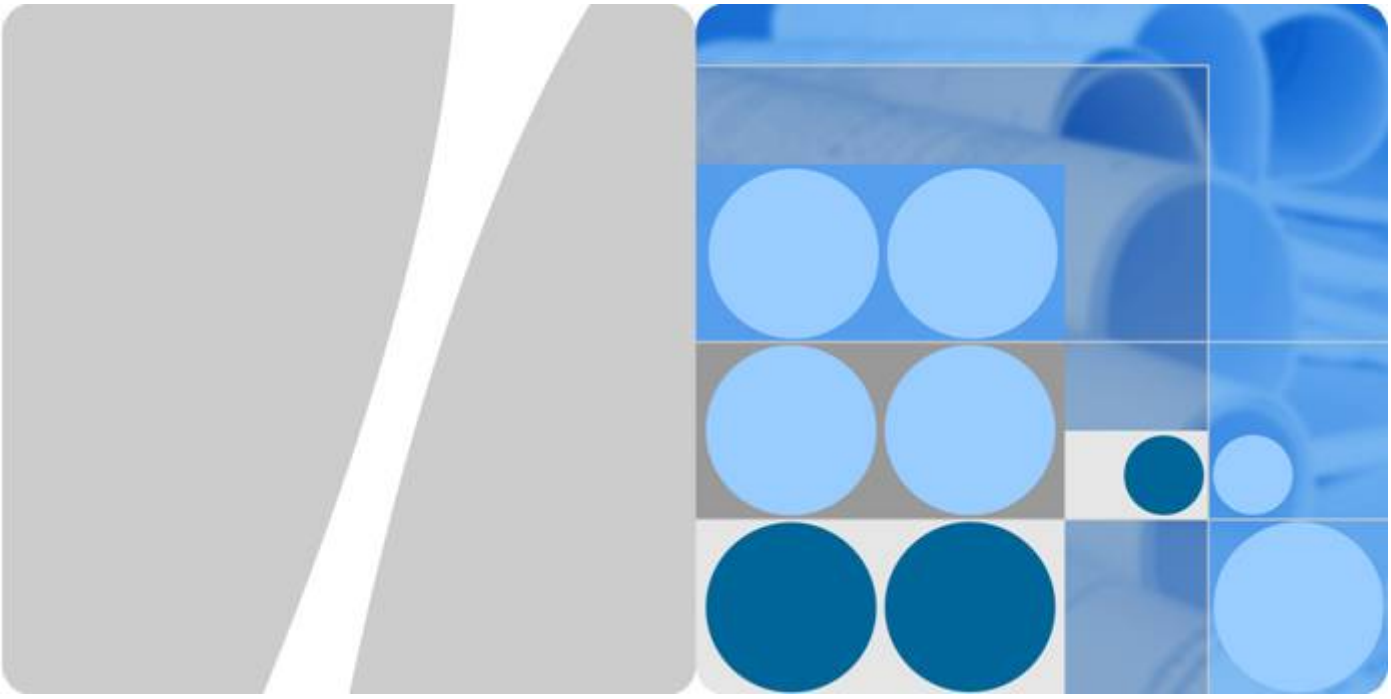


Document Database Service

User Guide

Issue 34

Date 2020-10-30

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Permissions Management

1.1 Creating a User and Granting Permissions

This section describes how to use **IAM** to implement fine-grained permissions control for your DDS resources. With IAM, you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials, providing access to DDS resources.
- Grant only the permissions required for users to perform a task.
- Entrust a HUAWEI CLOUD account or cloud service to perform professional and efficient O&M on your DDS resources.

If your HUAWEI CLOUD account does not need individual IAM users, then you may skip over this topic.

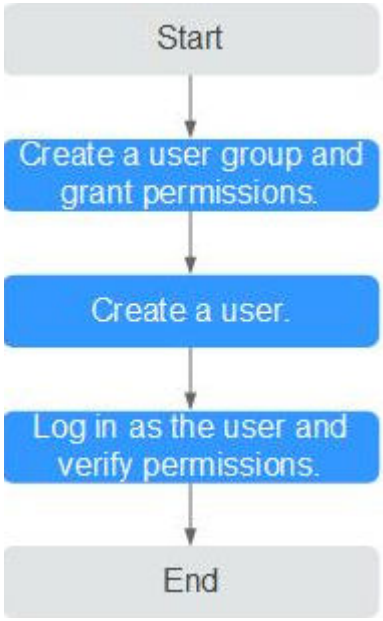
This section describes the procedure for granting permissions (see **Figure 1-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by DDS and choose policies or roles according to your requirements. For the system policies of other services, see **Permissions Policies**.

Process Flow

Figure 1-1 Process for granting DDS permissions



1. **Create a user group and assign permissions**.

   Create a user group on the IAM console, and assign the **DDS ReadOnlyAccess** policy to the group.

2. **Create an IAM user**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the DDS console by using the newly created user, and verify that the user only has read permissions for DDS.

   - Choose **Service List** > **Document Database Service** and click **Buy DB Instance**. If you cannot buy a DDS DB instance, the **DDS ReadOnlyAccess** permission has taken effect.
   - Choose any other service in the **Service List** (for example, there is only the **DDS ReadOnlyAccess** policy). If a message appears indicating insufficient permissions to access the service, the **DDS ReadOnlyAccess** policy has already taken effect.

1.2 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of DDS. For the actions supported for custom policies, see **DDS Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common DDS custom policies.

Example Custom Policies

- Example 1: Allowing users to create DDS DB instances

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "dds:instance:create"
         ]
      }
   ]
}
```

- Example 2: Denying DDS DB instance deletion

  A deny policy must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **DDS FullAccess** policy to a user but also forbid the user from deleting DDS DB instances. Create a custom policy for denying DDS DB instance deletion, and assign both policies to the group the user belongs to. Then the user can perform all operations on DDS except deleting DDS instances. The following is an example deny policy:

```
{
   "Version": "1.1",
   "Statement": [
      {
        "Effect": "Deny"
        "Action": [
            "dds:instance:deleteInstance"
         ],
        }
   ]
}
```

- Example 3: Defining permissions for multiple services in a policy

  A custom policy can contain actions of multiple services that are all of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "dds:instance:create",
                "dds:instance:modify",
                "dds:instance:deleteInstance",
             "vpc:publicIps:list",
             "vpc:publicIps:update"
            ],
            "Effect": "Allow"
        }
```



```
        ]
}
```


2 Billing Management

2.1 Renewing DB Instances

Scenarios

This section describes how to renew your yearly/monthly DB instances.

NOTE

- Pay-per-use DB instances do not support this function.
- Yearly/monthly DB instances can be renewed only when their statuses are **Available**.

Renewing DB Instances

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, select the target DB instances and click **Renew** in the upper left corner of the DB instance list. In the displayed dialog box, click **Yes**.

Step 3 On the displayed page, renew the DB instances.

----End

Renewing a DB Instance

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and click **Renew** in the **Operation** column.

Step 3 On the displayed page, renew the DB instance.

----End

2.2 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Scenarios

You can change the billing mode of a DB instance from pay-per-use to yearly/monthly to reduce your costs for using the instance for a long period of time.

NOTE

Only when the status of a pay-per-use DB instance is **Available**, its billing mode can be changed to yearly/monthly.

Changing DB Instance Billing in Batches

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, select the target DB instances and click **Change to Yearly/Monthly** above the DB instance list. In displayed dialog box, click **Yes**.

Step 3 On the displayed page, select the renewal duration in month. The minimum duration is one month.

Confirm the settings and click **Pay**.

Step 4 Select a payment method and click **OK**.

Step 5 View the results on the **Instance Management** page.

In the upper right corner of the DB instance list, click to refresh the list. After the DB instance billing mode is changed to yearly/monthly, the instance status will change to **Available**. The billing mode becomes to **Yearly/Monthly**.

----End

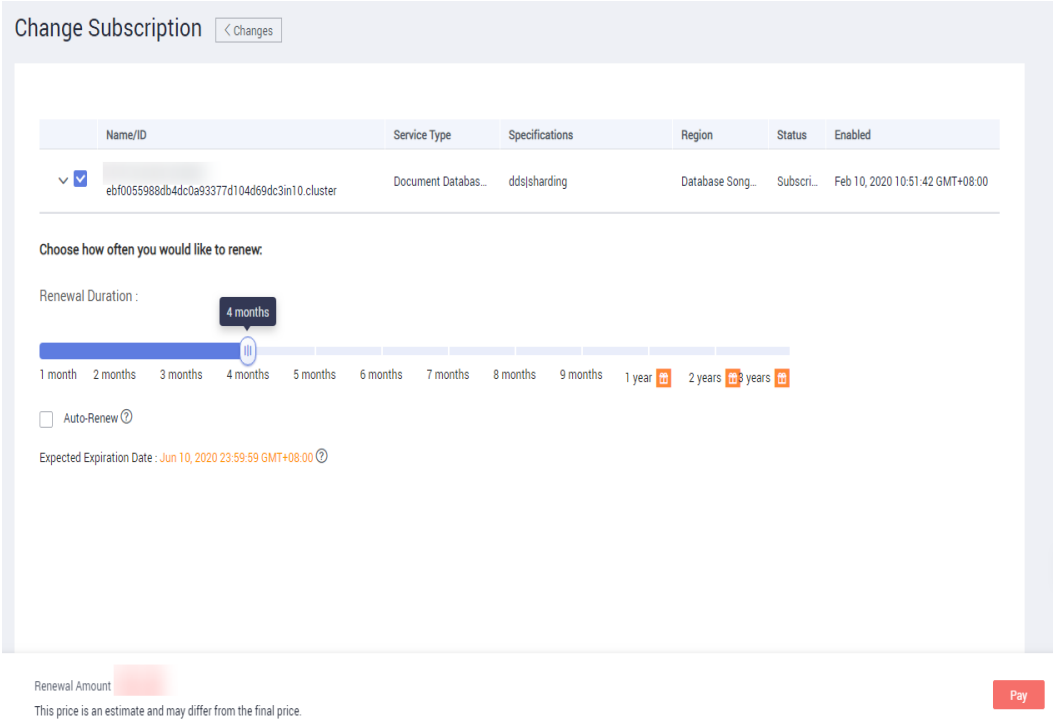
Changing the Billing Mode of a Single Instance

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and in the **Operation** column, click **Change to Yearly/Monthly**.

Step 3 On the displayed page, select the renewal duration in month. The minimum duration is one month. **Figure 2-1** shows how to change a pay-per-use cluster instance to yearly/monthly.

Figure 2-1 Changing a pay-per-use cluster instance to yearly/monthly



Confirm the settings and click **Pay**.

Step 4 Select a payment method and click **OK**.

Step 5 View the results on the **Instance Management** page.

In the upper right corner of the DB instance list, click to refresh the list. After the DB instance billing mode is changed to yearly/monthly, the instance status will change to **Available**. The billing mode becomes to **Yearly/Monthly**.

----End

2.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Scenarios

You can change yearly/monthly DB instances to pay-per-use DB instances on DDS to pay only for the actual usage of your resources.

NOTE

The billing mode can be changed only when the DB instance is in the **Available** state.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and click **Change to Pay-per-Use** in the **Operation** column.

Step 3 On the displayed page, confirm the instance information and click **Change to Pay-per-Use** to submit the change. The billing mode will change to pay-per-use after the DB instance expires.

NOTICE

Auto renewal will be disabled after the billing mode of your DB instances change to pay-per-use. Exercise caution when performing this operation.

Step 4 After you submit the change, a message is displayed in the **Billing Mode** column of the target DB instance, indicating that the billing mode will be changed to pay-per-use after the DB instance expires.

Step 5 To cancel the change, choose **Billing** > **Renewal** to enter the Billing Center. On the **Renewals** page, locate the target DB instance and click **More** > **Cancel Change to Pay-per-Use**.

Step 6 In the displayed dialog box, click **OK**.

----End

2.4 Unsubscribing from a Yearly/Monthly DB Instance

Scenarios

- To unsubscribe from a DB instance billed in the yearly/monthly mode, you need to unsubscribe from the order.

  If the DB instance is frozen, you can release the instance resource on the DDS console or in the billing center. For details about how to release resources in the billing center, refer to **Releasing Resources**.
- To unsubscribe from a DB instance billed in the pay-per-use mode, you need to locate the target DB instance and click **Delete** on the **Instance Management** page. For details, see **Deleting a Pay-per-Use DB Instance**.

Method 1

Unsubscribe from a yearly/monthly DB instance on the **Instance Management** page.

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, select target DB instances and click **Unsubscribe** above the DB instance list. Alternatively, in the **Operation** column, choose **More** > **Unsubscribe**.

Step 3 In the displayed dialog box, click **Yes**.

NOTICE

Unsubscribe operations cannot be undone. Exercise caution when performing this operation. To retain data, create a manual backup before unsubscription.

Step 4 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing from resources, see **Unsubscription Rules**.

Step 5 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 6 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance is no longer displayed in the instance list on the **Instance Management** page.

----End

Method 2

Unsubscribe from a yearly/monthly DB instance on the **Billing Center** page

Step 1 **Log in to the DDS console.**

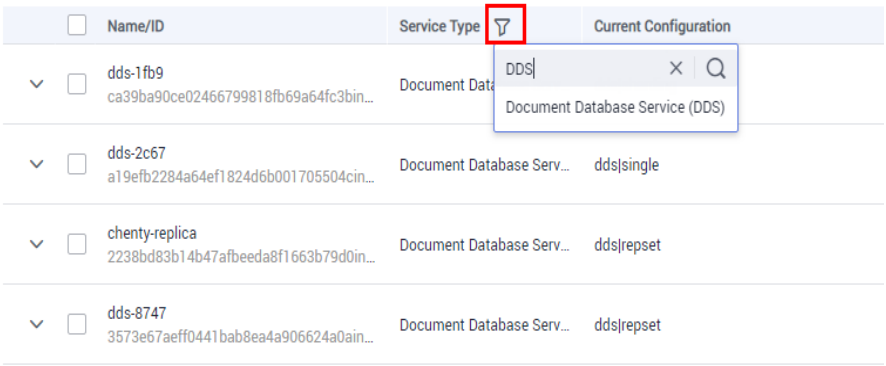
Step 2 In the upper right corner, click **Billing**.

Step 3 In the navigation pane on the left, choose **Unsubscriptions and Changes** > **Unsubscriptions**.

Step 4 On the displayed page, select the order to be unsubscribed and click **Unsubscribe** in the **Operation** column.

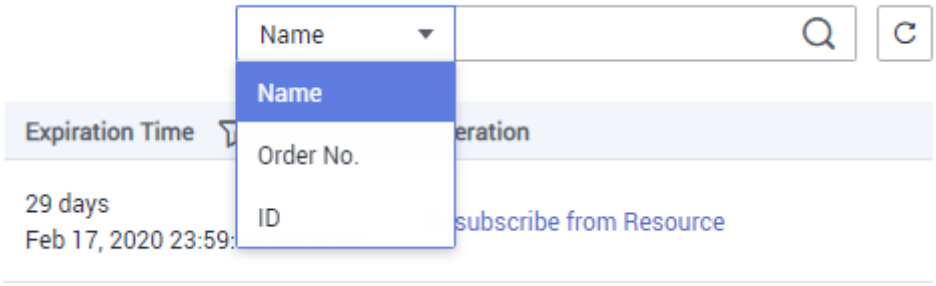
- You can select **Document Database Service** in the **Service Type** to filter all DDS orders.

