11:06:11    ok 62.12MB   231000    0 17-08-29 11:16:11    687    0B 64800 22544452 14.46
MB 14.46MB
$/etc/init.d/ilogtaild status history 2 5 sls-zc-test release-test-same
  begin_time      status   read parse_success parse_fail  last_read_time read_count avg_delay send_queue
network_error quota_error discard_error send_success send_block send_valid      max_unsend      min_unsend
max_send_success
17-08-29 11:16:11    ok 62.02MB   230615    0 17-08-29 11:26:10    667    0B    0    0    0
0    300   false   true 70-01-01 08:00:00 70-01-01 08:00:00 17-08-29 11:26:08
17-08-29 11:06:11    ok 62.12MB   231000    0 17-08-29 11:16:11    687    0B    0    0    0
0    303   false   true 70-01-01 08:00:00 70-01-01 08:00:00 17-08-29 11:16:10
17-08-29 10:56:11    ok 62.02MB   230615    0 17-08-29 11:06:10    687    0B    0    0    0
0    302   false   true 70-01-01 08:00:00 70-01-01 08:00:00 17-08-29 11:06:08
17-08-29 10:46:11    ok 62.12MB   231000    0 17-08-29 10:56:11    692    0B    0    0    0
0    302   false   true 70-01-01 08:00:00 70-01-01 08:00:00 17-08-29 10:56:10
```

Response

- The collection status of the Logstore or log file in each query time window is listed in a line.
- For more information about response parameters, see the Command `logstore` and Command `logfile` sections.

Response status codes

Success code

If parameters that you specify in a command is valid (even if the queried Logstore or log file is not found), the code 0 is returned. The following section provides two examples:

```
/etc/init.d/ilogtaild status logfile --format=json 1 error-project error-logstore /no/this/file
null
echo $?
0
/etc/init.d/ilogtaild status all
ok
echo $?
0
```

Error codes

If a non-zero code is returned, it indicates that an error occurs. The following table describes the possible non-zero codes.

Code	Description	Message	Troubleshooting
10	The command is invalid or required parameters in the command are not specified.	invalid param, use -h for help.	You can run the <code>-h</code> command for help.
1	The value of the index parameter is not in the range from 1 to 60.	invalid query interval	You can run the <code>-h</code> command for help.
1	The collection status in the specified query time window cannot be queried.	query fail, error: <code>\$(error)</code> . For more information, visit errno .	The startup time of Logtail is earlier than the query time window. Otherwise, you can submit a ticket for help.
1	The start time of querying is out of the query time window.	no match time interval, please check logtail status	You can check whether Logtail is running. If yes, you can submit a ticket for help.
1	No logs exist in the specified query time window.	invalid profile, maybe logtail restart	You can check whether Logtail is running. If yes, you can submit a ticket for help.

Examples

```
/etc/init.d/ilogtaild status nothiscmd
invalid param, use -h for help.
echo $?
10
/etc/init.d/ilogtaild status/all 99
invalid query interval
echo $?
1
```

Scenarios

You can query the overall status of Logtail, and specific metrics by querying the collection status during collection. You can customize a mechanism to monitor the log collection status based on the queried information.

Monitor the status of Logtail

You can monitor the status of Logtail by using the `all` command.

For example, you can run the command every minute to query Logtail status. If the `process_block`, `send_block` or `send_error` value is returned for 5 consecutive minutes, an alert is triggered.

You can adjust the alert duration and monitoring scope based on the priorities of the collected log files.

Monitor the log collection status

You can monitor the log collection status of a Logstore by using the `logstore` command.

For example, you can run the `logstore` command every 10 minutes to query the collection status of the Logstore. If the value of the `avg_delay_bytes` parameter exceeds 1 MB (1024 × 1024 bytes) or the value of the `status` parameter is not `ok`, an alert is triggered.

You can adjust the alert threshold for the `avg_delay_bytes` metric based on the size of data that is generated during the log collection.

Check whether Logtail has finished collecting log files

You can check whether Logtail has finished collecting log files by using the `logfile` command.

After Logtail stops collecting log files, you can run the `logfile` command every 10 minutes to query the status of the log file. If the value of the `read_offset_bytes` parameter is the same as that of the `file_size_bytes` parameter, it indicates that the log file is collected.

Troubleshoot log collection issues

If log collection latency occurs on a server, you can use the `history` command to query the status history of log collection.

1. The value of the `send_block_flag` parameter is true. This indicates that the log collection is blocked because of unstable network connections.
 - If the value of the `send_network_quota_count` parameter is greater than 0, split shards in the Logstore. For more information, see [Split a shard](#).
 - If the value of the `send_network_error_count` parameter is greater than 0, check the network connections.
 - If no network error occurs, adjust the [limit of concurrent packet sending and data transfer speed](#) of Logtail.
2. The parameters related to packet sending are set to appropriate values. However, the value of the `avg_delay_bytes` parameter is large.
 - Use the value of the `read_bytes` parameter to calculate the average speed of parsing logs, and then determine whether a large amount of data is transferred during log collection based on the average speed.
 - Adjust the [resource usage limits](#) for Logtail.
3. The value of the `parse_fail_lines` parameter is greater than 0.

Check whether the regular expression can match all required log fields as expected.

30.8.1.4. How do I test a regular expression?

If you select the full regex mode when you configure Logtail to collect and parse text logs, you must specify a regular expression based on your sample log entries. This topic describes how to test a regular expression.

Context

To test a regular expression that you have specified in the Log Service console, you can click **Validate** in the console and check the results as follows:

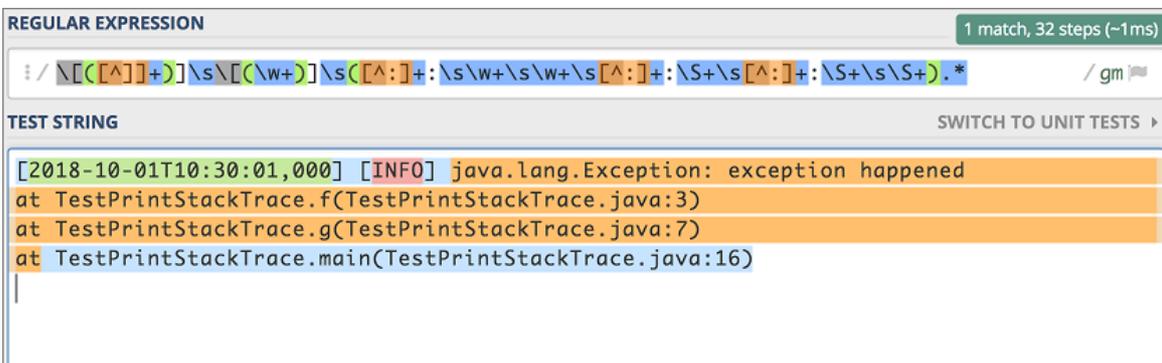
- For the regular expression that matches the first line of logs, check whether the regular expression can match the expected number of log entries.
- For the fields extracted by the regular expression, check whether the value of each field meets your expectations.

If you want to validate more items and test a regular expression, you can use online tools such as regex101.com and regextester.com. You can copy and paste the regular expression that is generated by Log Service to an online tool, and specify a sample log entry as the test string.

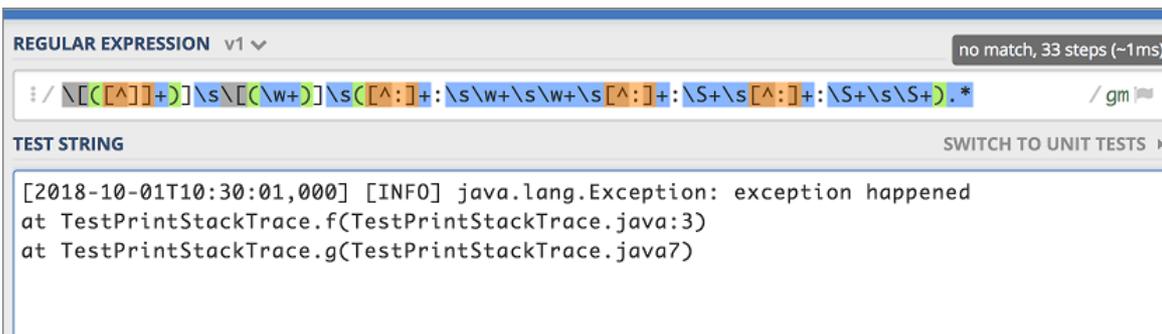
If you use the full regex mode, Log Service automatically generates a regular expression based on a sample log entry. However, the regular expression may fail to match the message field in multi-line log entries as expected. The following example describes how to use the regex101.com tool to test the regular expression.