Figure 2-2 Filtering all orders



- Alternatively, you can search for target orders by name, order No., or ID in the search box in the upper right corner of the order list.

  Figure 2-3 Searching for orders

- A maximum of 20 orders can be unsubscribed from at a time.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing from resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance is no longer displayed in the instance list on the **Instance Management** page.

----End

3 Connection Management

3.1 Enabling or Disabling SSL

Scenarios

DDS allows you to use SSL to encrypt connections to a DB instance to protect your data.

- If SSL is enabled, you can connect to a DB instance using SSL. For details, see sections about connecting to a DB instance using SSL over public or private networks in the *Document Database Service Getting Started*.
- If SSL is disabled, you can connect to the DB instance using a common connection. For details, see sections about connecting to a DB instance using a common connection over public or private networks in the *Document Database Service Getting Started*.

NOTICE

Enabling or disabling SSL will cause DB instance restart. Exercise caution when you perform this operation.

Enabling SSL

Step 1 On the **Instance Management** page, click the target DB instance.

Step 2 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Alternatively, in the navigation pane on the left, choose **Connections**. In the **Basic Information** area, click  next to the **SSL** field.

Step 3 In the displayed dialog box, click **Yes**.

Step 4 In the **Basic Information** area, view the modification result.

----End

Disabling SSL

Step 1 On the **Instance Management** page, click the target DB instance.

Step 2 In the **DB Information** area on the **Basic Information** page, click next to the **SSL** field.

Alternatively, in the navigation pane on the left, choose **Connections**. In the **Basic Information** area, click next to the **SSL** field.

Step 3 In the displayed dialog box, click **Yes**.

Step 4 In the **Basic Information** area, view the modification result.

----End

3.2 Configuring Cross-CIDR Access

Add the VPC CIDR block of your client. Ensure that the ECS where your client is installed can connect to the DB instance.

This section describes how to configure cross-CIDR access for a replica set instance.

Procedure

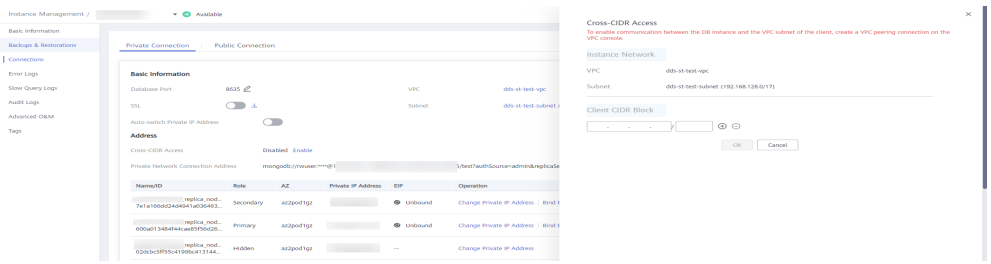
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Connections**.

Step 4 On the **Private Connection** tab, click **Enable** to the right of **Cross-CIDR Access** and specify the client CIDR block details.

Figure 3-1 Cross-CIDR Access



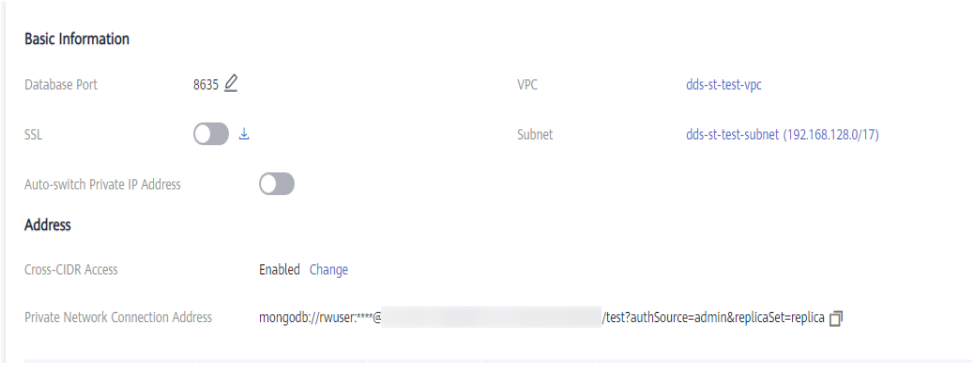
NOTE

Up to 9 CIDR blocks can be configured, and each of them does not overlap.

Step 5 After cross-CIDR access is enabled, **Enabled** is displayed to the right of **Cross-CIDR Access**.

If you need to change the client CIDR block, click **Change** to the right of **Cross-CIDR Access**. Currently, you can only add VPCs and subnets but not change or delete them.

Figure 3-2 Changing a CIDR block



NOTE

- To ensure the ECS and the DB instance can communicate with each other, configure the connection by referring to **VPC Peering Connection Overview**.

----End

3.3 Enabling IP Addresses of shard and config Nodes

A cluster instance of Community Edition consists of mongos, shard, and config nodes. When your services need to read and write data from and into databases, connect to the mongos node. In certain scenarios (for example, data migration and synchronization between clusters), you need to read data from the shard or config node. Therefore, obtaining the IP address of the corresponding node is necessary.

This section describes how to obtain the IP addresses of the shard and config nodes.

Before You Start

- DDS supports cluster instances of Community Edition 3.4 and 4.0.
- DDS creates two connection addresses for the primary node and secondary node in a shard respectively.

  The network type of the connection address is the same as that of the current mongos node.
- Once the connection addresses are assigned to your nodes, they cannot be modified or deleted.
- The connection address is accessible from private networks only.

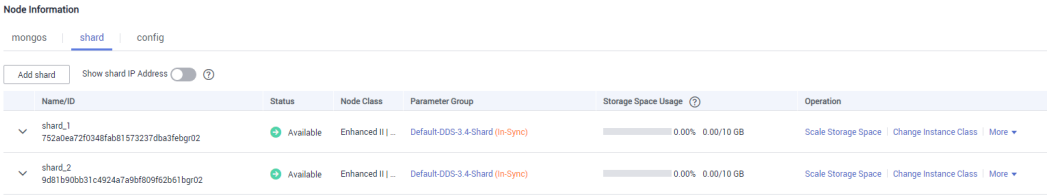
Enabling shard IP Address

Step 1 **Log in to the DDS console.**

Step 2 In the left navigation pane, choose **Instance Management**. In the DB instance list, click the target DB instance to go to the **Basic Information** page.

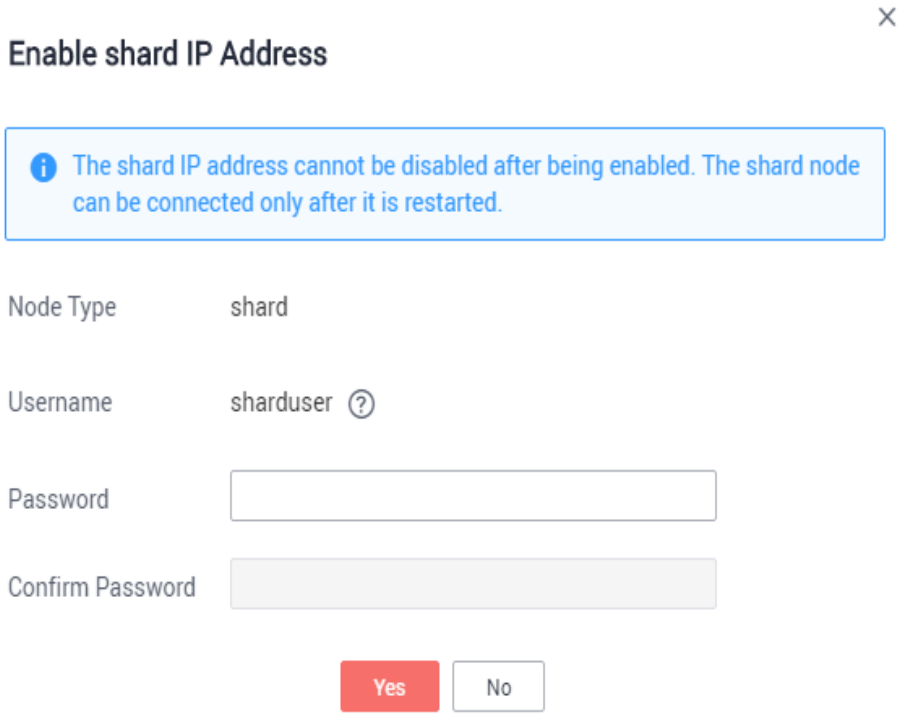
Step 3 On the displayed page, in the **Node Information** area, click the **shard** tab.

Figure 3-3 shard nodes



Step 4 Click **Show shard IP Address**. In the displayed dialog box, enter and confirm the password for connecting to the node.

Figure 3-4 Show shard IP Address



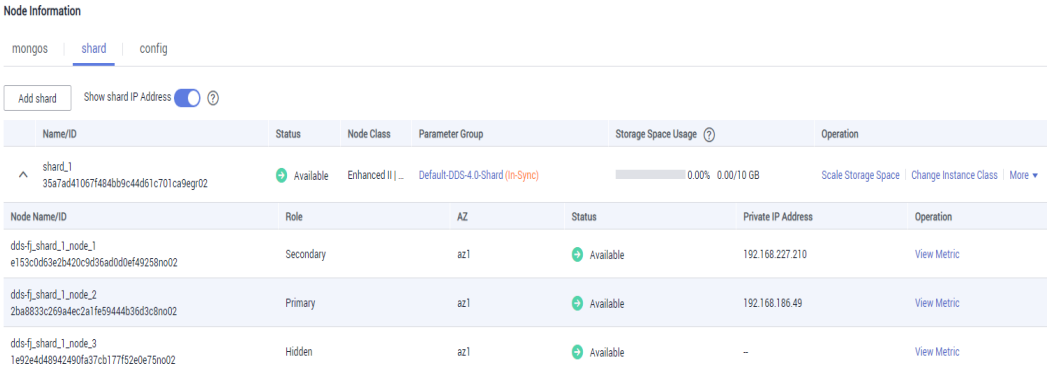
NOTE

- After the shard IP address is enabled, you need to restart the corresponding shard node for the configuration to take effect.
- The shard IP address cannot be modified or disabled after being enabled, and the password cannot be modified.
- After the shard IP address is enabled and new shard nodes are added, you need to manually locate the newly added shard node and choose **More** > **Show shard IP Address** in the **Operation** column to show the shard IP address.
- After the shard IP address is enabled, all shard nodes in the current instance apply for connection addresses.

Step 5 Obtain the private IP address of the shard node.

After the shard IP address is enabled, you can click **Connections** in the navigation pane on the left or on the current page to expand the node drop-down list and obtain the private IP address.

Figure 3-5 Private IP addresses of shard nodes



Step 6 Obtain the connection address of a shard node.

Example:

Based on the private IP address obtained in **Step 5**, the connection address of the current shard node is as follows:

mongodb://***sharduser:****@192.168.237.15:8637,192.168.255.217:8637***/test?authSource=admin&replicaSet=shard_?

NOTE

- **sharduser** indicates the username of the current shard node.
- ******** indicates the password of the current node.
- **192.168.237.15** and **192.168.255.217** are the private IP addresses of the current node.
- **8637** indicates the actual port number of the current node.
- **shard_?** indicates the name of the shard node to be connected, for example, shard_1.

----End

Enabling config IP Address

Step 1 Log in to the DDS console.

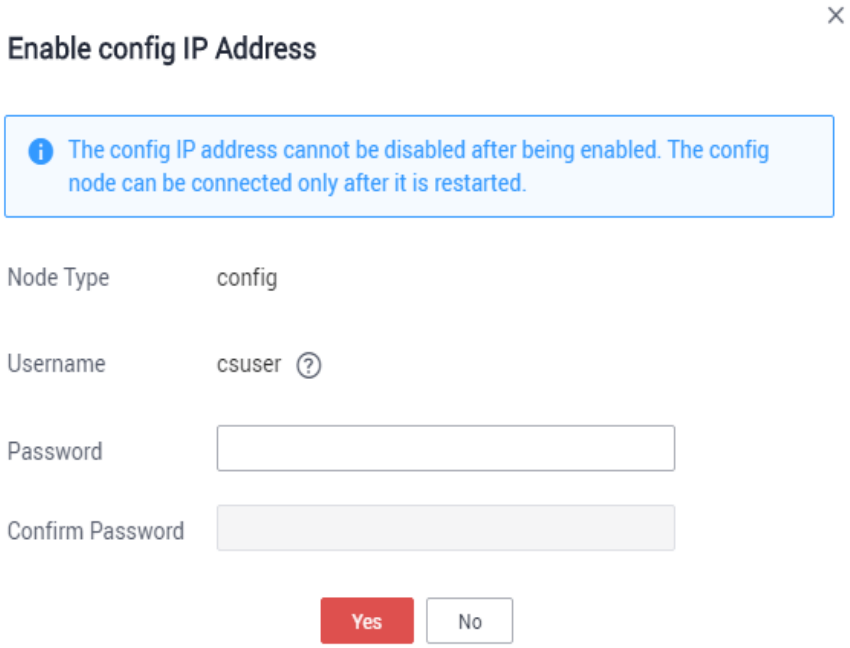
Step 2 In the left navigation pane, choose **Instance Management**. In the DB instance list, click the target DB instance to go to the **Basic Information** page.

Step 3 On the displayed page, in the **Node Information** area, click the **config** tab.

Figure 3-6 config nodes

Step 4 Click **Show config IP Address**. In the displayed dialog box, enter and confirm the password for connecting to the node.

Figure 3-7 Enable config IP Address



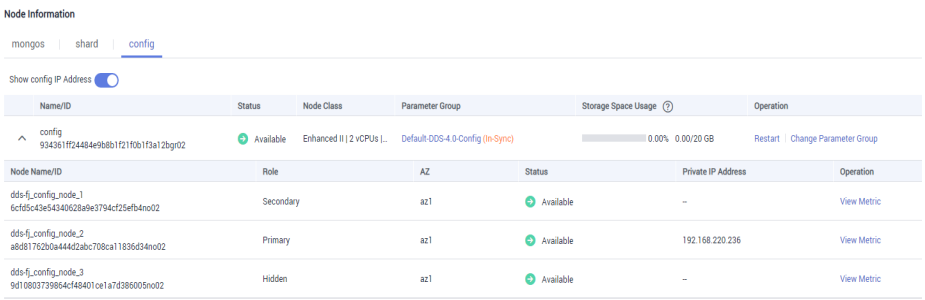
NOTE

- After the config IP address is enabled, you need to restart the corresponding config node for the configuration to take effect.
- The config IP address cannot be modified or disabled after being enabled, and the password cannot be modified.
- After the config IP address is enabled, all config nodes in the current instance apply for connection addresses.

Step 5 Obtain the private IP address of the config node.

After the config IP address is enabled, you can click **Connections** in the navigation pane on the left or on the current page to expand the node drop-down list and obtain the private IP address.

Figure 3-8 Private IP addresses of config nodes



Step 6 Obtain the connection address of a config node.

Example:

Based on the private IP address obtained in **Step 5**, the connection address of the current config node is as follows:

mongodb://***csuser:****@192.168.154.71:8636,192.168.143.227:8636***/test?authSource=admin&replicaSet=config

NOTE

- **csuser** indicates the username of the current config node.
- ******** indicates the password of the current node.
- **192.168.154.71** and **192.168.143.227** are the private IP addresses of the current node.
- **8636** indicates the actual port number of the current node.

----End

3.4 Changing a Private IP Address

Scenarios

After a database is migrated from on-premises or other cloud platforms to DDS, the private IP address of the database may be changed. DDS allows you to change the private IP address, reducing migration costs.

Constraints

Changing the private IP address of a node will invalidate the previous private IP address. If an EIP is bound to the node, do not unbind the EIP during the change of the private IP address. After the change, the new private IP address is bound to the EIP.

Procedure

Step 1 **Log in to the DDS console.**

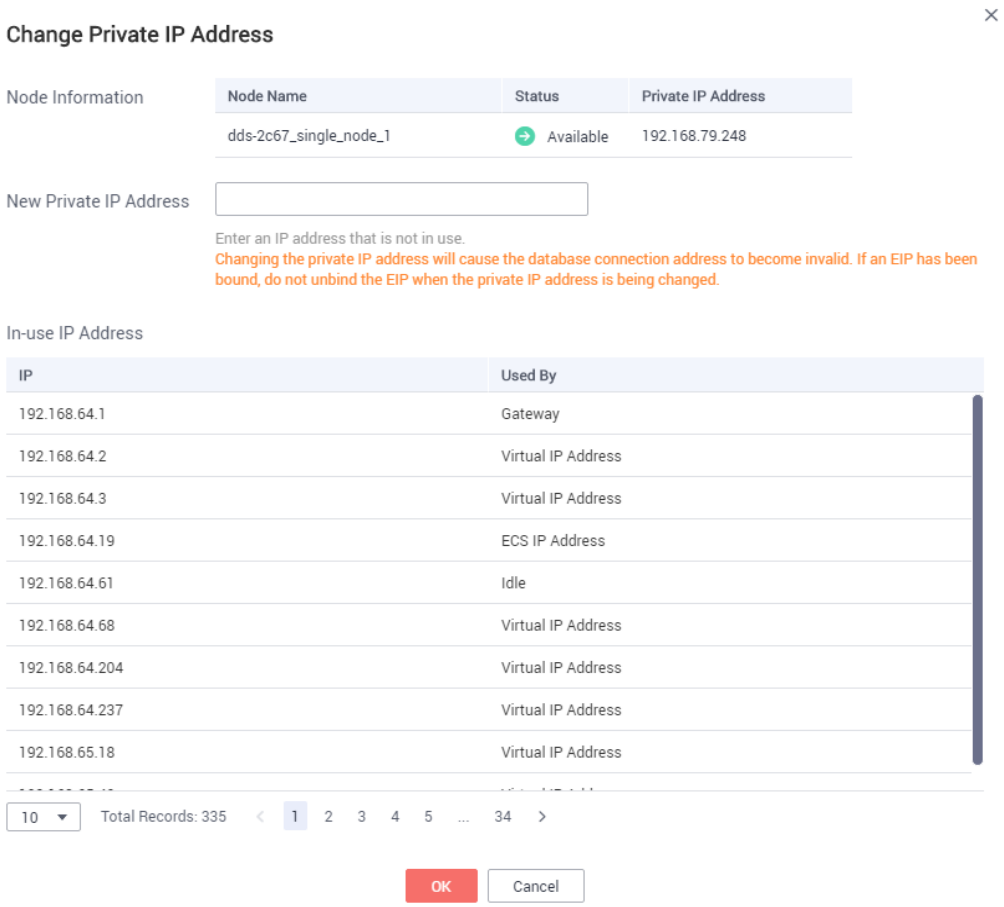
Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Connections**.

Step 4 In the **Node Information** area, locate the target node and in the **Operation** column, click **Change Private IP Address**.

Step 5 In the displayed dialog box, enter a private IP address that is not in use and click **OK**.

Figure 3-9 Changing a private IP address



Step 6 In the **Node Information** area, locate the target node and view the new private IP address.

----End

3.5 Changing the Database Port

Scenarios

This section guides you on how to modify the database port to ensure system security. The database port cannot be changed when the instance is in any of the following statuses:

- Frozen
- Restarting
- Adding node
- Switching SSL
- Changing instance class
- Deleting node

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Connections**.

Step 4 In the **Basic Information** area, click to right of the **Database Port** field.

The database port range is 2100 to 9500.

- To submit the change, click . This process takes about 1 to 5 minutes.
- To cancel the change, click .

Step 5 View the modification result.

----End

3.6 Changing a Security Group

Scenarios

This section guides you on how to change a security group for cluster, replica set, and single node instances. If any of the following operations is in progress, do not change the security group:

- Adding nodes
- Migrating data

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Connections**.

Step 4 In the **Security Group** area, click to select the security group to which the DB instance belongs.

- To submit the change, click . This process takes about 1 to 3 minutes.
- To cancel the change, click .

Step 5 View the modification result.

----End

4 Database Commands

4.1 Which Commands are Supported or Restricted by DDS?

The following tables list the commands supported and restricted by DDS (Community Edition).

For more information, see **official MongoDB documentation**.

Table 4-1 Commands supported and restricted by DDS

Type	Command	Supported	Description
Aggregates Commands	aggregate	√	-
	count	√	-
	distinct	√	-
	group	√	-
	mapReduce	√	This command can be used only when the **security.javascriptEnabled** parameter in the parameter group associated with the DB instance is set to **true**. For more information, see **MapReduce Commands**.
Geospatial Commands	geoNear	√	-
	geoSearch	√	-
Query and Write	find	√	-
	insert	√	-

Type	Command	Supported	Description
Operation Commands	update	√	-
	delete	√	-
	findAndModify	√	-
	getMore	√	-
	getLastError	√	-
	resetError	√	-
	getPrevError	√	-
	parallelCollectionScan	√	-
Query Plan Cache Commands	planCacheListFilters	√	-
	planCacheSetFilter	√	-
	planCacheClearFilters	√	-
	planCacheListQueryShapes	√	-
	planCacheListPlans	√	-
	planCacheClear	√	-
Authentication Commands	logout	√	-
	authenticate	√	-
	copydbgetnonce	√	-
	getnonce	√	-
	authSchemaUpgrade	x	System command
User Management Commands	createUser	√	-
	updateUser	√	-
	dropUser	√	-
	dropAllUsersFromDatabase	√	-
	grantRolesToUser	√	-

Type	Command	Supported	Description
	revokeRolesFromUser	√	-
	usersInfo	√	-
Role Management Commands	invalidateUserCache	√	-
	createRole	√	-
	updateRole	√	-
	dropRole	√	-
	dropAllRolesFromDatabase	√	-
	grantPrivileges-ToRole	√	-
	revokePrivilegesFromRole	√	-
	grantRolesToRole	√	-
	revokeRolesFromRole	√	-
	rolesInfo	√	-
Replication Commands	replSetElect	x	System command
	replSetUpdatePosition	x	System command
	appendOplogNote	x	System command
	replSetFreeze	x	System command
	replSetGetStatus	√	-
	replSetInitiate	x	System command
	replSetMaintenance	x	System command
	replSetReconfig	x	System command
	replSetStepDown	x	System command
	replSetSyncFrom	x	System command

Type	Command	Supported	Description
	replSetRequestVotes	x	System command
	replSetDeclareElectionWinner	x	System command
	resync	x	System command
	applyOps	x	System command
	isMaster	√	-
	replSetGetConfig	x	System command
Sharding Commands	flushRouterConfig	x	High-risk commands
	addShard	x	Unauthorized operation
	addShardToZone	√	-
	balancerStart	√	-
	balancerStatus	√	-
	balancerStop	√	-
	removeShardFromZone	√	-
	updateZoneKeyRange	√	-
	cleanupOrphaned	x	High-risk commands
	checkShardingIndex	x	System command
	enableSharding	√	-
	listShards	x	System command
	removeShard	x	High-risk commands
	getShardMap	x	System command
	getShardVersion	√	-
	mergeChunks	√	-
	setShardVersion	x	System command
	shardCollection	√	-

Type	Command	Supported	Description
	shardingState	x	System command
	unsetSharding	x	System command
	split	√	-
	splitChunk	√	-
	splitVector	√	-
	moveChunk	√	-
	movePrimary	√	-
	isdbgrid	√	-
Administration Commands	setFeatureCompatibilityVersion	√	-
	renameCollection	√	-
	dropDatabase	√	-
	listCollections	√	-
	drop	√	-
	create	√	-
	clone	x	System command
	cloneCollection	√	-
	cloneCollectionAsCapped	√	-
	convertToCapped	√	-
	filemd5	√	-
	createIndexes	√	-
	listIndexes	√	-
	dropIndexes	√	-
	fsync	√	-
	clean	x	System command
	connPoolSync	x	System command
	connectionStatus	√	-

Type	Command	Supported	Description
	compact	x	High-risk commands
	collMod	√	-
	reIndex	√	-
	setParameter	x	System configuration command
	getParameter	√	-
	repairDatabase	x	High-risk commands
	repairCursor	x	System command
	touch	√	-
	shutdown	x	High-risk commands
	logRotate	x	High-risk commands
	killOp	√	-
Diagnostic Commands	availableQueryOptions	√	-
	buildInfo	√	-
	collStats	√	-
	connPoolStats	x	System command
	cursorInfo	x	System command
	dataSize	√	-
	dbHash	x	System command
	dbStats	√	-
	diagLogging	x	System command
	driverOIDTest	x	System command
	explain	√	-
	features	√	-
	getCmdLineOpts	x	System command
	getLog	x	System command
	hostInfo	x	System command
	isSelf	x	System command
	listCommands	√	-

Type	Command	Supported	Description
	listDatabases	√	-
	netstat	x	System command
	ping	√	-
	profile	√	-
	serverStatus	√	-
	shardConnPoolStats	x	System command
	top	√	-
	validate	x	System configuration command
	whatsmyuri	√	-
Internal Commands	handshake	x	System command
	_recvChunkAbort	x	System command
	_recvChunkCommit	x	System command
	_recvChunkStart	x	System command
	_recvChunkStatus	x	System command
	_replSetFresh	x	System command
	mapreduce.shardedfinish	x	System command
	_transferMods	x	System command
	replSetHeartbeat	x	System command
	replSetGetRBID	x	System command
	_migrateClone	x	System command
	replSetElect	x	System command
	writeBacksQueued	x	System command
	writebacklisten	x	System command

Type	Command	Supported	Description
System Events Auditing Commands	logApplication Message	x	System command

4.2 MapReduce Commands

Overview

MapReduce commands are used for executing map-reduce operations on big data.

How Do I Enable MapReduce Commands?

The MapReduce command is controlled by the DDS parameter **security.javascriptEnabled**. The default value is **false**, indicating that the **mapReduce** and **group** commands are unavailable. If you need to use the MapReduce command, change the parameter value to **true**. Then, restart the DB instance for the change to take effect.

- For a cluster instance, change the parameter values in the parameter groups associated with all shard nodes and restart the instance for the change to take effect.
- For a replica set or single-node instance, change the parameter values in the parameter group associated with the instance and restart the instance for the change to take effect.

For details about how to change parameter values, see **Editing a Parameter Group**.

Common Errors and Handling Methods for MapReduce Command Execution Failures

Error information: cannot run map reduce without the js engine or map is not defined

Figure 4-1 Case 1

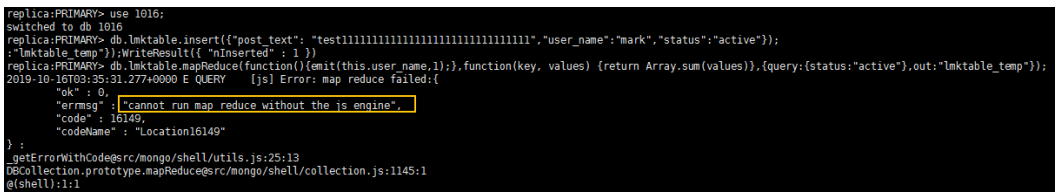
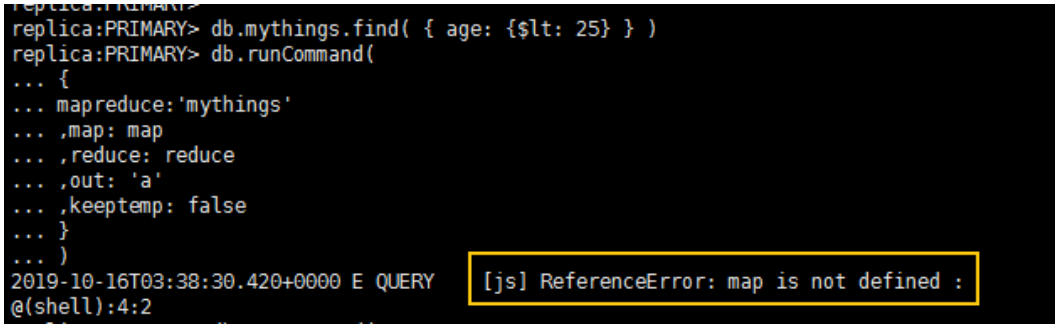


Figure 4-2 Case 2



Possible cause: MapReduce commands are restricted and cannot be used.

Solution: Change the value of **security.javascriptEnabled** in the instance parameter group to **true** and restart the instance for the change to take effect.

NOTE

If a parameter group is a default parameter group, you are not allowed to change its parameter values. You can create a parameter group and change the corresponding parameter values. After the change, associate the new parameter group with the DB instance. For details, see **Changing Associated Parameter Group**.

5 Migrating Data

5.1 Migrating Data Using mongoexport and mongoimport

Scenarios

DDS supports access through EIPs by enabling public accessibility. You can also access a database through an ECS in a private network.

Before migrating data from a MongoDB database to DDS, transfer data to a .json file using the mongoexport tool. This section describes how to import the data from the JSON files to DDS using the mongoimport tool on the ECS or the device that can access DDS.

Precautions

- You are advised to perform the migration during off-peak hours to avoid the impact of migration on your services.
- The admin and local system databases cannot be migrated.
- Ensure that no service set is created in the system databases admin and local in the source database. If service sets already exist, migrate them out of the system databases admin and local before migration.
- Before importing data, ensure that necessary indexes exist on the source database. That is, delete unnecessary indexes and create necessary indexes before migration.
- If you choose to migrate a sharded cluster, you must create a set of shards in the destination database and configure sharding. In addition, indexes must be created before migration.

Prerequisites

1. An ECS or a device that can access DDS is ready for use.
   - To connect to a DDS DB instance through a private network from an ECS, you need to create and log in to the ECS. For details, see **Purchasing an ECS** and **Logging In to an ECS**.

- To bind an EIP to a DB instance:
  i. Bind an EIP to a node in the DB instance. For details about how to bind an EIP to a node, see "Binding an EIP" in the *Document Database Service Getting Started*.
  ii. Ensure that your local device can access the EIP that has been bound to the DB instance.

2. A migration tool has been installed on the prepared ECS.

   For details on how to install the migration tool, see **How Can I Install a MongoDB Client?**

   NOTE

   The MongoDB client provides the mongoexport and mongoimport tools.

Exporting Data

Step 1 Log in to the ECS or the device that can access DDS.

Step 2 Use the mongoexport tool to transfer data from the source database to a .json file.

The SSL connection is used as an example. If you select a common connection, delete **--ssl --sslAllowInvalidCertificates** from the following command.