Procedure

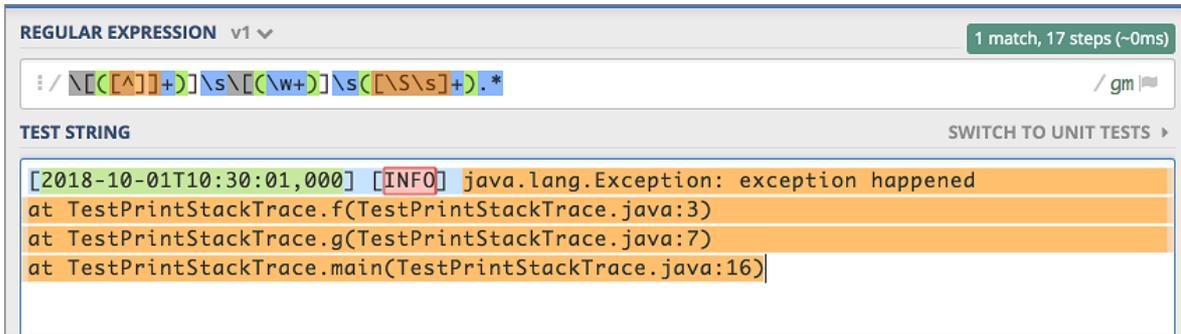
1. Copy the generated regular expression.
2. Visit the regex101.com website.
3. Paste the regular expression in the **REGULAR EXPRESSION** field. On the right side of the page, you can view the explanation of the regular expression.
4. In the **TEST STRING** field, paste a log sample entry. In the following figure, the log contents that are included in the message field are highlighted in orange, and the log contents that are not included are highlighted in blue. The figure shows that the substring following the `at` word is not included in the `message` field. Therefore, this regular expression does not match fields in the sample log entry as expected and cannot be used to collect log data.



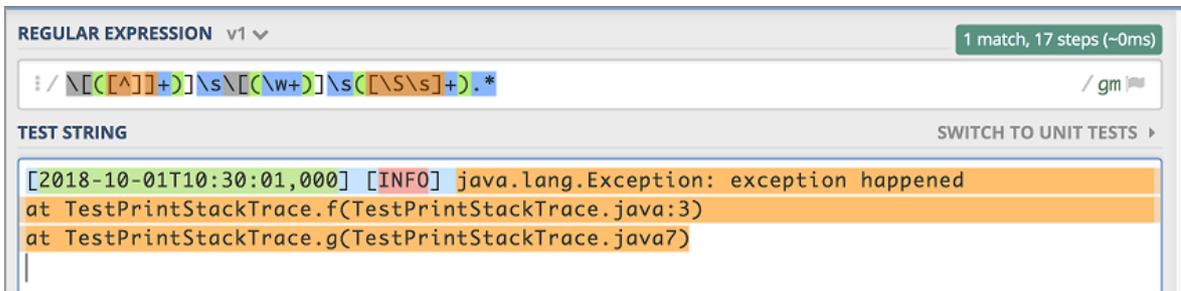
5. Check whether the regular expression can match fields in a sample log entry with two colons as expected. The following figure shows that the regular expression fails to match fields in the sample log entry as expected.



- Replace the last subexpression in the regular expression with `[\S\s]+`, and check whether the regular expression can match fields in the sample log entries as expected. The following figure shows how the modified regular expression matches the substring following the `at` word.



The following figure shows how the modified regular expression matches the sample log entry with two colons.



You can follow the preceding instructions to test your regular expression. After you validate the regular expression, you can apply it to a Logtail configuration.

30.8.1.5. How do I optimize regular expressions?

You can optimize regular expressions to improve the Logtail performance.

When you optimize regular expressions, we recommend that you follow these rules:

- Use precise characters

We recommend that you do not use the wildcard characters `.*` in a regular expression to match fields in log entries. Using wildcard characters may lead to mismatches and low matching performance. For example, if a field that you want to match only consists of letters, use `[A-Za-z]`.
- Use appropriate quantifiers

We recommend that you do not use plus signs (+) or asterisks (*). For example, you can use `\d` instead of `\d+` or `\d{1,3}` to match the IP address.
- Test and modify regular expressions

You can visit the regex101.com website to test and modify a regular expression to decrease the time required to match log entries.

30.8.1.6. How do I use the full regex mode to collect log entries in multiple formats?

The full regex mode requires that log entries to be collected be in the same format. Therefore, if you want to collect log entries that are in multiple formats, you must use the schema-on-write or schema-on-read solution.

Taking Java logs as an example, the following section lists the types of error log entry and normal log entry.

- Multi-line WARNING log entries
- Simple text INFO log entries
- Key-value DEBUG log entries

```
[2018-10-01T10:30:31,000] [WARNING] java.lang.Exception: another exception happened
  at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
  at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
  at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
[2018-10-01T10:30:32,000] [INFO] info something
[2018-10-01T10:30:33,000] [DEBUG] key:value key2:value2
```

To collect log entries of these types, you can use the following solutions:

- Schema-on-write: To extract log fields, you must apply multiple Logtail configurations with different regular expressions to a log file.

 **Note** However, Logtail cannot apply multiple Logtail configurations directly to the same log file. Therefore, you must set up multiple symbolic links for the directory in which the log file resides. Each Logtail configuration applies to a symbolic link to collect log entries in a specific format.

- Schema-on-read: you can use a common regular expression to collect log entries in different formats.

For example, if you want to collect log entries in multiple formats, you can configure a regular expression that matches the time and log level fields as the first line, and specify the rest of the log entries as the log message. If you want to parse the message, create an index for the message, specify a regular expression to extract log messages, and then extract target fields.

 **Note** We recommend that you use this solution only for scenarios in which tens of millions of log entries are collected, or fewer.

30.8.1.7. How do I set the time format for logs?

You must be familiar with the following rules before setting the time format for logs in Logtail configurations.

- The unit of the timestamp in Log Service is seconds. Therefore, you cannot set the unit as milliseconds or microseconds.
- You only need to set the time field. Other parameters are not required.

The following section lists commonly used formats:

* Log Sample: 12345""67890

The sample log entry is different from the original entry. [Modify Sample Log Entry](#)

7. Delete the quotation marks between the digits 5 and 6. Then, a non-printable character is configured in a sample log entry.

* Log Sample: 1234567890

The sample log entry is different from the original entry. [Modify Sample Log Entry](#)

30.8.1.9. How do I troubleshoot errors during container log collection?

Perform the steps that are described in this topic to troubleshoot an error that occurs when you use Logtail to collect logs from common containers or containers in a Kubernetes cluster.

Related O&M operations

- [Log on to a Logtail container](#)
- [View the operational logs of Logtail](#)
- [Ignore the stdout logs of a Logtail container](#)
- [View the status of Log Service components in a Kubernetes cluster](#)
- [View the version number, IP address, and startup time of Logtail](#)

Troubleshoot an error if Log Service does not receive heartbeats from Logtail clients

Perform the following steps to check whether Logtail is installed:

1. In the machine group, count the number of the servers whose heartbeat status is OK.
 - i. [Log on to the Log Service console](#).
 - ii. Click the project to which the machine group belongs.
 - iii. In the left-side navigation pane, click **Machine Groups**.
 - iv. In the **Machine Groups** pane, click the name of the machine group. In the **Machine Group Status** section, count the number of the servers whose heartbeat status is OK.