./**mongoexport --host** *<DB_ADDRESS>* **--port** *<DB_PORT>* **--ssl --sslAllowInvalidCertificates --type json --authenticationDatabase** *<AUTH_DB>* **-u** *<DB_USER>* **--db** *<DB_NAME>* **--collection** *<DB_COLLECTION>* **--out** *<DB_PATH>*

- **DB_ADDRESS** indicates the database address.
- **DB_PORT** indicates the database port.
- **AUTH_DB** indicates the database for storing DB_USER information. Generally, this value is **admin**.
- **DB_USER** indicates the database user.
- **DB_NAME** indicates the name of the database from which data will be exported.
- **DB_COLLECTION** indicates the collection of the database from which data will be exported.
- **DB_PATH** indicates the path where the .json file is located.

Enter the database administrator password when prompted:

```
Enter password:
```

The following is an example. After the command is executed, the **exportfile.json** file will be generated:

./mongoexport --host 192.168.1.21 --port 8635 --ssl --sslAllowInvalidCertificates --type json --authenticationDatabase admin -u rwuser --db test02 --collection Test --out /tmp/mongodb/export/exportfile.json

Step 3 Check the result.

If information similar to the following is displayed, the data is successfully exported. **x** indicates the number of exported data records.

```
exported x records
```

Step 4 Compress the exported .json file.

gzip exportfile.json

Compressing the file helps reduce the time needed to transmit all the data. The compressed file is **exportfile.json.gz**.

----End

Importing Data

Step 1 Log in to the ECS or the device that can access DDS.

Step 2 Upload the data to be imported to the ECS or the device that can access DDS.

Select an uploading method based on the OS you are using. In Linux, for example, run the following command:

scp *<IDENTITY_FILE> <REMOTE_USER>@<REMOTE_ADDRESS>:<REMOTE_DIR>*

- **IDENTITY_FILE** indicates the directory where the **exportfile.json.gz** file is located. The file access permission is 600.
- **REMOTE_USER** indicates the ECS OS user.
- **REMOTE_ADDRESS** indicates the ECS address.
- **REMOTE_DIR** indicates the directory of the ECS to which the **exportfile.json.gz** file is uploaded.

In Windows, upload **exportfile.json.gz** to the ECS using file transfer tools.

Step 3 Decompress the package.

gzip -d *exportfile.json.gz*

Step 4 Import the JSON file to the DDS database.

The SSL connection is used as an example. If you select a common connection, delete **--ssl --sslAllowInvalidCertificates** from the following command.

./**mongoimport --host** <*DB_ADDRESS*> **--port** <*DB_PORT*> **--ssl --sslAllowInvalidCertificates --type json --authenticationDatabase** <*AUTH_DB*> **-u** <*DB_USER*> **--db** <*DB_NAME*> **--collection** <*DB_COLLECTION*> **--file** <*DB_PATH*>

- **DB_ADDRESS** indicates the DB instance IP address.
- **DB_PORT** indicates the database port.
- **AUTH_DB** indicates the database that authenticates DB_USER. Generally, this value is **admin**.
- **DB_USER** indicates the account name of the database administrator.
- **DB_NAME** indicates the name of the database to which data will be imported.
- **DB_COLLECTION** indicates the collection of the database to which data will be imported.

- **DB_PATH** indicates the path where the .json file is located.

Enter the database administrator password when prompted:

```
Enter password:
```

The following is an example:

./mongoimport --host 192.168.1.21 --port 8635 --ssl --sslAllowInvalidCertificates --type json --authenticationDatabase admin -u rwuser --db test02 --collection Test --file /tmp/mongodb/export/exportfile.json

Step 5 Check the result.

If information similar to the following is displayed, the data is successfully imported. **x** indicates the number of imported data records.

```
imported x records
```

----End

5.2 Migrating Data Using mongodump and mongorestore

Scenarios

DDS supports access through EIPs by enabling public accessibility. To access DDS from an ECS, you need to create an ECS first and then install mongodump and mongorestore on it.

Precautions

- You are advised to perform the migration during off-peak hours to avoid the impact of migration on your services.
- The admin and local system databases cannot be migrated.
- Ensure that no service set is created in the system databases admin and local in the source database. If service sets already exist, migrate them out of the system databases admin and local before migration.
- Before importing data, ensure that necessary indexes exist on the source database. That is, delete unnecessary indexes and create necessary indexes before migration.
- If you choose to migrate a sharded cluster, you must create a set of shards in the destination database and configure sharding. In addition, indexes must be created before migration.
- If the backup using the mongodump tool fails (for example, an error is reported when the backup progress reaches 97%), you are advised to increase the VM storage space and reserve some redundant space before performing the backup again.

Prerequisites

1. An ECS or a device that can access DDS is ready for use.

- To connect to a DDS DB instance through a private network from an ECS, you need to create and log in to the ECS. For details, see **Purchasing an ECS** and **Logging In to an ECS**.
- To bind an EIP to a DB instance:
  i. Bind an EIP to a node in the DB instance. For details about how to bind an EIP to a node, see "Binding an EIP" in the *Document Database Service Getting Started*.
  ii. Ensure that your local device can access the EIP that has been bound to the DB instance.

2. A migration tool has been installed on the prepared ECS.

   For details on how to install the migration tool, see **How Can I Install a MongoDB Client?**

   NOTE

   The MongoDB client provides the mongoexport and mongoimport tools.

Backing Up Data

Step 1 Log in to the ECS or the device that can access DDS.

Step 2 Back up the source database data using the mongodump tool.

The SSL connection is used as an example. If you select a common connection, delete **--ssl --sslCAFile** *<FILE_PATH>* **--sslAllowInvalidCertificates** from the following command.

mongodump --host *<DB_HOST>* **--port** *<DB_PORT>* **-u** *<DB_USER>* **--authenticationDatabase** *<AUTH_DB>* **--ssl --sslCAFile** *<FILE_PATH>* **--sslAllowInvalidCertificates**

- **DB_HOST** indicates the database address.
- **DB_PORT** indicates the database port.
- **DB_USER** indicates the database user.
- **AUTH_DB** indicates the database for storing DB_USER information. Generally, this value is **admin**.
- **FILE_PATH** indicates the path where the root certificate is stored.

Enter the database administrator password when prompted:

```
Enter password:
```

After the following example command is executed, the source database data is backed up to the **dump** folder in the current directory.

./mongodump --host 192.168.6.39 --port 8635 -u rwuser --authenticationDatabase admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidCertificates

```
2019-03-04T18:42:10.687+0800    writing admin.system.users to
2019-03-04T18:42:10.688+0800    done dumping admin.system.users (1 document)
2019-03-04T18:42:10.688+0800    writing admin.system.roles to
2019-03-04T18:42:10.690+0800    done dumping admin.system.roles (0 documents)
2019-03-04T18:42:10.690+0800    writing admin.system.version to
2019-03-04T18:42:10.691+0800    done dumping admin.system.version (2 documents)
2019-03-04T18:42:10.691+0800    writing test.test_collection to
```



```
2019-03-04T18:42:10.691+0800    writing admin.system.profile to
2019-03-04T18:42:10.692+0800    done dumping admin.system.profile (4 documents)
2019-03-04T18:42:10.695+0800    done dumping test.test_collection (198 documents)
```

----End

Importing Data

Step 1 Log in to the ECS or the device that can access DDS.

Step 2 Upload the data to be imported to the ECS or the device that can access DDS.

Select an uploading method based on the OS you are using. In Linux, for example, run the following command:

scp -r *<IDENTITY_DIR> <REMOTE_USER>@<REMOTE_ADDRESS>:<REMOTE_DIR>*

- **IDENTITY_DIR** indicates the directory that stores the backup file.
- **REMOTE_USER** indicates the ECS OS user in **Step 1**.
- **REMOTE_ADDRESS** indicates the ECS address in **Step 1**.
- **REMOTE_DIR** indicates the directory of the ECS to which the data is uploaded.

In Windows, upload the backup directory to the ECS using file transfer tools.

Step 3 Import the backups in the DDS database.

The SSL connection is used as an example. If you select a common connection, delete **--ssl --sslCAFile** <*FILE_PATH*> **--sslAllowInvalidCertificates** from the following command.

./mongorestore --host *<DB_HOST>* **--port** *<DB_PORT>* **-u** *<DB_USER>* **--authenticationDatabase** *<AUTH_DB> <Backup directory>* **--ssl --sslCAFile** <*FILE_PATH*> **--sslAllowInvalidCertificates**

- **DB_HOST** indicates the database address.
- **DB_PORT** indicates the database port.
- **DB_USER** indicates the account name of the database administrator. The default value is **rwuser**.
- **AUTH_DB** indicates the database that authenticates DB_USER. Generally, this value is **admin**.
- **Backup directory**: indicates the directory for storing backup files. The default value is **dump**.
- **FILE_PATH** indicates the path where the root certificate is stored.

Enter the database administrator password when prompted:

```
Enter password:
```

The following is an example:

./mongorestore --host 192.168.6.187 --port 8635 -u rwuser --authenticationDatabase admin dump --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidCertificates

```
2019-03-05T14:19:43.240+0800    preparing collections to restore from
2019-03-05T14:19:43.243+0800    reading metadata for test.test_collection from dump/test/
test_collection.metadata.json
```



```
2019-03-05T14:19:43.263+0800    restoring test.test_collection from dump/test/test_collection.bson
2019-03-05T14:19:43.271+0800    restoring indexes for collection test.test_collection from metadata
2019-03-05T14:19:43.273+0800    finished restoring test.test_collection (198 documents)
2019-03-05T14:19:43.273+0800    restoring users from dump/admin/system.users.bson
2019-03-05T14:19:43.305+0800    roles file 'dump/admin/system.roles.bson' is empty; skipping roles
restoration
2019-03-05T14:19:43.305+0800    restoring roles from dump/admin/system.roles.bson
2019-03-05T14:19:43.333+0800    done
```

----End

5.3 Migrating Data Using DRS

Data Replication Service (DRS) helps migrate your databases to DDS DB instances. During the database migration, the source remains operational even if a transfer is interrupted, thereby minimizing application downtime.

DRS supports the following source database types:

- On-premises MongoDB databases
- Self-built MongoDB database on the ECS
- Databases on other clouds
- HUAWEI CLOUD DDS DB instances

The following database versions are supported:

- Cluster 3.4 and 4.0
- Replica set 3.4 and 4.0
- Single node 3.4 and 4.0

NOTE

Data cannot be migrated from a later version database to an earlier version database.

Migrate Scenario

Scenario 1: Full Migration

This migration type is suitable for scenarios where service interruption is acceptable. All objects and data in non-system databases are migrated to the destination database at one time. The objects include tables, views, and stored procedures.

NOTE

If you perform a full migration, you are advised to stop operations on the source database. Otherwise, data generated in the source database during the migration will not be synchronized to the destination database.

Currently, DRS supports full migration between the following types of DB instances:

- Replica set -> Single node
- Replica set -> Replica set
- Replica set -> Cluster
- Single node -> Single node

- Single node -> Replica set
- Single node -> Cluster
- Cluster -> Cluster

Scenario 2: Full+Incremental Migration

This migration type allows you to migrate data without interrupting services. After a full migration initializes the destination database, an incremental migration initiates and parses logs to ensure data consistency between the source and destination databases.

NOTE

If you select the **Full+Incremental** migration type, data generated during the full migration will be synchronized to the destination database with zero downtime, ensuring that both the source and destination databases remain accessible.

Currently, DRS supports combined full and incremental migration between the following types of DB instances:

- Replica set -> Single node
- Replica set -> Replica set
- Replica set -> Cluster
- Single node -> Single node
- Single node -> Replica set
- Single node -> Cluster
- Cluster -> Cluster

Before You Start

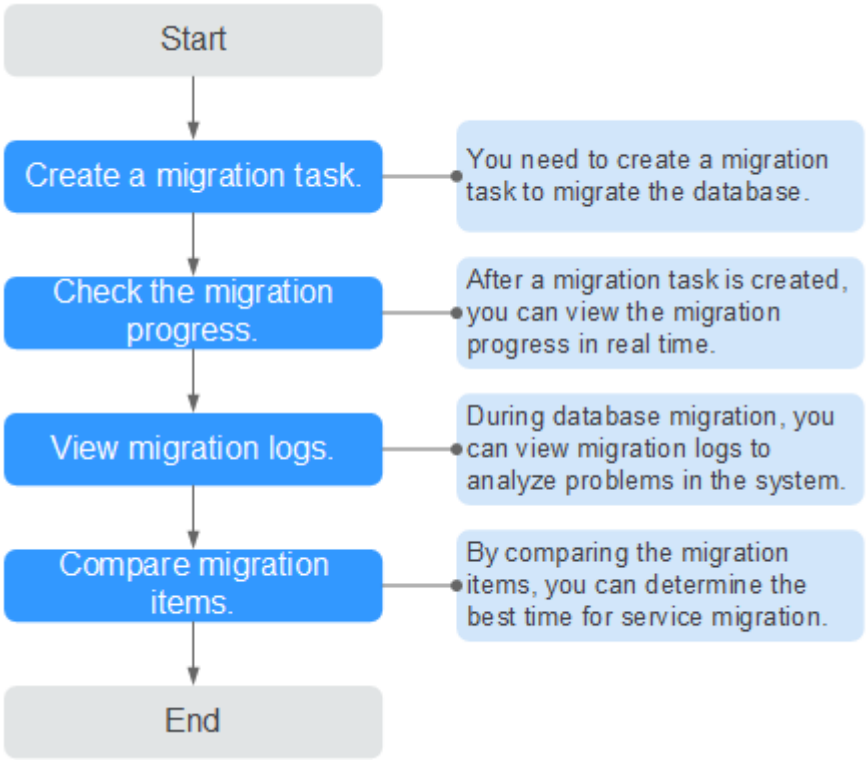
To improve the stability and security of data migration ensure that your DB instances meet the migration requirements described in **Migration Preparations**.

Migration Operations

A complete real-time migration consists of creating a migration task, tracking task progress, analyzing migration logs, and comparing data consistency. By comparing multiple items and data, you can determine the proper time for service migration to minimize the service downtime.

The following flowchart shows the migration procedure.

Figure 5-1 Migration process



For details, see section **MongoDB Database Migration** in the *Data Replication Service Best Practices*.

6 Database Management

6.1 Creating a Database Account through DAS

Scenarios

This section describes how to create a database account and change the account password on the DAS console after the DDS DB instances are created.

NOTE

When creating a database account for a specified DB instance, you are advised to enable the SSL connection to improve data security.

Constraints

If the existing DDS DB instances are of version 3.2, you cannot create database accounts for them. You can only change the password of the administrator account **rwuser**.

Prerequisites

You have logged in to a DDS DB instance through DAS.

- For details on how to connect a cluster instance through DAS, see **Connecting to a DB Instance Through DAS**.
- For details on how to connect a replica set instance through DAS, see **Connecting to a DB Instance Through DAS**.
- For details on how to connect a single-node instance through DAS, see **Connecting to a DB Instance Through DAS**.

Account Description

To manage DDS DB instances, users **root** (or **admin**), **monitor**, and **backup** are automatically created when you create a DDS DB instance. Attempting to delete, rename, change the passwords, or change privileges for these accounts will result in errors.

You can change the password of the database administrator **rwuser** and any accounts you create.

Setting Password Strength for Database Accounts

- The administrator password must meet the following password policy:
  - Contains 8 to 32 characters.
  - Must be a combination of uppercase letters, lowercase letters, digits, and the following special characters: ~!@#%^*-_=+?
- The database user created on the client must meet the following password policy:
  - Contains 8 to 32 characters.
  - A combination of uppercase letters, lowercase letters, digits, and the following special characters: ~@#%-_!*+=^?

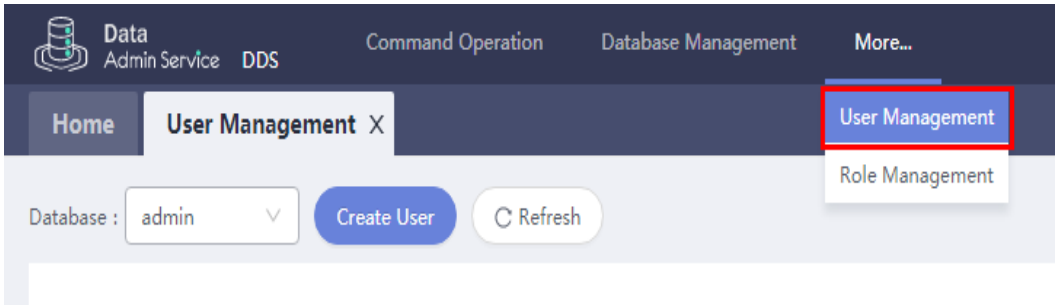
When you create a DB instance, DDS automatically checks your password strength. You can change the password as user **rwuser**. For security reasons, you are advised to set up a strong password.

Creating an Account

Step 1 You have logged in to a DDS DB instance through DAS.

Step 2 On the DAS console, choose **More** > **User Management** and click **Add User**.

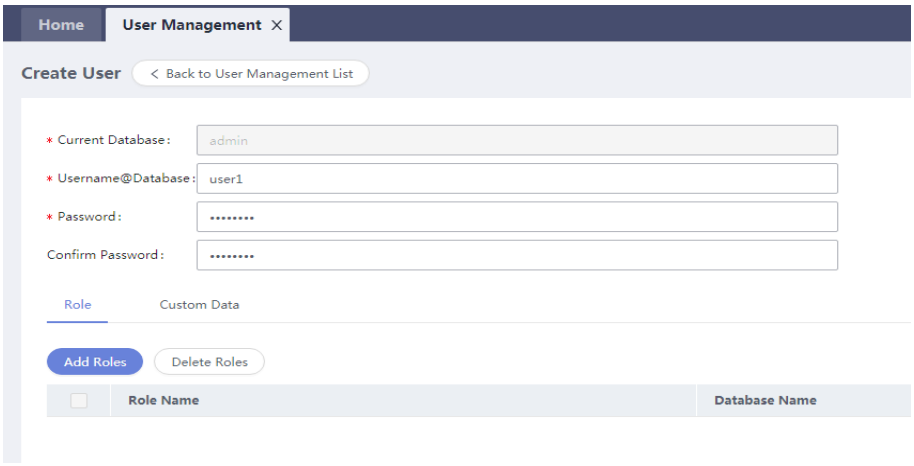
Figure 6-1 User Management



Step 3 On the displayed page, specify required parameters. The following uses adding user **user1** and the **root** role for the admin database as an example:

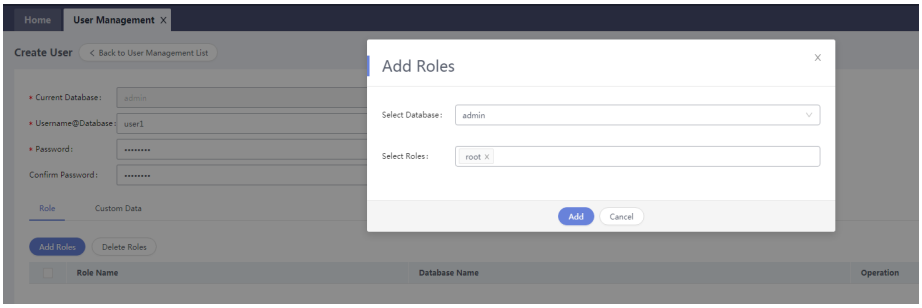
1. Enter the database username and password.

Figure 6-2 Add User



2. Click **Add Roles**, select **root**, and click **Add**.

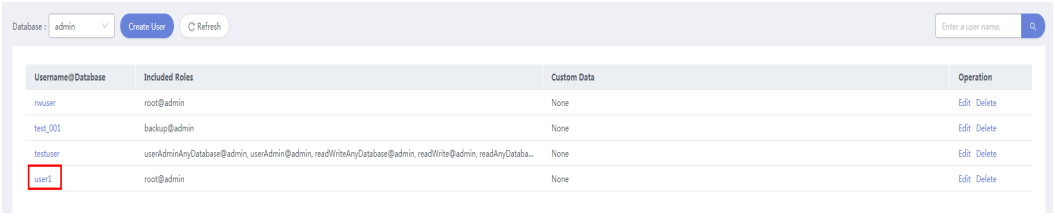
Figure 6-3 Add Roles



3. Click **Save**.

Step 4 If you need to change the password or role, click **Alter**.

Figure 6-4 Add User



----End

6.2 Creating a Database Account Using Commands

Scenarios

This section describes how to create a database account and change the account password using commands after the DDS DB instances are created.

NOTE

When creating a database account for a specified DB instance, you are advised to enable the SSL connection to improve data security.

Constraints

If the existing DDS DB instances are of version 3.2, you cannot create database accounts for them. You can only change the password of the administrator account **rwuser**.

Prerequisites

A DDS DB instance has been connected. For details, see section "Connecting to a DB Instance Through a Public Network" and "Connecting to a DB Instance Through a Private Network" in *Document Database Service Getting Started*.

Account Description

To manage DDS DB instances, users **root** (or **admin**), **monitor**, and **backup** are automatically created when you create a DDS DB instance. Attempting to delete, rename, change the passwords, or change privileges for these accounts will result in errors.

You can change the password of the database administrator **rwuser** and any accounts you create.

Setting Password Strength for Database Accounts

- The administrator password must meet the following password policy:
  - Contains 8 to 32 characters.
  - Must be a combination of uppercase letters, lowercase letters, digits, and special characters: ~!@#%^*-_=+?
- The database user created on the client must meet the following password policy:
  - Contains 8 to 32 characters.
  - Must be a combination of uppercase letters, lowercase letters, digits, and special characters: ~@#%-_!*+=^?

When you create a DB instance, DDS automatically checks your password strength. You can change the password as user **rwuser**. For security reasons, you are advised to set up a strong password.

Creating an Account

Step 1 Run the following command to select the admin database:

use admin

Step 2 Run the following command to create a database account (**user1** as an example):

db.createUser({user: "user1", pwd: "*Test_12345***", passwordDigestor:"server", roles:[{role: "root", db: "admin"}]})**

- ***server***: indicates that the password is encrypted on the server.
- ***Test_12345***: indicates the example new password. The password must be 8 to 32 characters in length and contain uppercase letters, lowercase letters, digits, and special characters, such as ~@#%-_!*+=^?
- ***roles*** restricts the rights of the account. If an empty array is specified, the account does not have any permission.

Step 3 Check the result:

The account is successfully created if the following information is displayed:

```
Successfully added user: {
	"user" : "user1",
	"passwordDigestor" : "server",
	"roles" : [
		{
			"role" : "root",
			"db" : "admin"
		}
	]
}
```

----End

Changing a Password

Step 1 Run the following command to select the admin database:

use admin

Step 2 Uses user **user1** as an example. Run the following command to change its password:

db.updateUser("user1", {passwordDigestor:"server",pwd:"newPasswd12#"})

- ***server***: indicates that the password is encrypted on the server.
- ***newPasswd12#***: indicates the example new password. The password must be 8 to 32 characters in length and contain uppercase letters, lowercase letters, digits, and special characters, such as ~@#%-_!*+=^?

Step 3 Check the setting result. The password is successfully changed if the following information is displayed:

- Cluster
  ```
  mongos>
  ```
- Replica set
  ```
  replica:PRIMARY>
  ```
- Single node
  ```
  replica:PRIMARY>
  ```

----End

6.3 Creating a Database Using Commands

Scenarios

A database is a collection of tables, indexes, views, stored procedures, and operators. To make it easier to manage DDS DB instances, you can create a

database by running commands on the newly-created DB instance. If the database does not exist, create the database and switch to the new database. If the database exists, directly switch to the database.

Prerequisites

A DDS DB instance has been connected. For details, see section "Connecting to a DB Instance Through a Public Network" and "Connecting to a DB Instance Through a Private Network" in *Document Database Service Getting Started*.

Procedure

Step 1 Create a database.

use *dbname*

dbname: indicates the name of the database to be created.

Figure 6-5 Creating databases

```
replica:PRIMARY> use test001
switched to db test001
```

Step 2 After a database is created, insert data into the database so that you can view the database in the database list.

Figure 6-6 Inserting data

```
replica:PRIMARY> db.user.insert({"key1":"value1"})
WriteResult({ "nInserted" : 1 })
replica:PRIMARY> show dbs
admin    0.000GB
local    0.004GB
test     0.000GB
test001  0.000GB
replica:PRIMARY>
```

NOTE

There are three system databases created by default: admin, local, and test. If you directly insert data without creating a database, the data is inserted to the test database by default.

Figure 6-7 Viewing the database

```
replica:PRIMARY> show dbs
admin  0.000GB
local  0.004GB
test   0.000GB
```

Step 3 View data in the database.

Figure 6-8 Viewing data

```
replica:PRIMARY> show collections
user
replica:PRIMARY> db.user.find()
{ "_id" : ObjectId("5da1880d2b4ccf2e1163ad1d"), "key1" : "value1" }
```

----End

6.4 Resetting the Administrator Password

Scenarios

For security reasons, you are advised to periodically change administrator passwords.

If you do not set the administrator password for the DB instance that you are creating, you need to reset the password before connecting to the DB instance.

You cannot reset the administrator password under the following circumstances:

- Frozen
- Restarting
- Adding node
- Switching SSL
- Changing port
- Changing instance class
- Deleting node

Background

If you enable operation protection to improve the security of your account and cloud products, two-factor authentication is required for sensitive operations. For details about how to enable operation protection, see **Critical Operations** in the *Identity and Access Management User Guide*.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 3 Enter and confirm the new administrator password and click **OK**.

The password is a string of 8 to 32 characters. It must be a combination of uppercase letters, lowercase letters, digits, and special characters. You can also use the following special characters: ~!@#%^*-_=+?

Step 4 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify** to close the page.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the **DB Information** area on the **Basic Information** page, click **Reset Password** to the right of the **Administrator** field.

Step 4 Enter and confirm the new administrator password and click **OK**.

The password is a string of 8 to 32 characters. It must be a combination of uppercase letters, lowercase letters, digits, and special characters. You can also use the following special characters: ~!@#%^*-_=+?

Step 5 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify** to close the page.

----End

7 Instance Management

7.1 Changing a DB Instance Name

Scenarios

This section describes how to change a DB instance name to identify different DB instances.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click next to the DB instance name you wish to change.

- If you want to submit the change, click **OK**. The DB instance name can be 4 to 64 characters long. It must start with a letter and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- If you want to cancel the change, click **Cancel**.

Step 3 View the change result on the **Instance Management** page.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the **DB Information** area on the **Basic Information** page, click in the **DB Instance Name** field to change the instance name.

- To submit the change, click . The DB instance name can be 4 to 64 characters long. It must start with a letter and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- To cancel the change, click .

Step 4 View the results on the **Instance Management** page.

----End

7.2 Adding Cluster Instance Nodes

Scenarios

This section describes how to add nodes to a DB instance.

NOTE

- You can add nodes only when your account balance is greater than or equal to ¥0.
- You can add nodes when the instance status is **Available**, **Deleting backup**, or **Checking restoration**.
- A DB instance cannot be deleted when one or more nodes are being added.

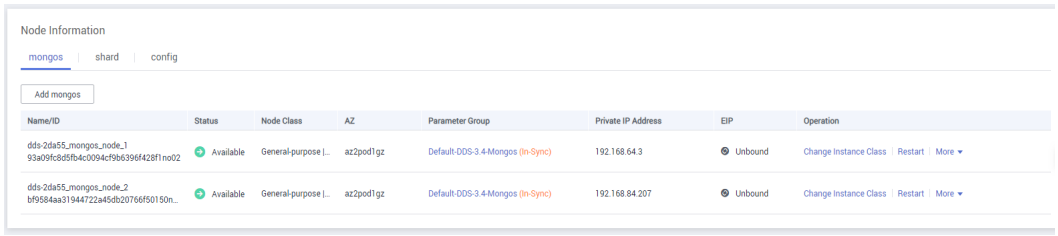
Add mongos

Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 On the **mongos** tab in the **Node Information** area, click **Add mongos**.

Figure 7-1 Node information

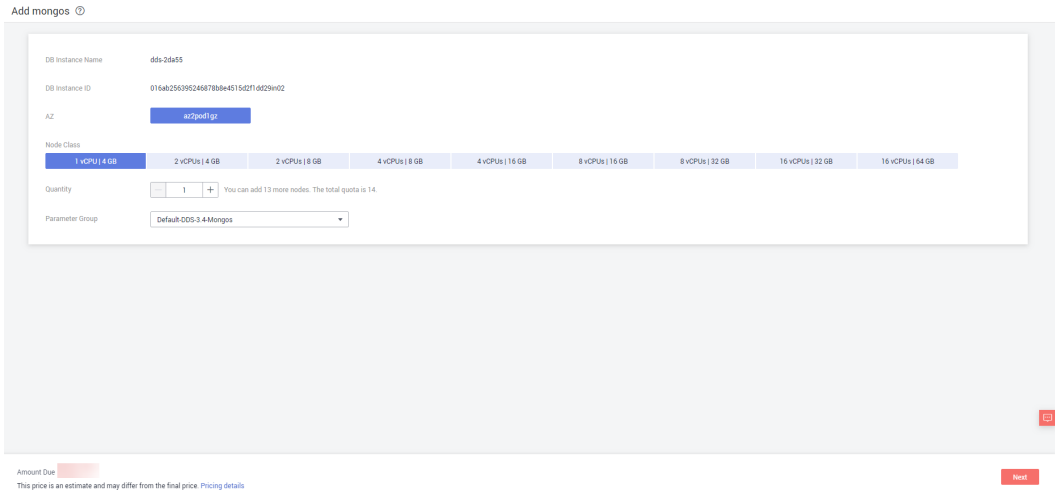


Step 4 On the displayed page, specify **Node Class**, **Quantity**, and **Parameter Group** and click **Next**.

NOTE

If you add mongos nodes to a cluster DB instance deployed across AZs, the new mongos nodes can be deployed in the primary or standby AZ. For details about how to deploy a cluster DB instance across AZs, see **Buying a Cluster Instance**

Figure 7-2 Add mongos



A cluster instance of Community Edition supports up to 32 mongos nodes.

Step 5 On the displayed page, confirm the node configuration information.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to add the nodes.

Step 6 View the result of adding nodes.

- This process takes about 10 to 15 minutes. The status of the DB instance in the instance list is **Adding node**.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- On the **mongos** tab in the **Node Information** area, view the information about the node you added.
- If the mongos fail to be added, you can revert them in batches or delete them one by one. For details, see section **Reverting Cluster Instance Nodes**.