2. Count the number of worker nodes in the cluster. Run the `kubectl get node | grep -v master` command to query the work nodes in the cluster. Count the number of the work nodes that are returned.

```
$kubectl get node | grep -v master
NAME                                STATUS ROLES   AGE   VERSION
cn-hangzhou.i-bp17enxc2us3624wexh2 Ready <none> 238d  v1.10.4
cn-hangzhou.i-bp1ad2b02jtqd1shi2ut Ready <none> 220d  v1.10.4
```

3. Check whether the number of servers whose heartbeat status is OK in the machine group is equal to the number of worker nodes in the cluster. Troubleshoot the error based on the check result.
 - The number of servers whose heartbeat status is OK is equal to the number of worker nodes. This means that the heartbeat status of all of the servers in the machine group is **Failed**.

- If **Logtail is installed into a common container**, check whether the values of the `your_region_name`, `your_aliyun_user_id`, and `your_machine_group_user_defined_id` parameters are correct. For information about how to set these parameters, see [Collect standard Docker logs](#).
 - If **Logtail is installed into a container in a Container Service for Kubernetes cluster**, submit a ticket.
 - If **Logtail is installed into a container in a user-created Kubernetes cluster**, check whether the values of the `your-project-suffix`, `regionId`, `aliuid`, `access-key-id`, and `access-key-secret` parameters are correct. If the value of a parameter is incorrect, run the `helm del --purge alibaba-log-controller` command to delete the installation package and re-install Logtail. For information about how to set these parameters, see [Collect Kubernetes logs](#).
- The number of servers whose heartbeat status is OK is less than the number of worker nodes.
 - a. Check whether you used a YAML file to manually deploy a DaemonSet.

Run the `kubectl get po -n kube-system -l k8s-app=logtail` command to perform the check. If the command returns pod information, it indicates that you manually deployed a DaemonSet by using a YAML file.
 - b. Download the latest version of the [Logtail DaemonSet template](#).
 - c. Set the `your_region_name`, `your_aliyun_user_id`, and `your_machine_group_name` parameters to the values that are specific to your environment.
 - d. Run the `kubectl apply -f ./logtail-daemonset.yaml` command to update the DaemonSet YAML file.

Submit a ticket if the error persists.

Troubleshoot an error if Log Service collects no logs from containers

If no log is displayed in the **Consumption Preview** pane or on the **Search & Analysis** page of a Logstore, it indicates that Log Service does not collect logs from the machine group of the Logstore. Check the status of the containers that correspond to the servers in the machine group. If the containers are working as expected, perform the following steps to troubleshoot the error:

1. [Check the heartbeat status of the servers in the machine group](#).
2. Check whether the parameter settings in the Logtail configuration files are correct.

Check whether the values of the `IncludeLabel`, `ExcludeLabel`, `IncludeEnv`, and `ExcludeEnv` parameters in the Logtail collection configuration files meet your requirements.

 **Note** The `IncludeLabel` or `ExcludeLabel` parameter specifies whether to include or exclude the container images to which specified labels are attached. You can retrieve a list of container image labels by running the `docker inspect` command. The labels are not the labels that are defined by using Kubernetes. To check whether the parameter settings are correct in a Logtail configuration file, delete the `IncludeLabel`, `ExcludeLabel`, `IncludeEnv`, and `ExcludeEnv` parameters from the file. If Log Service can collect logs from the containers after the parameters are deleted, it indicates that the settings of the parameters are incorrect.

3. Check other items.

Log Service does not collect logs from containers in the following scenarios:

- Log files are not updated.
- The log files of a container are stored in locations that are neither the default storage nor a storage attached to the container.

Log on to a Logtail container

- Common container
 - i. Run the `docker ps | grep logtail` command on the host to search for the Logtail container.
 - ii. Run the `docker exec -it ***** bash` command to log on to the container.

```
$docker ps | grep logtail
223fbd3ed2a6e registry.cn-hangzhou.aliyuncs.com/log-service/logtail "/usr/local/ilogta..." 8 da
ys ago Up 8 days logtail-iba
$docker exec -it 223fbd3ed2a6e bash
```

- Container in a Kubernetes cluster

- Run the `kubectl get po -n kube-system | grep logtail` command to search for the pod where the Logtail container resides.
- Run the `kubectl exec -it -n kube-system ***** bash` command to log on to the pod.

```
$kubectl get po -n kube-system | grep logtail
logtail-ds-g5wgd          1/1   Running   0    8d
logtail-ds-slpn8         1/1   Running   0    8d
$kubectl exec -it -n kube-system logtail-ds-g5wgd bash
```

View the operational logs of Logtail

The operational logs of Logtail are saved in the files named *ilogtail.LOG* and *logtail_plugin.LOG* under the `/usr/local/ilogtail/` directory of a Logtail container.

1. Log on to a Logtail container.
2. Open the `/usr/local/ilogtail/` directory.

```
cd /usr/local/ilogtail
```

3. View the *ilogtail.LOG* and *logtail_plugin.LOG* files.

```
cat ilogtail.LOG
cat logtail_plugin.LOG
```

Ignore the stdout logs of a Logtail container

The standard output (stdout) logs of a Logtail container are useless for troubleshooting. Ignore the following stdout logs:

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500f8e2bdb95d13b1e110172ef57fe840c8215
5/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e697
18/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22d
be/merged: must be superuser to unmount
.....
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

View the status of Log Service components in a Kubernetes cluster

Run the `helm status alibaba-log-controller` command to view the status of Log Service components in a Kubernetes cluster.

View the version number, IP address, and startup time of Logtail

View the information in the `app_info.json` file under the `/usr/local/ilogtail/` directory of the Logtail container. For example, you can run the following command to view the content of the file:

```
kubect exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json
{
  "UUID" : "",
  "hostname" : "logtail-gb92k",
  "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_10.10.10.10_1517810940",
  "ip" : "10.10.10.10",
  "logtail_version" : "0.16.2",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
  "update_time" : "2018-02-05 06:09:01"
}
```

30.8.2. Log search and analysis

30.8.2.1. FAQ about log query

This topic describes the common issues that may occur when you query log data in the Log Service console. It also includes solutions to these issues.

How do I identify the source server from which Logtail collects logs during a query?

If a server group uses IP addresses as its identifier when logs are collected by using Logtail, servers in the server group are distinguished from one another by internal IP addresses. When querying logs, you can use the hostname and custom IP address to identify the source server from which logs are collected.

For example, you can use the following statement to count the times different hostnames appear in logs:

 **Note** You must enable the index feature for the Logstore and enable the statistics feature for the `__tag__:__hostname__` field in advance.

```
* | select "__tag__:__hostname__", count(1) as count group by "__tag__:__hostname__"
```

How do I query IP addresses in logs?

You can use the exact match method to query IP addresses in logs. For example, you can specify IP addresses to query log data that includes or excludes the specified IP addresses. However, you cannot use the partial match method to query log data related to specified IP addresses. This is because decimal points contained in an IP address are not default delimiters in Log Service. If you want to use the partial match method, you can configure the decimal point as a delimiter for indexes. For example, you can use the SDK to download data and then use regular expressions or the `string.indexOf` method in the code.

For example, you use the following statement to query projects that meet the specified conditions.

```
not ip:121.42.0 not status:200 not 360jk not DNSPod-Monitor not status:302 not jiankongbao
not 301 and status:403
```


If the monitoring file is written in real time, you can open the `/usr/local/ilogtail/ilogtail.LOG` file to view the error message. Common error messages are as follows:

- parse delimiter log fail: The error message is returned because an error has occurred when Log Service collects logs in the delimiter mode.
- parse regex log fail: The error message is returned because an error has occurred when Log Service collects logs in the regex mode.

Delimiter setting errors

View the configured delimiters. Check whether you can use a keyword to query log data after the log content is split by using a delimiter. For example, the default delimiters `, ; = () [] { } ? @ & < > / : ' "` are used. If a log entry contains `abc"defg,hij`, it is split into `abc"defg` and `hij`. In this case, you cannot retrieve this log entry by searching for `abc`.