----End

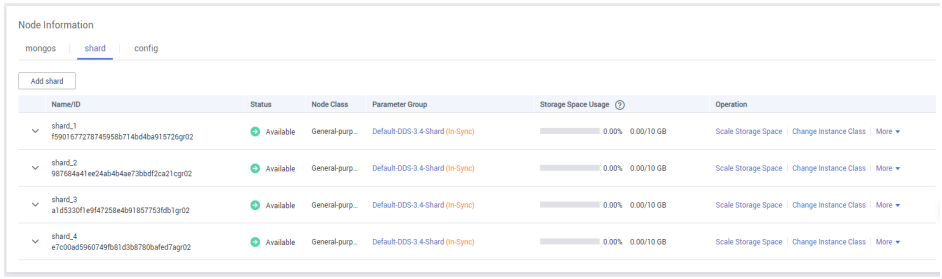
Add shard

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

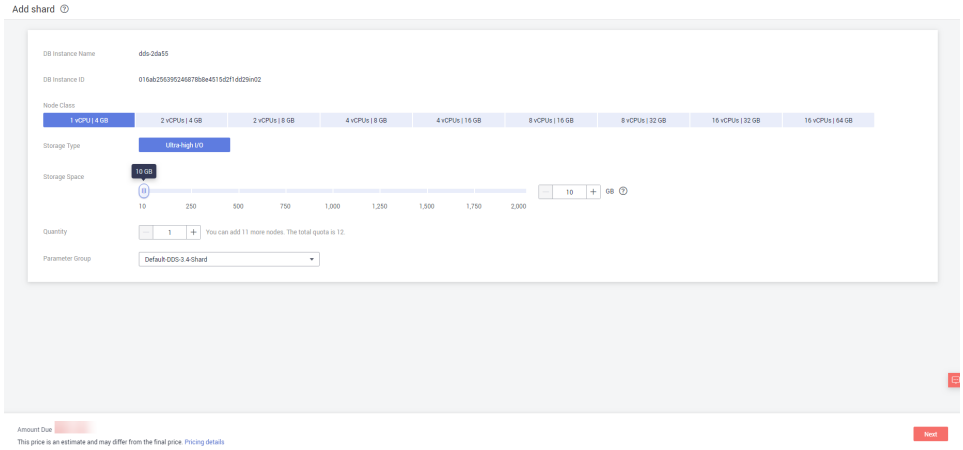
Step 3 On the **shard** tab in the **Node Information** area, click **Add shard**.

Figure 7-3 Node information



Step 4 Specify **Node Class**, **Storage Space**, **Quantity**, and **Parameter Group** and click **Next**.

Figure 7-4 Add shard



- The storage space you applied for will contain the system overhead required for inode, reserved block, and database operation. The storage space must be an integer multiple of 10.
- A cluster instance of Community Edition supports up to 32 shard nodes.

Step 5 On the displayed page, confirm the node configuration information.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to add the nodes.

Step 6 View the result of adding nodes.

- This process takes about 10 to 15 minutes. The status of the DB instance in the instance list is **Adding node**.

- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- On the **shard** tab in the **Node Information** area, view the information about the node you added.
- If the shards fail to be added, you can revert them in batches or delete them one by one. For details, see section **Reverting Cluster Instance Nodes**.

----End

7.3 Reverting Cluster Instance Nodes

Scenarios

This section describes how to revert nodes that fail to be added.

Reverting Nodes in Batches

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the cluster instance to which nodes fail to be added and choose **More** > **Revert** in the **Operation** column.

Step 3 In the displayed dialog box, click **Yes**.

During reversal, the instance status is **Deleting node**. This process takes about 1 to 3 minutes.

----End

Deleting a Single Node

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance to which the node fails to be added.

Step 3 In the **Node Information** area on the **Basic Information** tab, click the **mongos** or **shard** tab, locate the mongos or shard that fail to be added, and choose **More** > **Delete**.

Step 4 In the displayed dialog box, click **Yes**.

During deletion, the node status is **Deleting node**. This process takes about 1 to 3 minutes.

----End

7.4 Adding Replica Set Instance Nodes

Scenarios

DDS allows you to scale the three-node replica set instances up to five-node or even seven-node architecture. All new nodes are secondary nodes and support primary/secondary switchovers, improving data reliability.

Constraints

- You can add nodes when the instance status is **Available**, **Deleting backup**, or **Checking restoration**.
- A DB instance cannot be deleted when one or more nodes are being added.
- After a node is added, data is continuously synchronized to the secondary node. The new connection address configured for your application may take effect only after the application is restarted. To ensure high availability of the connection, you need to activate the newly added node. Then, the new node can participate in the primary/secondary switchover.
- Nodes cannot be manually deleted.
- If you use a three-node, five-node, or seven-node replica set backup to restore data to a new DB instance, the new DB instance uses the three-node replica set architecture.

Procedure

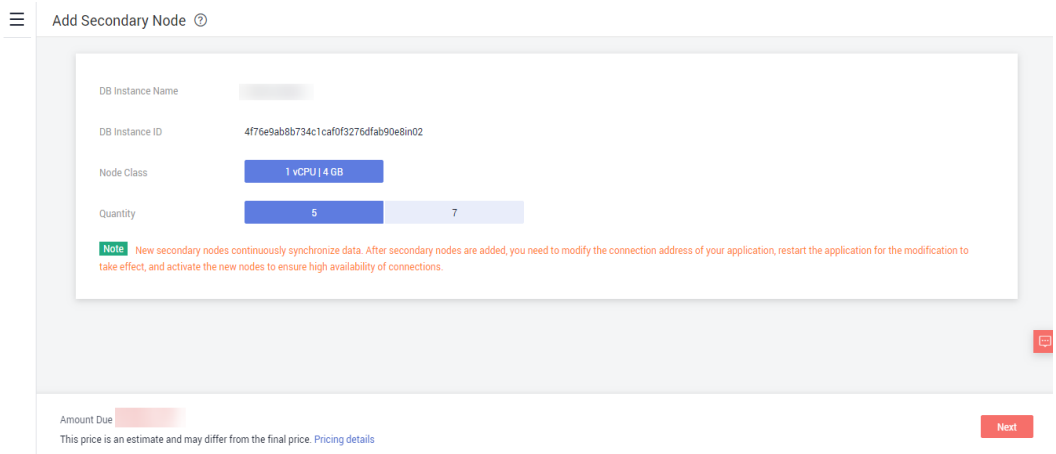
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target replica set instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click **Add Secondary Node**.

Step 4 Specify **Quantity** and click **Next**.

Figure 7-5 Adding secondary nodes



You can add five or seven nodes.

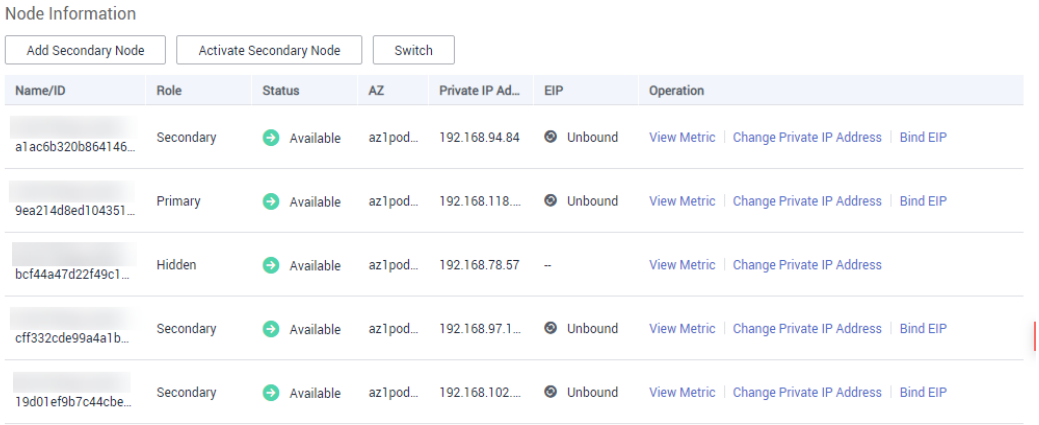
Step 5 On the displayed page, confirm the node configuration information.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to add the nodes.

Step 6 View the result of adding nodes.

- The status of the DB instance in the instance list is **Adding node**.
- In the **Node Information** area, view the information about the nodes you added.

Figure 7-6 Node information



Step 7 In the **Node Information** area, click **Activate Secondary Node** so that the new secondary node can participate in the primary/secondary switchover.

NOTE

During the primary/secondary switchover, the replica set randomly selects a secondary node as the primary node. Ensure that you have modified the connection address of your application.

----End

7.5 Scaling Up Storage Space

Scenarios

This section describes how to scale up the storage space of a DB instance to suit your service requirements.

NOTE

- You cannot scale up a DB instance in **Creating**, **Changing instance class**, **Adding node**, or **Deleting node** status.
- Storage space can only be scaled up. It cannot be scaled down.
- If you scale up a DB instance with disks encrypted, the expanded storage space will be encrypted using the original encryption key.
- You cannot scale up the storage space of a config for the cluster instances.
- During the scale-up process, the DB instance will not restart, and your services will not be interrupted.

Cluster

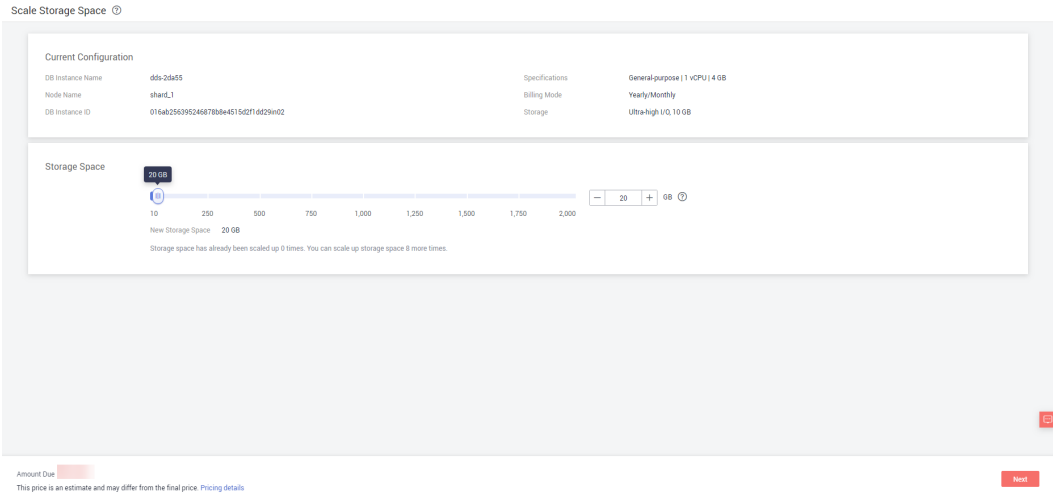
Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click the **shard** tab, locate the target shard, and click **Scale Storage Space** in the **Operation** column.

Step 4 On the displayed page, specify the desired storage space, and click **Submit**.

Figure 7-7 Scale shard



You must add a minimum of 10 GB each time you scale up, and only multiples of 10 GB are allowed. The maximum amount of storage space is 2000 GB.

Step 5 On the displayed page, confirm the storage space.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.

  - If you do not need to modify the specifications, click **Submit** to scale up the storage space.

Step 6 Check the scale-up result.

- This process takes about 3 to 5 minutes. The status of the DB instance in the instance list is **Scaling up**.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- In the **Node Information** area on the **Basic Information** page, click the **shard** tab and check whether the scale up was successful.

----End

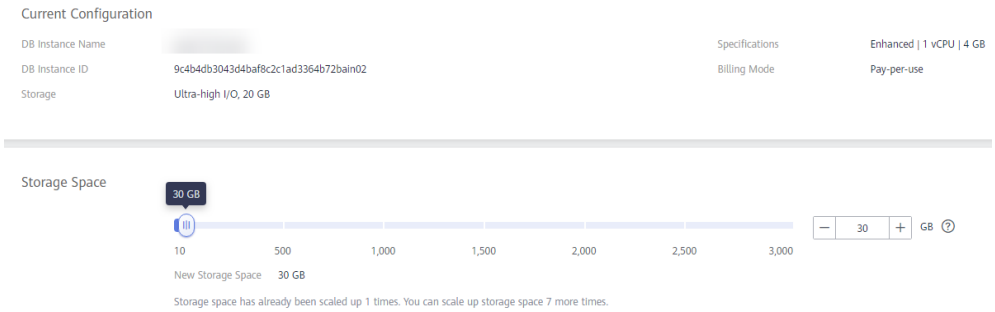
Replica Set

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target replica set instance and click **Scale Storage Space** in the **Operation** column.

Step 3 On the displayed page, specify the desired storage space, and click **Next**.

Figure 7-8 Scale Replica Set



You must add a minimum of 10 GB each time you scale up, and only multiples of 10 GB are allowed. The maximum amount of storage space is 3000 GB.

Step 4 On the displayed page, confirm the storage space.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify the specifications, click **Submit** to scale up the storage space.

Step 5 Check the scale-up result.

- This process takes about 3 to 5 minutes. The status of the DB instance in the instance list is **Scaling up**.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- In the **Storage Space** area on the **Basic Information** page, check whether the scaling up is successful.

----End

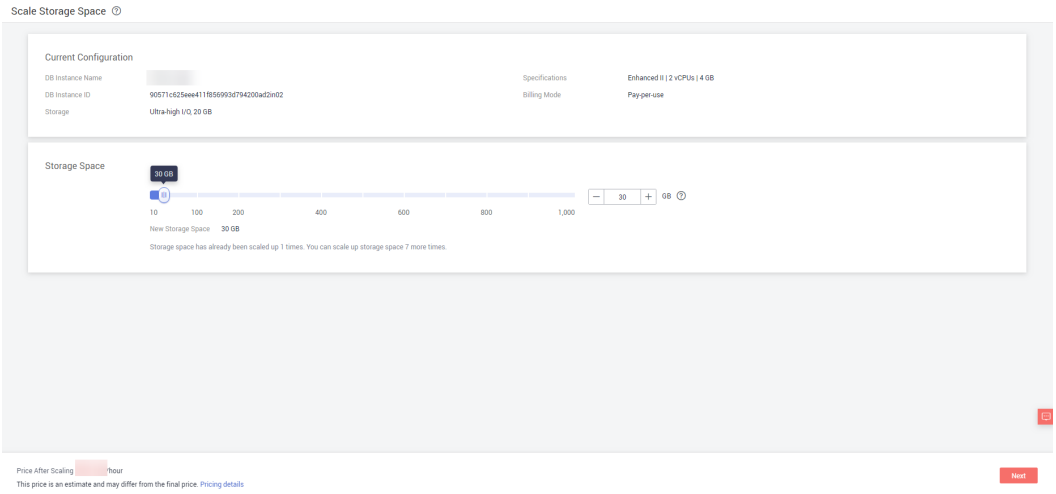
Single Node

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target single node instance and click **Scale Storage Space** in the **Operation** column.

Step 3 On the displayed page, specify the desired storage space and click **Next**.

Figure 7-9 Scale Single Node



You must add a minimum of 10 GB each time you scale up, and only multiples of 10 GB are allowed. The maximum amount of storage space is 1000 GB.

Step 4 On the displayed page, confirm the storage space.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify the specifications, click **Submit** to scale up the storage space.

Step 5 Check the scale-up result.

- This process takes about 3 to 5 minutes. The status of the DB instance in the instance list is **Scaling up**.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- In the **Storage Space** area on the **Basic Information** page, check whether the scaling up is successful.

----End

7.6 Changing the CPU or Memory of a Cluster DB Instance

Scenarios

This section describes how to change the CPU or memory of a cluster instance.

NOTE

- You can change the CPU and memory of a DB instance only when your account balance is greater than or equal to ¥0.
- A DB instance cannot be deleted when you are changing its CPU or memory.
- Instances can be scaled up or down.
- When the CPU and memory specifications are changed, a primary/secondary switchover may occur once or twice and the database connection will be interrupted for up to 30s. You are advised to change the specifications during off-peak hours to reduce impacts and ensure that the service system of your client can reconnect to the database if the connection is interrupted.

Precautions

- Instances in pay-per-use mode are still charged based on the time used after the instance CPU or memory is changed.
- If you change the CPU or memory of a yearly/monthly DB instance, you will pay price difference or get a refund.
  - If you scale up the instance class, you need to pay extra fee based on the period in which instances are put into use.
  - If you scale down the instance class, you will get a refund based on the period in which instances are put into use. The refund will be sent to your account. You can click **Billing Center** in the upper right corner of the console to view your account balance.

Changing mongos

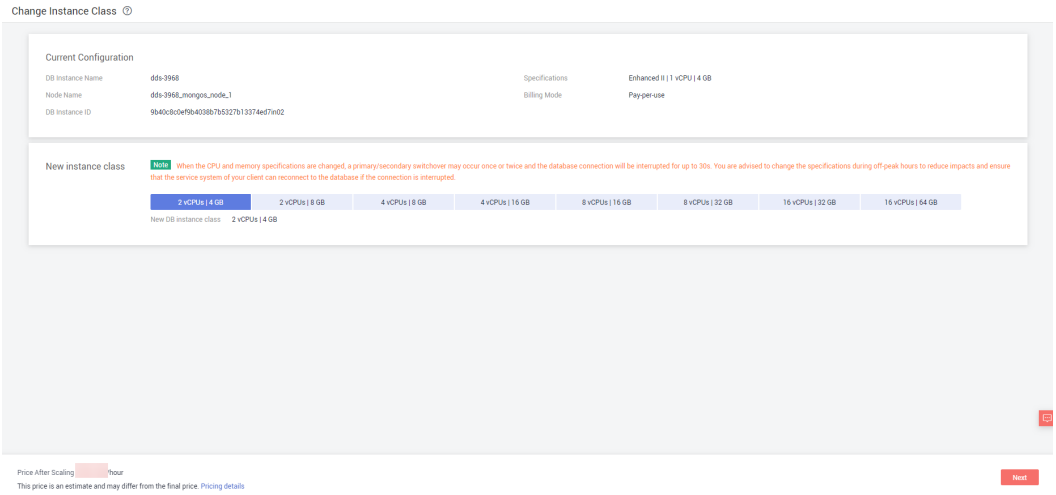
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click the **mongos** tab, locate the target mongos, and click **Change Instance Class** in the **Operation** column.

Step 4 On the displayed page, select the new instance class and click **Next**.

Figure 7-10 Scale mongos



Step 5 On the displayed page, confirm the instance class.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to change the instance class.

Step 6 View the DB instance class change result.

- When the CPU or memory of a DB instance is being changed, the status displayed in the **Status** column is **Changing instance class**. This process takes about 10 minutes.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- In the **Node Information** area on the **Basic Information** page, click the **mongos** tab and view the new instance class.

----End

Changing shard

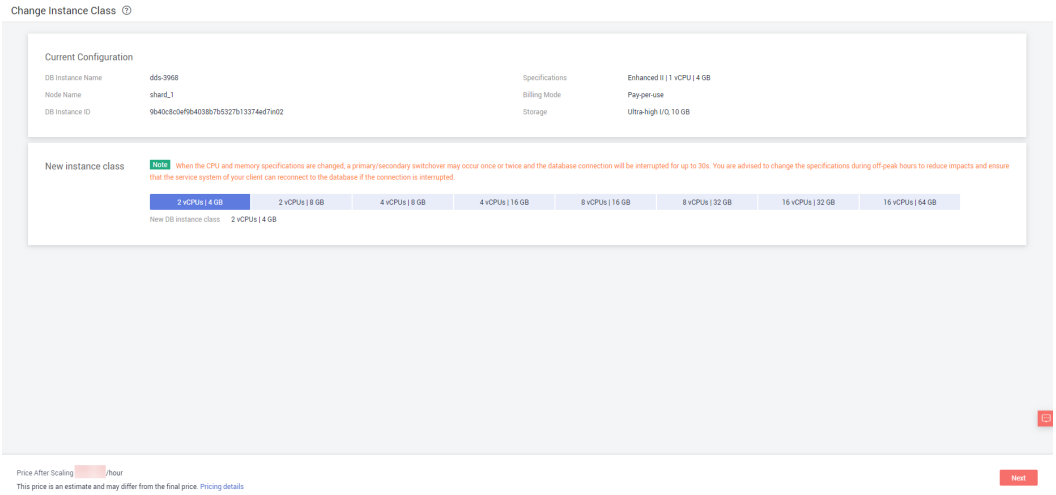
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click the **shard** tab, locate the target shard, and click **Change Instance Class** in the **Operation** column.

Step 4 On the displayed page, select the new instance class and click **Next**.

Figure 7-11 Scale shard



Step 5 On the displayed page, confirm the instance class.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to change the instance class.

Step 6 View the DB instance class change result.

- When the CPU or memory of a DB instance is being changed, the status displayed in the **Status** column is **Changing instance class**. This process takes about 25 to 30 minutes.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- Go to the **Basic Information** page of the cluster instance you scaled up, click the **shard** tab in the **Node Information** area, and view the new instance class.

----End

7.7 Changing the CPU or Memory of a Replica Set DB Instance

Scenarios

This section describes how to change the CPU or memory of your replica set instance.

NOTE

- You can change the CPU and memory of a DB instance only when your account balance is greater than or equal to ¥0.
- A DB instance cannot be deleted when you are changing its CPU or memory.
- Instances can be scaled up or down.
- When the CPU and memory specifications are changed, a primary/secondary switchover may occur once or twice and the database connection will be interrupted for up to 30s. You are advised to change the specifications during off-peak hours to reduce impacts and ensure that the service system of your client can reconnect to the database if the connection is interrupted.

Precautions

- Instances in pay-per-use mode are still charged based on the time used after the instance CPU or memory is changed.
- If you change the CPU or memory of a yearly/monthly DB instance, you will pay price difference or get a refund.
  - If you scale up the instance class, you need to pay extra fee based on the period in which instances are put into use.
  - If you scale down the instance class, you will get a refund based on the period in which instances are put into use. The refund will be sent to your account. You can click **Billing Center** in the upper right corner of the console to view your account balance.

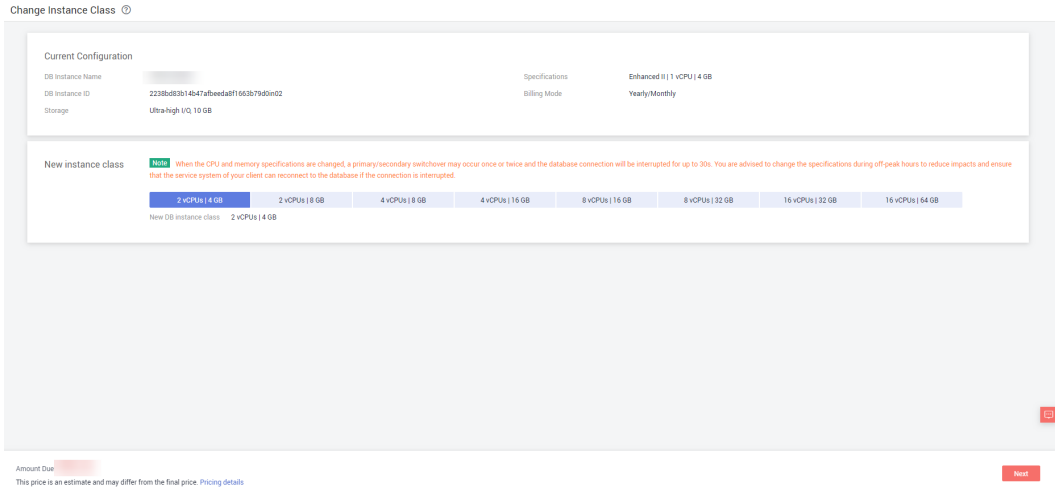
Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target replica set instance and choose **More** > **Change Instance Class** in the **Operation** column.

Step 3 On the displayed page, modify required parameters and click **Next**.

Figure 7-12 Scale Replica Set



Step 4 On the displayed page, confirm the instance class.

- Yearly/Monthly
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to change the instance class.

Step 5 View the DB instance class change result.

- When the CPU or memory of a DB instance is being changed, the status displayed in the **Status** column is **Changing instance class**. This process takes about 25 to 30 minutes.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- Go to the **Basic Information** page of the replica set instance you scaled up and check whether the scaling up is successful in the **DB Information** area.

----End

7.8 Changing the CPU or Memory of a Single Node DB Instance

Scenarios

This section describes how to change the CPU or memory of your single node instance.

NOTE

- You can change the CPU and memory of a DB instance only when your account balance is greater than or equal to ¥0.
- A DB instance cannot be deleted when you are changing its CPU or memory.
- Instances can be scaled up or down.
- Services will be interrupted for 5 to 10 minutes when you change DB instance CPU and memory. You are advised to perform this operation during off-peak hours. After the restart is complete, the cached memory will be automatically cleared. The DB instance needs to be warmed up to prevent congestion during peak hours.

Precautions

- Instances in pay-per-use mode are still charged based on the time used after the instance CPU or memory is changed.
- If you change the CPU or memory of a yearly/monthly DB instance, you will pay price difference or get a refund.
  - If you scale up the instance class, you need to pay extra fee based on the period in which instances are put into use.
  - If you scale down the instance class, you will get a refund based on the period in which instances are put into use. The refund will be sent to your account. You can click **Billing Center** in the upper right corner of the console to view your account balance.

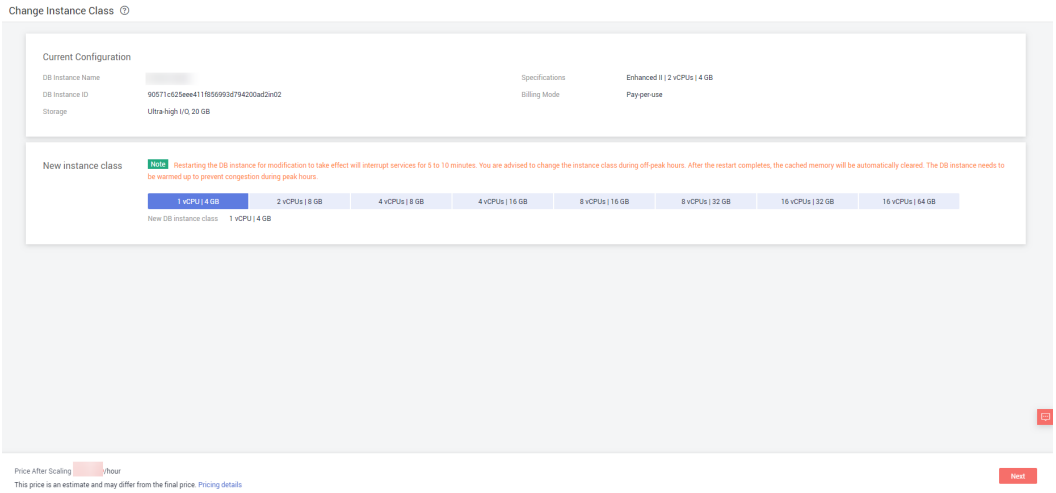
Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target single node instance and choose **More** > **Change Instance Class** in the **Operation** column.

Step 3 On the displayed page, modify required parameters and click **Next**.

Figure 7-13 Scale Single Node



Step 4 On the displayed page, confirm the instance class.

- Yearly/Monthly

  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous** to go back to the page for you to specify details.
  - If you do not need to modify your settings, click **Submit** to change the instance class.

Step 5 View the DB instance class change result.

- When the CPU or memory of a DB instance is being changed, the status displayed in the **Status** column is **Changing instance class**. This process takes about 10 minutes.
- In the upper right corner of the DB instance list, click  to refresh the list. The instance status changes to **Available**.
- Go to the **Basic Information** page of the single node you scaled up and check whether the scaling process is successful in the **Configuration** area.

----End

7.9 Manually Switching the Primary and Secondary Nodes of a Replica Set

Scenarios

A replica set consists of the primary node, secondary node, and hidden node. Primary and secondary nodes allow access from external services by providing IP addresses. Hidden nodes are only used for backing up data. When a primary node becomes faulty, the system automatically selects a new primary node to ensure high availability. In addition, DDS supports the primary/secondary switchover so you can perform switchovers in scenarios such as disaster recovery.

NOTE

- You can perform a switchover when the DB instance status is **Available**, **Changing to yearly/monthly**, and **Changing a security group**.
- The database connection may be interrupted during the switchover. Ensure that your client supports reconnection.
- The longer the delay for primary/secondary synchronization, the more time is needed for a primary/secondary switchover. Therefore, if the primary to secondary synchronization delay exceeds 300s, the primary/secondary switchover is not allowed. For details about the synchronization delay, see **What Is the Time Delay for Primary/Secondary Synchronization in a Replica Set?**

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target replica set instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click **Switch**.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 Check the result.

- During the switchover process, the DB instance status changes to **Switchover in progress**. After the switchover is complete, the status is restored to **Available**.
- In the **Node Information** area, you can view the switchover result.
- After the switchover, the previous primary node becomes the secondary node. You need to reconnect to the primary node. For details, see **Connecting to a DB Instance**.

----End

7.10 Exporting DB Instance Information

Scenarios

This section describes how to export DB instance information for analysis.

Exporting DB Instance Information

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click in the upper right corner of the instance list.

Step 3 In the pop-up box, select the desired items and click **OK**.

Step 4 View the .xls file exported to your local PC.

----End

Export Specified Instance

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, select the target DB instance and click in the upper right corner of the instance list.

Step 3 In the pop-up box, select the desired items and click **OK**.

Step 4 View the .xls file exported to your local PC.

----End

7.11 Restarting a DB Instance or a Node

Scenarios

You may need to occasionally restart a DB instance to perform routine maintenance. For example, after modifying certain parameters, you must restart the DB instance for the modifications to take effect on the management console.

You can restart a DB instance only when its status is **Available**.

NOTICE

- Restarting a DB instance interrupts services, so you should exercise caution when performing this operation.
- If you restart a DB instance, all nodes in the instance are also restarted.

Background

If you enable operation protection to improve the security of your account and cloud products, two-factor authentication is required for sensitive operations. For details about how to enable operation protection, see **Critical Operations** in the *Identity and Access Management User Guide*.

Restarting a DB Instance

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and in the **Operation** column, choose **More** > **Restart**.

Alternatively, click the target DB instance and on the displayed **Basic Information** page, click **Restart** in the upper right corner of the page.

Step 3 If you have enabled operation protection, click **Start Verification** in the **Restart DB Instance** dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 View the restart status.

1. On the **Instance Management** page, the instance status is **Restarting**.
2. On the **Basic Information** page, all nodes of the cluster instance cannot be restarted.