Fuzzy match is also supported. For more information, see [Query syntax](#).

Note

- To save your indexing cost, Log Service has optimized the index feature. If you configure an index for a field, full-text indexing is ineffective for the key of this field. For example, an index is configured for a log field whose key is `message`, and a whitespace character is used as a delimiter. To use a whitespace character as a delimiter, put it in the middle of delimiters that you have configured for an index. You can retrieve the log entries that contain "message: this is a test message" by searching for the key-value-pair-formatted keyword `message:this`. However, if you use the keyword `this` to query the log entries, you cannot retrieve the data because an index is configured for the `message` field and full-text indexing is ineffective.
- You can create indexes or modify existing indexes. However, new or modified indexes take effect only for new data.
You can click Index Attributes to check whether the configured delimiters meet the requirements.

Other reasons

If log data is available, modify the time range of the query and try again. Log Service allows you to preview log data in real time. Due to a maximum latency of one minute, we recommend that you query log data at least one minute after logs are generated.

30.8.2.3. What are the differences between log consumption and log search and analytics?

Both the log consumption and log search and analytics features provided by Log Service need to read log data. The log consumption feature provides log collection and distribution channels. In contrast, the log search and analytics feature allows you to query log data.

Both the log consumption and log search and analytics features need to read log data:

Log collection and consumption (LogHub): provides public channels for log collection and distribution. It reads and writes full data in first-in, first-out (FIFO) order, which is similar to Kafka.

- Each Logstore has one or more shards. Data is written to a random shard.
- You can read multiple log entries at a time from a specified shard based on the order in which the log entries were written to the shard.
- You can set the start position (cursor) to pull log entries from shards based on the time when Log Service receives the log entries.

Log search and analytics: enables you to set conditions to search and analyze large amounts of log data based on LogHub.

- This feature allows you to search for required data based on query conditions.

- This feature allows you to include a combination of Boolean keywords AND, NOT, and OR and SQL statements in a query.
- This feature is independent of shards.

The following table lists the differences between the log search and analytics feature and the LogHub feature.

Feature	Log search and analytics (LogSearch)	LogHub
Search by keyword	Supported.	Not supported.
Data read (a small amount of data)	Fast.	Fast.
Data read (full data)	Slow. LogSearch reads 100 log entries in 100 milliseconds. This method is not recommended.	Fast. LogHub reads 1 MB of log data in 10 milliseconds. This method is recommended.
Data read by topic	Yes.	No. Data is identified only by shard.
Data read by shard	No. Data in all shards of a Logstore is queried.	Yes. You need to specify a shard each time to read data.
Price	Relatively high.	Low.
Scenario	Monitoring, problem investigation, and analysis.	Full data processing scenarios, such as stream computing and batch processing.

30.8.2.4. How do I resolve common errors returned in log data queries?

Common errors returned in log data queries are as follows:

line 1:44: Column 'my_key_field' cannot be resolved;please add the column in the index attribute

- Cause

The `my_key_field` key cannot be included in the query statement because it does not exist.

- Solution

In the upper-right corner of the Search & Analysis page, click Index Attributes to create an index for this field and enable the statistics feature for this field.

Column 'xxxxline' not in GROUP BY clause;please add the column in the index attribute

- Cause

You use the GROUP BY clause and include a non-aggregated field in a SELECT statement. However, this field is not specified in the GROUP BY clause. For example, the key1 field in the `select key1, avg(latency) group by key2` statement is not specified in the GROUP BY clause.

- Solution

An example correct statement is `select key1,avg(latency) group by key1,key2` .

sql query must follow search query,please read syntax doc

- Cause

You do not include a filtering condition in a query statement, for example, `select ip,count(*) group by ip` .

- Solution

An example correct statement is `*|select ip,count(*) group by ip` .

please read syntax document,and make sure all related fields are indexed. error after select .error detail:line 1:10: identifiers must not start with a digit; surround the identifier with double quotes

- Cause

The column name or variable name referenced in an SQL statement starts with a number and does not comply with the rules.

- Solution

Change the name so the name starts with a letter.

please read syntax document,and make sure all related fields are indexed. error after select .error detail:line 1:9: extraneous input " expecting

- Cause

Misspelled words exist in the query statement.

- Solution

Correct the misspelled words.

key (category) is not config as key value config,if symbol : is in your log,please wrap : with quotation mark "

- Cause

No index is configured for the category field. It cannot be used in a query statement.

- Solution

Configure an index for this field in the index attributes. For more information, see [Enable the index feature and configure indexes for a Logstore](#).

Query exceeded max memory size of 3GB

- Cause

The size of memory used by the current query exceeds 3 GB. The common cause is that a large number of values are still returned in the query result after you use the GROUP BY clause to remove duplicates.

- Solution

Reduce the number of keys specified in the GROUP BY clause.

ErrorType:ColumnNotExists.ErrorPosition,line:0,column:1.ErrorMessage:line 1:123: Column '__raw_log__' cannot be resolved; it seems __raw_log__ is wrapper by ";if __raw_log__ is a string ,not a key field, please use '__raw_log__'

- Cause

The `my_key_field` key cannot be included in the query statement because it does not exist.

- Solution

In the upper-right corner of the Search & Analysis page, click Index Attributes to create an index for this field and enable the statistics feature for this field.

30.8.2.5. Why data queries are inaccurate?

This topic describes the causes for inaccurate data queries. It also includes solutions to these issues.

When you search and analyze log data, the message **The results are inaccurate** may prompt in the console. This indicates that the returned result is inaccurate because some log data in a Logstore was not queried.

Possible causes include:

The time range for queries is excessive.

- Cause

The time range for a query is excessively wide, for example, three months or a year. In this case, Log Service cannot scan all log data generated within this time period for one query.

- Solution

Narrow down the query time range and perform multiple queries.

Query statements are complex.

- Cause

The query statement is exceedingly complex or contains multiple frequently used words. In this case, Log Service cannot scan all related log data or read the query results at one time.

- Solution

Narrow down the query scope and perform multiple queries.

The SQL computing needs to read an excessively large amount of data.

- Cause

The SQL computing needs to read an excessively large amount of data. In this case, query results are likely to become inaccurate. A maximum of 1 GB of data can be read from each shard. For example, if the SQL computing needs to read strings from multiple columns, which exceed the threshold data volume, inaccurate query results will be returned.

- Solution

Narrow down the query scope and perform multiple queries.

30.8.2.6. How do I configure indexes for historical log data?

You cannot directly configure indexes for historical log data in Log Service. To configure indexes for historical log data, you can use DataWorks or the command line interface (CLI) to move data into another Logstore.

Indexes take effect on log data that is collected after the indexes are configured. You cannot use indexes to search and analyze historical log data. To configure indexes for historical log data, you can use either of the following two methods:

- Configure indexes in a new Logstore and then use DataWorks to move data into the Logstore.

After you configure indexes in a new Logstore, you can use DataWorks to move historical log data from the Logstore where it is stored to the new Logstore. Then you can use the configured indexes to search and analyze the data.

- After you configure indexes in a new Logstore, you can use the CLI to export historical log data from the Logstore where it is stored to the new Logstore.

 **Note** The preceding two methods copy historical log data and then export the data into a Logstore. They do not change or delete the data.

30.8.3. Alarm

30.8.3.1. FAQ about alerts

This topic describes the common issues that may occur when you configure alerts in the Log Service console. It also includes solutions to these issues.

How can I include the raw error log entries in the notification content?

- Issue

More than five error log entries were generated in the past five minutes, which triggered an alert. How can I include the raw error log entries in notifications that were sent when the alert was triggered?

- Solution

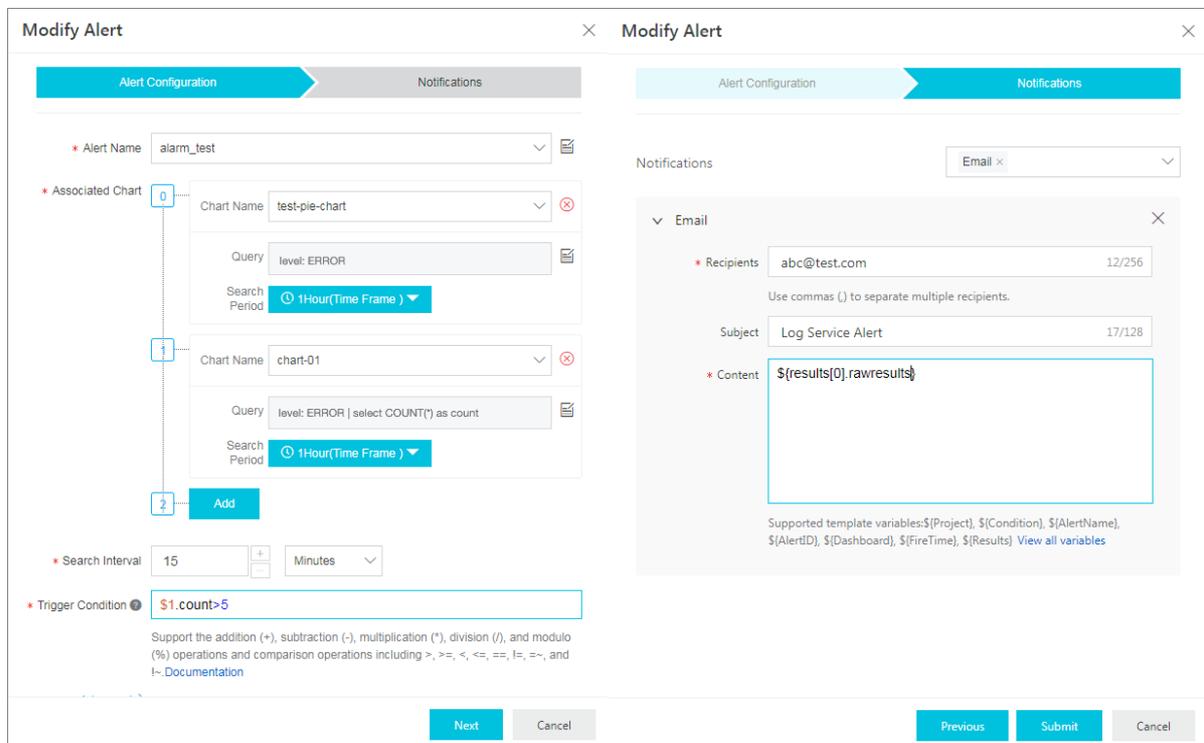
- Associated query statements

- Number 0: `level:ERROR`
- Number 1: `level:ERROR | select COUNT(*) as count`

- Trigger condition: `$1.count > 5`

- Notification content: `${results[0].rawresults}`

- Configuration examples



31. Apsara Stack DNS

31.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that runs on Apsara Stack to resolve domain names. You can configure rules to map domain names to IP addresses. Apsara Stack DNS then distributes domain name requests from clients to cloud resources, business systems on your internal networks, or the business resources of Internet service providers (ISPs).

Apsara Stack DNS provides DNS resolution in VPCs. You can perform the following operations on your VPC by using Apsara Stack DNS:

- Access other ECS instances deployed in your VPC.
- Access cloud service instances provided by Apsara Stack.
- Access custom enterprise business systems.
- Access Internet services and businesses.
- Establish network connections between DNS and user-created DNS over a leased line.
- Manage internal domain names.
- Manage DNS records of internal domain names.
- Manage forwarding configurations.
- Manage recursive resolution configurations.

31.2. User roles and permissions

Role	Permission
System administrator	A user of this role has read, write, and execute permissions on all level-1 organization resources, global resources, and system configurations.
Level-1 organization administrator	A user of this role has read, write, and execute permissions on level-1 organization resources to which the user belongs, but does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Lower-level organization administrator	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Resource user	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.
Other roles	A user of this role does not have permissions on Apsara Stack DNS. The user does not have permissions on level-1 organization resources to which the user belongs, and does not have permissions on level-1 organization resources, global resources, or system configurations to which other users belong.

31.3. Log on to the Apsara Stack DNS console

This topic describes how to log on to the Apsara Stack DNS console by using Google Chrome.

Prerequisites

- The domain name of the ASCM console is obtained from the deployment personnel before you log on to the ASCM console.
- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL used to log on to the ASCM console. Press the Enter key.
2. Enter your username and password.

Obtain the username and password used to log on to the console from the operations administrator.

 **Note** When you log on to the ASCM console for the first time, you must change the password of your username. For security reasons, your password must meet the minimum complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters.
- Digits.
- Special characters. Special characters include exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%).

3. Click **Login** to go to the ASCM console homepage.
4. In the top navigation bar, choose **Products > Networking > Apsara Stack DNS**.

31.4. Internal DNS resolution management

Internal DNS resolution management allows you to manage global internal domain names, global forwarding configurations, and global recursive resolution configurations that you have created in Apsara Stack.

31.4.1. Global internal domain names

31.4.1.1. Overview

All the operations of this feature require administrator privileges.

31.4.1.2. View an internal domain name

Procedure

1. [Log on to the Apsara Stack DNS console](#).
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click **Search**. The search result is displayed.

31.4.1.3. Add a domain name

This topic describes how to add a domain name in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, enter **Global Internal Domains**.
5. Click **OK**.

31.4.1.4. Add a description for a domain name

This topic describes how to add a description for a domain name in the Apsara Stack Cloud Management (ASCM) console.

Context

You can add a description for a domain name for easy identification. For example, you can add a hostname or internal system information to describe a domain name.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Description**.
4. In the dialog box that appears, enter a description.
5. Click **OK**.

31.4.1.5. Delete a domain name

This topic describes how to delete a domain name in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

31.4.1.6. Delete multiple domain names

This topic describes how to delete multiple domain names at a time in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Select the domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

31.4.1.7. Configure DNS records

This topic describes how to configure DNS records in the Apsara Stack Cloud Management (ASCM) console.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Configure DNS Records**.
4. In the upper-right corner of the **Configure DNS Records** page, click **Add DNS Record**.
5. Perform the following operations as needed:

- Add a description for a DNS record

Select the DNS record for which you want to add a description, click  in the Actions column, and then select **Description** from the shortcut menu. In the dialog box that appears, enter a description and click **OK**.

- Delete a DNS record

Select the DNS record that you want to delete, click  in the Actions column, and then select **Delete** from the shortcut menu. In the message that appears, click **OK**.

- Modify a DNS record

Select the DNS record that you want to modify, click  in the Actions column, and then select **Modify** from the shortcut menu. In the dialog box that appears, set the required parameters and click **OK**.

- Delete DNS records in batches

Select the DNS records that you want to delete and click **Delete** in the upper-right corner. In the message that appears, click **OK**.

31.4.1.8. View a resolution policy

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Configure DNS Records**.
4. On the page that appears, select the domain name for which you want to configure DNS records, and click **Weight** in the **Resolution Policy** column.
5. On the page that appears, view the details of **Resolution Policy**.

31.4.2. Global forwarding configurations

31.4.2.1. Global forwarding domain names

31.4.2.1.1. Overview

All operations of this feature require administrator privileges.

Apsara Stack DNS forwards specific domain names to other DNS servers for resolution.

Two forwarding modes are available: forward all requests without recursion and forward all requests with recursion.

- **Forward all requests without recursion:** Only a specified DNS server is used to resolve domain names. If the

specified DNS server cannot resolve the domain names or the request times out, a message is returned to the DNS client to indicate that the query failed.

- Forward all requests with recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, the local DNS is used instead.

31.4.2.1.2. View global forwarding domain names

This topic describes how to view forwarding domain names in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Forwarding Domains**.
3. In the **Domain Name** search box, enter the target domain name and click **Search**.

31.4.2.1.3. Add a domain name

This topic describes how to add a domain name in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Forwarding Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, configure parameters such as *Domain Name*, *Forwarding Mode*, and *Forwarder IP Addresses*. Then, click **OK**.

31.4.2.1.4. Add a description for a domain name

This topic describes how to add a description for a domain name in the ASCM console. This operation requires administrator privileges.

Context

You can add a description for a domain name for identification. For example, you can describe a domain name by using a hostname or internal system information.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Global Forwarding Settings > Global Forwarding Domains**.
3. Select the domain name for which you want to add a description, click  in the **Actions** column, and then select **Description** from the shortcut menu.
4. In the dialog box that appears, enter a description and click **OK**.

31.4.2.1.5. Modify the forwarding configurations of a domain name

This topic describes how to modify the forwarding configurations of a domain name in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Modify**.
4. In the dialog box that appears, change the value of *Forwarding Mode* or *Forwarder IP Addresses*, and click **OK**.

31.4.2.1.6. Delete a domain name

This topic describes how to delete a domain name in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Forwarding Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

31.4.2.1.7. Delete multiple domain names

This topic describes how to delete multiple domain names at a time in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Forwarding Domains**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner of the domain name list.
4. In the message that appears, click **OK**.

31.4.2.2. Global default forwarding configurations

31.4.2.2.1. Enable default forwarding

This topic describes how to enable default forwarding in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Default Forwarding**.
3. Click the  icon in the Actions column and select **Enable**.
4. In the dialog box that appears, configure *Forwarding Mode* and *Forwarder IP Addresses*. Then, click **OK**. Make sure that **Enable Default Forwarding** is set to **ON**.

31.4.2.2.2. Modify default forwarding configurations

This topic describes how to modify default forwarding configurations in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Default Forwarding**.
3. Click the  icon in the Actions column and select **Modify**.
4. In the dialog box that appears, configure *Forwarding Mode* and *Forwarder IP Addresses*. Then, click **OK**.

31.4.2.2.3. Disable default forwarding

This topic describes how to disable default forwarding in the Apsara Stack Cloud Management (ASCM) console. This operation requires administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Internal Domains > Global Forwarding Settings > Global Default Forwarding**.
3. Click the  icon in the Actions column and select **Disable**.
4. In the message that appears, click **OK**.

31.4.3. Global recursive resolution

31.4.3.1. Enable global recursive resolution

Prerequisites

You have administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. Choose **Internal Domains > Global Recursive Resolution**.
3. Click the  icon in the Actions column and select **Enable**.
4. In the dialog box that appears, click **OK**.

31.4.3.2. Disable global recursive resolution

Prerequisites

You have administrator permissions.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. Choose **Internal Domains > Global Recursive Resolution**.
3. Click the  icon in the Actions column and select **Disable**.

4. In the dialog box that appears, click OK.

31.5. PrivateZone (DNS Standard Edition only)

The PrivateZone feature allows you to create VPC-specific tenant domain names. You can bind the domain names to VPCs as required to achieve tenant isolation.

31.5.1. Tenant internal domain name

31.5.1.1. View a domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains.**
3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click **Search.** The search result is displayed.

31.5.1.2. Add a domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains.**
3. Click **Add Domain Name.**
4. In the dialog box that appears, set *Tenant Internal Domain Name.*
5. Click **OK.**

31.5.1.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. To ensure that the DNS forwarding configurations take effect, you must bind the organization of domain names to a VPC.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains.**
3. Find the target domain name, click the  icon in the **Actions** column, and select **Associate VPCs.**
4. Select one or more VPCs from the list of VPCs to **Select**, click the right arrow to add them to the list of VPCs **Selected**, and then click **OK.**

31.5.1.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains.**
3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the **VPCs Associated** page, find the target VPC, click the  icon in the **Actions** column, and then select **Disassociate.** Make sure that the unbound VPC is no longer displayed on the **VPCs Associated** page.

31.5.1.5. Add a description for a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose Forwarding Configurations > Tenant Internal Domains.
3. Find the target domain name, click the  icon in the Actions column, and then select Description.
4. In the dialog box that appears, enter a description.
5. Click OK.

31.5.1.6. Delete a domain name

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose Forwarding Configurations > Tenant Internal Domains.
3. Find the target domain name, click the  icon in the Actions column, and then select Delete.
4. In the message that appears, click OK.

31.5.1.7. Delete multiple domain names

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose Forwarding Configurations > Tenant Internal Domains.
3. Select one or more domain names that you want to delete and click Delete in the upper-right corner.
4. In the message that appears, click OK.

31.5.1.8. Configure DNS records

Procedure

1. Log on to the Apsara Stack DNS console.
2. In the left-side navigation pane, choose Forwarding Configurations > Tenant Internal Domains.
3. Find the target domain name, click the  icon in the Actions column, and then select Configure DNS Records.
4. In the upper-right corner of the Configure DNS Records page, click Add DNS Record.
5. In the Add DNS Record dialog box, configure *Host*, *Type*, *TTL*, *Resolution Policy*, and *Record Set*. Then, click OK. The following tables describe the types of DNS records.
 - o A record

Resolution policy	Formatting rule
-------------------	-----------------

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique IPv4 addresses, each in a separate row.</p> <p>Make sure that the IPv4 addresses are valid.</p> <p>Example:</p> <ul style="list-style-type: none"> ▪ 192.168.1.1 ▪ 192.168.1.2 ▪ 192.168.1.3
Weight	<p>You can enter up to 100 unique IPv4 addresses, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [IPv4 address] [Weight] (The IPv4 address and weight are separated with a space.) ▪ Make sure that the IPv4 addresses are valid. ▪ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. <p>Example:</p> <ul style="list-style-type: none"> ▪ 192.168.1.1 20 ▪ 192.168.1.1 30 ▪ 192.168.1.1 50

○ AAAA record

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique IPv6 addresses, each in a separate row.</p> <p>Make sure that the IPv6 addresses are valid.</p> <p>Example:</p> <ul style="list-style-type: none"> ▪ 2400:3200::6666 ▪ 2400:3200::6688 ▪ 2400:3200::8888
Weight	<p>You can enter up to 100 unique IPv6 addresses, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [IPv6 address] [Weight] (The IPv6 address and weight are separated with a space.) ▪ Make sure that the IPv6 addresses are valid. ▪ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. <p>Example:</p> <ul style="list-style-type: none"> ▪ 2400:3200::6666 20 ▪ 2400:3200::6688 20 ▪ 2400:3200::8888 60

○ CNAME record

Resolution policy	Formatting rule
None	<p>You can enter only one domain name.</p> <p>The domain name must be a fully qualified domain name (FQDN) that ends with a dot (.). It must be 1 to 255 characters in length.</p> <p>Example: www.example.com.</p>
Weight	<p>You can enter up to 100 unique domain names, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [Domain name] [Weight] (The domain name and weight are separated with a space.) ▪ The domain name must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. ▪ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. <p>Example:</p> <ul style="list-style-type: none"> ▪ www1.example.com. 20 ▪ www2.example.com. 20 ▪ www3.example.com. 60

○ MX record

Resolution policy	Formatting rule
None	<p>You can enter 100 unique email server hostnames, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [Priority] [Email server hostname] (The priority and hostname are separated with a space.) ▪ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ▪ The email server hostname must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. <p>Example:</p> <ul style="list-style-type: none"> ▪ 10 mailserver1.example.com. ▪ 20 mailserver2.example.com.

○ TXT record

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique character strings, each in a separate row.</p> <p>A string must be 1 to 255 characters in length. No row can be left blank.</p> <p>Example: "v=spf1 ip4:192.168.0.1/16 ip6:2001::1/96 ~all"</p>

○ PTR record

Resolution policy	Formatting rule

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique domain names, each in a separate row.</p> <p>The DNS server address must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length.</p> <p>Example:</p> <ul style="list-style-type: none"> ▪ www1.example.com. ▪ www2.example.com. ▪ www3.example.com.

○ SRV record

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique application server hostnames, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [Priority] [Weight] [Port number] [Application server hostname] (Every two items are separated with a space.) ▪ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ▪ The weight value is an integer ranging from 0 to 999. A larger value indicates a greater weight. ▪ The port number is an integer ranging from 0 to 65535. It indicates the TCP or UDP port used for network communications. ▪ The application server hostname must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. <p>Example:</p> <ul style="list-style-type: none"> ▪ 1 10 8080 www1.example.com. ▪ 2 20 8081 www2.example.com.

○ NAPTR record

Resolution policy	Formatting rule
-------------------	-----------------

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique NAPTR record values, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [Serial number] [Priority] [Flag] [Service information] [Regular expression] [Substitute domain name] (Every two items are separated with a space.) ▪ The serial number is an integer ranging from 0 to 999. A smaller value indicates a higher priority. ▪ The priority value is an integer ranging from 0 to 999. A smaller value indicates a higher priority. If two records have the same serial number, the one with a higher priority takes effect first. ▪ The flag value can be left blank or be a character from A to Z, a to z, or 0 to 9. It is not case-sensitive and must be enclosed in double quotation marks (""). ▪ The service information can be left blank or be a string of 1 to 32 characters. It must start with a letter and be enclosed in double quotation marks (""). ▪ The regular expression can be left blank or be a string of 1 to 255 characters enclosed in double quotation marks (""). ▪ The substitute domain name must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. <p>Example:</p> <ul style="list-style-type: none"> ▪ 100 50 "S" "Z3950+I2L+I2C" "" "_z3950_tcp.example.com." ▪ 100 50 "S" "RCDS+I2C" "" "_rcds_udp.example.com." ▪ 100 50 "S" "HTTP+I2L+I2C+I2R" "" "_http_tcp.example.com."

○ **CAA record**

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique CAA records, each in a separate row.</p> <p>Format:</p> <ul style="list-style-type: none"> ▪ [Certificate authority flag] [Certificate property tag] [Authorization information] (Every two items are separated with a space.) ▪ The certification authority flag is an integer ranging from 0 to 255. ▪ The certificate property tag can be issue, issuewild, or iodef. ▪ The authorization information must be 1 to 255 characters in length and enclosed in double quotation marks (""). <p>Example:</p> <ul style="list-style-type: none"> ▪ 0 issue "caa.example.com" ▪ 0 issuewild ";" ▪ 0 iodef "mailto:example@example.com"

○ **NS record**

Resolution policy	Formatting rule

Resolution policy	Formatting rule
None	<p>You can enter up to 100 unique DNS server addresses, each in a separate row.</p> <p>The DNS server address must be an FQDN that ends with a period (.). It must be 1 to 255 characters in length. Wildcard domain names are not allowed.</p> <p>Example:</p> <ul style="list-style-type: none"> ▪ ns1.example.com. ▪ ns2.example.com.

6. After you add DNS records, perform the following operations as required:

- Add a description for a DNS record.

Find the target DNS record, click the  icon in the Actions column, and then select **Description**. In the dialog box that appears, enter a description and click **OK**.

- Delete a DNS record.

Find the target DNS record, click the  icon in the Actions column, and then select **Delete**. In the message that appears, click **OK**.

- Modify a DNS record.

Find the target DNS record, click the  icon in the Actions column, and then select **Modify**. In the dialog box that appears, modify the required parameters and click **OK**.

- Delete multiple DNS records.

Select the DNS records that you want to modify and click **Delete** in the upper-right corner. In the message that appears, click **OK**.

31.5.1.9. View a resolution policy

This topic describes how to view the details of a resolution policy.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Internal Domains**.
3. Find the target domain name, click the  icon in the Actions column, and then select **Configure DNS Records**.
4. View the resolution policy in the DNS Records list.

31.5.2. Tenant forwarding configurations

31.5.2.1. Tenant forwarding domain names

31.5.2.1.1. View a tenant forwarding domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.

3. In the **Domain Name** search box, enter the domain name that you want to view.
4. Click **Search**. The search result is displayed.

31.5.2.1.2. Add a tenant forwarding domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Click **Add Domain Name**.
4. In the dialog box that appears, configure parameters such as *Domain Name*, *Forwarding Mode*, and *Forwarder IP Addresses*.

Parameter	Description
Domain Name	<p>The domain name, which must meet the following formatting rules:</p> <ul style="list-style-type: none"> ◦ The domain name must be 1 to 255 characters in length. This includes the period (.) at the end of the domain name. ◦ The domain name can contain multiple domain name segments that are separated with periods (.). A domain name segment must be 1 to 63 characters in length. It cannot contain consecutive periods (.) or be left blank. ◦ The domain name can only contain letters (a to z, A to Z), digits (0 to 9), hyphens (-), and underscores (_). ◦ The domain name must start with a letter, digit, or underscore (_) and end with a letter, digit, or period (. ◦ The domain name is not case-sensitive. The system saves the domain name in lowercase letters. ◦ The period (.) at the end of the domain name is optional. The system adds a period (.) to the end of the domain name.
Forwarding Mode	<p>For both domain name-based forwarding and default forwarding, the following two forwarding modes are supported:</p> <ul style="list-style-type: none"> ◦ Forward All Requests without Recursion: forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names, a message is returned to the DNS client indicating that the query failed. ◦ Forward All Requests with Recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, the local DNS is used instead. If you enter internal IP addresses in the Forwarder IP Addresses field, unexpected results may occur during recursive resolution. For example, a domain name used for internal network services may be resolved to a public IP address.
Forwarder IP Addresses	<p>A list of destination IP addresses.</p> <div style="background-color: #e6f2ff; padding: 5px;"> <p> Note Multiple IP addresses are separated with semicolons (;).</p> </div>

5. Click **OK**.

31.5.2.1.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. You must bind the organization of domain names to a VPC before the DNS forwarding configurations can take effect.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains.**
3. Find the target domain name, click the  icon in the Actions column, and then select **Associate VPCs.**
4. Select one or more VPCs from the list of VPCs to Select, click the right arrow to add them to the list of VPCs Selected, and then click **OK.**

31.5.2.1.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains.**
3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the **VPCs Associated** page, find the target VPC, click the  icon in the Actions column, and then select **Disassociate.** Make sure that the unbound VPC is no longer displayed on the **VPCs Associated** page.

31.5.2.1.5. Modify the forwarding configurations of a domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains.**
3. Find the target domain name, click the  icon in the Actions column, and then select **Modify.**
4. In the dialog box that appears, change the value of **Forwarding Mode** or **Forwarder IP Addresses.**
5. Click **OK.**

31.5.2.1.6. Add a description for a tenant forwarding domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains.**
3. Find the target domain name, click the  icon in the Actions column, and then select **Description.**
4. In the dialog box that appears, enter a description.
5. Click **OK.**

31.5.2.1.7. Delete a tenant forwarding domain name

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings >**

Tenant Forwarding Domains.

3. Find the target domain name, click the  icon in the Actions column, and then select **Delete**.
4. In the message that appears, click **OK**.

31.5.2.1.8. Delete multiple tenant forwarding domain names

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Forwarding Domains**.
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK**.

31.5.2.2. Tenant default forwarding configurations

31.5.2.2.1. View default forwarding configurations

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.

31.5.2.2.2. Add a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Click **Add Settings**.
4. In the dialog box that appears, configure parameters such as *Forwarding Mode* and *Forwarder IP Addresses*.

Parameter	Description
-----------	-------------

Parameter	Description
Forwarding Mode	<p>For both domain name-based forwarding and default forwarding, the following two forwarding modes are available:</p> <ul style="list-style-type: none"> Forward All Requests without Recursion: Only a specified DNS server is used to resolve domain names. If the specified DNS server cannot resolve the domain names, a message is returned to the DNS client to indicate that the query failed. Forward All Requests with Recursion: A specified DNS server is preferentially used to resolve domain names. If the specified DNS server cannot resolve the domain names, a local DNS server is used instead. If you enter internal IP addresses in the Forwarder IP Addresses field, unexpected results may occur during recursive resolution. For example, a domain name used for internal network services may be resolved to a public IP address.
Forwarder IP Addresses	<p>A list of destination IP addresses.</p> <p> Note Multiple IP addresses are separated with semicolons (;).</p>

5. Click **OK**.

31.5.2.2.3. Bind an organization to a VPC

Tenant domain names are isolated based on VPCs. You must bind the organization of domain names to a VPC before the DNS forwarding configurations can take effect.

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console](#).
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target organization, click the  icon in the Actions column, and then select **Associate VPCs**.
4. Select one or more VPCs from the list of VPCs to Select, click the right arrow to add them to the list of VPCs Selected, and then click **OK**.

31.5.2.2.4. Unbind a domain name from a VPC

This topic describes how to unbind a domain name from a VPC.

Procedure

1. [Log on to the Apsara Stack DNS console](#).
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding**.
3. Find the target domain name and click the number in the **VPCs Associated** column.
4. On the VPCs Associated page, find the target VPC, click the  icon in the Actions column, and then select **Disassociate**. Make sure that the unbound VPC is no longer displayed on the VPCs Associated page.

31.5.2.2.5. Modify a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding.**
3. Find the target organization, click the  icon in the Actions column, and then select **Modify.**
4. In the dialog box that appears, change the value of **Forwarding Mode** or **Forwarder IP Addresses.**
5. Click **OK.**

31.5.2.2.6. Add a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding.**
3. Find the target organization, click the  icon in the Actions column, and then select **Description.**
4. In the dialog box that appears, enter **Description.**
5. Click **OK.**

31.5.2.2.7. Delete a default forwarding configuration

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding.**
3. Find the target organization, click the  icon in the Actions column, and then select **Delete.**
4. In the dialog box that appears, click **OK.**

31.5.2.2.8. Delete multiple default forwarding configurations

Prerequisites

You have the permissions of a system administrator or level-1 organization administrator.

Procedure

1. [Log on to the Apsara Stack DNS console.](#)
2. In the left-side navigation pane, choose **Forwarding Configurations > Tenant Forwarding Settings > Tenant Default Forwarding.**
3. Select one or more domain names that you want to delete and click **Delete** in the upper-right corner.
4. In the message that appears, click **OK.**

31.6. Internal Global Traffic Manager (internal GTM Standard Edition only)

Internal Global Traffic Manager (GTM) supports multi-cloud disaster recovery for domain names of customers. This feature manages traffic loads between multiple Apsara Stack networks.

31.6.1. Scheduling instance management

31.6.1.1. Scheduling Instance

The Scheduling Instance tab displays all existing scheduling instances. You can add, delete, modify, and configure scheduling instances on this tab. When you create a scheduling instance, you must associate an address pool and scheduling domain with the instance.

31.6.1.1.1. Create a scheduling instance

After you create a scheduling instance, you can associate the scheduling instance with a scheduling domain and address pool.

1. Log on to the Apsara Stack DNS console and choose **Recursion Configurations > Scheduling Instances > Scheduling Instance.**
2. Click **Create Scheduling Instance** in the upper-right corner of the list.
3. In the dialog box that appears, enter information of the scheduling instance and click **OK.**

31.6.1.1.2. Modify a scheduling instance

1. Log on to the Apsara Stack DNS console and choose **Recursion Configurations > Scheduling Instances > Scheduling Instance.**
2. Find the target instance and click **Modify** in the Actions column.
3. Modify the configurations of the instance and click **OK.**

31.6.1.1.3. Configure a scheduling instance

This topic describes how to add and delete a scheduling instance and modify the configurations of a scheduling instance.

31.6.1.1.3.1. Create an access policy for a scheduling instance

1. Log on to the Apsara Stack DNS console and choose **Recursion Configurations > Scheduling Instances > Scheduling Instance.** Find the target scheduling instance and click **Configure** in the Actions column. On the **Access Policy Configuration** page, click **Create Access Policy** in the upper-right corner of the page.
2. In the dialog box that appears, enter the required information and click **OK.**

31.6.1.1.3.2. Modify the access policy of a scheduling instance

1. On the **Access Policy Configuration** page, find the target access policy and click **Modify** in the Actions column.

2. In the dialog box that appears, modify the configurations and click OK.

31.6.1.1.3.3. Delete the access policy of a scheduling instance

1. On the Access Policy Configuration page, find the target access policy and click Delete in the Actions column.
2. In the dialog box that appears, click OK after you verify that the displayed information is correct.

31.6.1.1.4. Delete a scheduling instance

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Instance.
2. Find the target instance and click Delete in the Actions column.
3. In the dialog box that appears, click OK.

Note: After you delete the instance, its configuration data is also deleted.

31.6.1.2. Address Pool

The Address Pool tab allows you to manage address pools. The address pools are classified into three types: IPv4 address pool, IPv6 address pool, and CNAME address pool. You can associate a new scheduling instance with a specific address pool. You can also add weights to address pools.

31.6.1.2.1. Create an address pool

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Address Pool.
2. Click Create Address Pool in the upper-right corner of the address pool list.
3. In the dialog box that appears, enter information of the address pool and click OK.

31.6.1.2.2. Modify the configurations of an address pool

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Address Pool.
2. Find the target address pool and click Modify in the Actions column.
3. In the dialog box that appears, modify the configurations and click OK.

31.6.1.2.3. Delete an address pool

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Address Pool.
2. Find the target address pool and click Delete in the Actions column.
3. In the dialog box that appears, click OK after you verify that the displayed information is correct.

31.6.1.3. Scheduling Domain

The Scheduling Domain tab allows you to add, delete, modify, and query scheduling domains.

You can log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Domain to view the scheduling domain list.

31.6.1.3.1. Create a scheduling domain

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Domain. Then, click Create Scheduling Domain in the upper-right corner of the scheduling domain list.
2. In the dialog box that appears, enter the custom domain name and click OK.

31.6.1.3.2. Add a description for a scheduling domain

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Domain.
2. Find the target scheduling domain and click Edit in the Actions column.
3. In the dialog box that appears, add a description in the Edit field and click OK.

31.6.1.3.3. Delete a scheduling domain

1. Log on to the Apsara Stack DNS console and choose Recursion Configurations > Scheduling Instances > Scheduling Domain.
2. Find the target scheduling domain and click Delete in the Actions column.
3. In the dialog box that appears, click OK after you verify that the displayed information is correct.

31.6.2. Data synchronization management

Data synchronization management is used to synchronize Global Traffic Manager (GTM) data between clouds.

31.6.2.1. Data Synchronization Link

The Data Synchronization Link tab displays local system information, information of the cluster nodes between which data synchronization has been established, and their primary/secondary relationships.

31.6.2.1.1. Standalone node on the Data Synchronization Link tab

Services can be switched over between Global Traffic Manager (GTM) nodes. If GTM is deployed in standalone mode, you must add slave nodes because two or more nodes are required for a switchover.

1. On the Data Synchronization Link tab, click Add Slave Node in the Data Synchronization Cluster section.
2. In the dialog box that appears, enter two IP addresses of the slave node that you want to add. Enter one IP address in each row.
3. Click OK. Then, wait for the confirmation of the slave node.

31.6.2.1.2. Master node on the Data Synchronization Link tab

You can switch the master node to the slave node and add slave nodes for the master node.

1. To add a slave node, click Add Slave Node in the Data Synchronization Cluster section on the Data Synchronization Link tab.
2. To switch the master node to the slave node, click Switch to Slave.
3. In the dialog box that appears, click OK after you verify that the displayed information is correct.

31.6.2.1.3. Slave node on the Data Synchronization Link tab

You can switch the standalone node to the slave node or the master node.

31.6.2.2. Link Change Messages

The Link Change Messages tab displays the status of each node in a Global Traffic Manager (GTM) cluster, which can be To Be Processed, Accepted, or Rejected.

You can log on to the Apsara Stack DNS console and choose Recursion Configurations > Data Synchronization > Link Change Messages to view the status of each cluster node.

The Link Change Messages tab displays the node status and the link change messages of the following node types:

Standalone node

Master node

Slave node