----End

Restarting a Node (Cluster)

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click the **mongos**, **shard**, or **config** tab, locate the target node, and in the **Operation** column, click **Restart** or choose **More** > **Restart**.

Step 4 In the displayed dialog box, click **Yes**.

Step 5 View the node status.

When one node status is **Restarting**, other nodes of the instance cannot be restarted.

----End

7.12 Deleting a Pay-per-Use DB Instance

Scenarios

To delete a DB instance billed in the pay-per-use mode, you need to locate the target DB instance and click **Delete** on the **Instance Management** page.

(To delete a DB instance billed in the yearly/monthly mode, you need to unsubscribe from the order. For details, see **Unsubscribing from a Yearly/Monthly DB Instance**.)

- Cluster instance
- Replica set instance
- Single node instance

NOTICE

- Yearly/Monthly DB instances cannot be deleted.
- After you delete an instance, all nodes in the instance are also deleted.
- After you delete the DB instance, all data in it and all automated backups are automatically deleted and cannot be restored. Exercise caution when performing this operation.
- By default, all manual backups are retained in DDS. You can use a backup to restore a deleted instance.

Background

If you enable operation protection to improve the security of your account and cloud products, two-factor authentication is required for sensitive operations. For details about how to enable operation protection, see **Critical Operations** in the *Identity and Access Management User Guide*.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target DB instance and choose **More** > **Delete** in the **Operation** column.

Step 3 If you have enabled operation protection, click **Start Verification** in the **Delete DB Instance** dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

Step 4 In the displayed dialog box, click **Yes**.

----End

7.13 Recycling a DB Instance

DDS can move unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances to the recycle bin, so you can rebuild the DB instance and restore data from it.

The recycling policy is enabled by default and cannot be disabled. DB instances in the recycle bin are retained for 1 day by default, and this will not incur any charges.

Currently, you can put up to 100 instances into the recycle bin. If the maximum number of instances is reached, you cannot put instances into the recycle bin any more.

Modifying the Recycling Policy

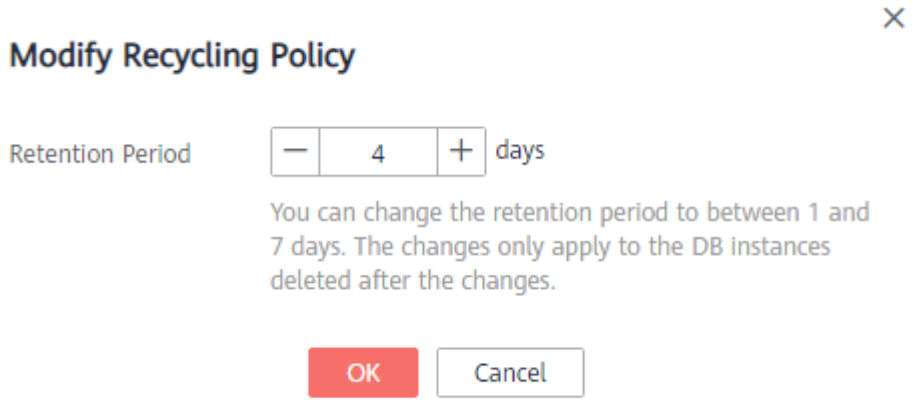
NOTICE

You can modify the retention period, and the changes only apply to the DB instances deleted after the changes, so exercise caution when performing this operation.

Step 1 **Log in to the DDS console.**

Step 2 On the **Recycling Management** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances from 1 day to 7 days. Then, click **OK**.

Figure 7-14 Modifying the recycling policy



----End

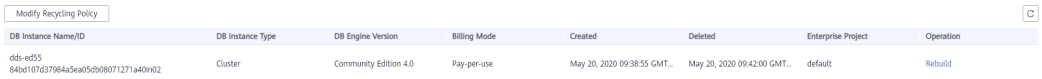
Rebuilding a DB Instance

You can rebuild DB instances from the recycle bin within the retention period to restore data.

Step 1 **Log in to the DDS console.**

Step 2 On the **Recycling Management** page, locate the DB instance to be rebuilt and in the **Operation** column, click **Rebuild**.

Figure 7-15 Rebuilding a DB instance



Step 3 On the displayed page, set required parameters and submit the rebuilding task.

----End

8 Backup and Restore

8.1 Overview

DDS supports DB instance backups and restorations to ensure data reliability.

Automated Backup

Automated backups are created during the backup time window of your DB instances. DDS saves automated backups based on the backup retention period you specified. If necessary, you can restore data to any point in time during the backup retention period.

Manual Backup

Manual backups are user-initiated full backups of DB instances. They are retained until you delete them manually.

Full Backup

Full backup is to back up all selected data, even if no data is updated since the last backup.

Incremental Backup

DDS automatically backs up data updated since the last automated or incremental backup every five minutes.

Restoration Task Historical Records

You can view the task execution time and historical records on the **Task Center** page. For details, see **Tasks Overview**.

Backup Space Billing

DDS backups are stored in OBS buckets. For details about the billing policy, see **Pricing Details**.

8.2 Setting Automated Backup Policy

Scenarios

DDS backs up data automatically based on the automated backup policy you set. You are advised to regularly back up data in your database. If the database becomes faulty or data is damaged, you can restore it with the backup, ensuring data reliability.

Precautions

- DDS checks existing automated backup files. If the retention period of a file exceeds the backup retention period you set, DDS will delete the file.
- After the backup policy is modified, an automated backup will be triggered based on the new backup policy. The retention period of the previously generated automated backups remains unchanged.
- Backup files are stored in OBS buckets.
- When a DB instance is created, DDS enables the automated backup policy by default. The default settings of the parameters are as follows. You can modify them after a DB instance is created.
  - Backups are retained for 7 days by default.
  - The time window is in UTC by default.
  - Data is backed up every day by default.
- Set the backup window when the local computer time is not in the GMT+8 time zone.

  Case analysis: The default time zone is GMT+08:00. If your local computer time is not in the GMT+8 time zone and the backup window of each day is set to 00:00-01:00, the actual backup time is not 00:00-01:00.

  Solution: Change the time zone of your local computer to GMT+8, and set the backup window to 00:00-01:00 of GMT+8.

Enabling or Modifying an Automated Backup Policy

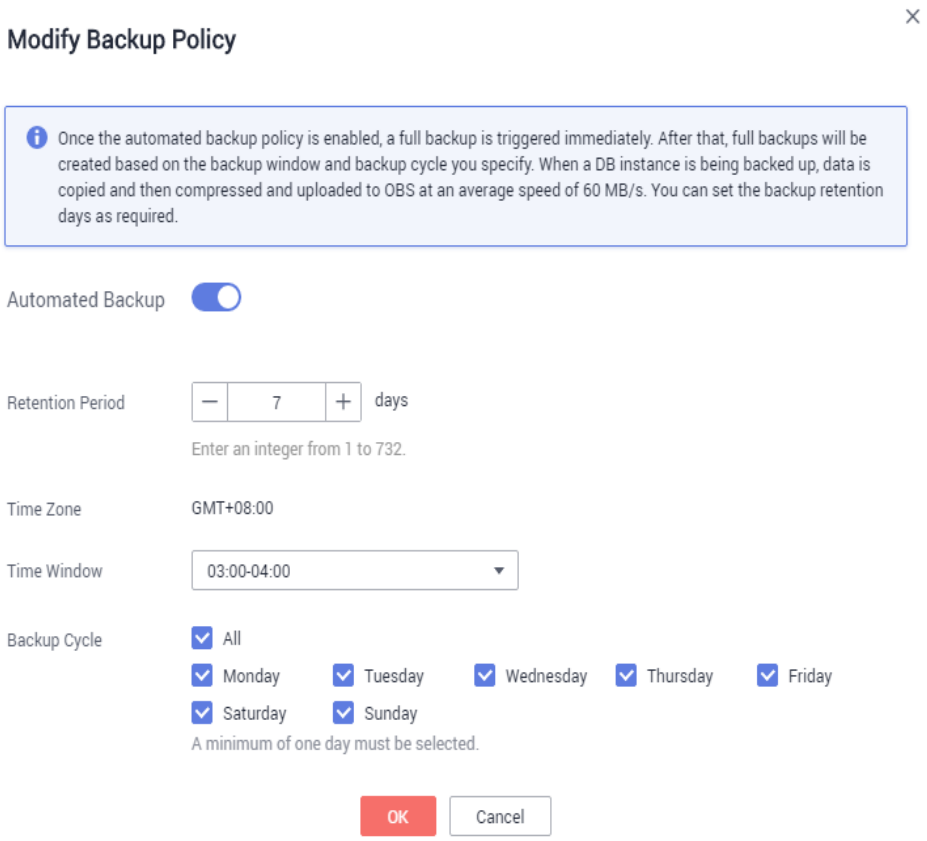
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, click **Modify Backup Policy**. If you want to enable the automated backup policy, click  . Once enabled, the backup policy can be modified as shown in **Figure 8-1**.

Figure 8-1 Modify Backup Policy



Retention Period refers to the number of days that data is kept. You can increase the retention period to improve data reliability.

The backup retention period can range from 1 to 732 days, with a time window of one hour. The backup cycle varies according to the retention period you have set.

- If you set the retention period to 1 to 6 days, data is automatically backed up each day of the week.
- If you set the retention period to 7 to 732 days, you must select at least one day of the week for the backup cycle.

Step 5 Click **OK** to save the modification.

Step 6 View the backup result.

- If the automated backup policy is enabled, an automated full backup is immediately triggered. The time it takes to complete the backup depends on the size of the job.
- If the automated backup policy is modified, an automated full backup is randomly triggered during the time window you set. The time it takes to complete the backup depends on the size of the job.
- During the creation of an automated backup, you can query the backup status on the **Backup Management** page or the **Backups & Restorations** tab. The backup status is **Backing up**.

- In the upper right corner of the backup list, click  to refresh the list. The backup status changes to **Complete**. The backup type is **Automated** and the backup method is **Physical**.

----End

Disabling an Automated Backup Policy

NOTICE

Observe the following constraints when disabling the automated backup policy:

- Your data cannot be backed up.
- All the existing incremental backups on the DB instance will be deleted. Your replica set instances cannot be restored to a specified point in time.
- If you choose to delete all the existing automated backup when disabling the automated backup policy, related restoration or download operations will fail.

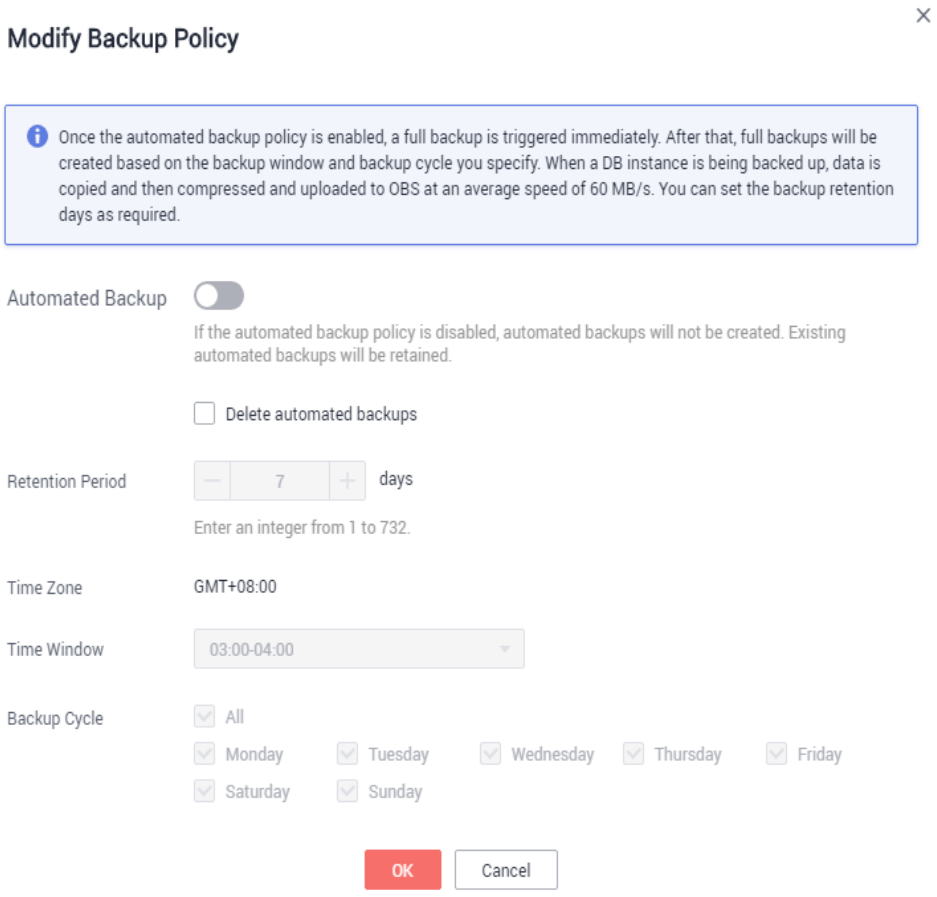
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, click **Modify Backup Policy**. On the displayed page, click  to disable the automated backup policy. **Figure 8-2** shows the dialog box for modifying the backup policy.

Figure 8-2 Modify Backup Policy



You can determine whether to delete all automated backup files:

- If you do not select **Delete automated backups**, all backup files within the retention period will be retained. You can manually delete them. For details, see section **Deleting an Automated Backup**.
- If you select **Delete automated backups**, all backup files within the retention period will be deleted.

Step 5 Click **OK**.

----End

8.3 Creating a Manual Backup

Scenarios

This section describes how to create a manual backup. Creating a backup for a DB instance helps ensure data can be restored if needed, ensuring data reliability.

NOTE

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

You can create manual backups only when your account balance is more than ¥0.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate an available DB instance and click **Create Backup** or choose **More** > **Create Backup**.

Step 3 In the displayed dialog box, specify **Backup Name** and **Description** and click **OK**.

- The manual backup name can be 4 to 64 characters long. It must start with a letter and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 4 Check the result:

- During the creation of a manual backup, you can query the backup status on the **Backup Management** or the **Backups & Restorations** page. The backup status is **Backing up**. The time it takes to complete the backup depends on the size of the job.
- If the manual backup is successfully created, the backup status is **Complete**. The manual backup type is **Manual** and the backup method is **Physical**.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, click **Create Backup**.

Step 4 In the displayed dialog box, specify **DB Instance Type**, **DB Instance Name**, **Backup Name** and **Description** and click **OK**.

- Only DB instances in **Available** status can be manually backed up.
- The manual backup name can be 4 to 64 characters long. It must start with a letter and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 5 Check the result:

- During the creation of a manual backup, you can query the backup status on the **Backup Management** or the **Backups & Restorations** page. The backup status is **Backing up**. The time it takes to complete the backup depends on the size of the job.

- If the manual backup is successfully created, the backup status is **Complete**. The manual backup type is **Manual** and the backup method is **Physical**.

----End

Method 3

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click an available DB instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, click **Create Backup**.

Step 5 In the displayed dialog box, specify **Backup Name** and **Description** and click **OK**.

- The manual backup name can be 4 to 64 characters long. It must start with a letter and can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 6 Check the result:

- During the creation of a manual backup, you can query the backup status on the **Backup Management** or the **Backups & Restorations** page. The backup status is **Backing up**. The time it takes to complete the backup depends on the size of the job.
- If the manual backup is successfully created, the backup status is **Complete**. The manual backup type is **Manual** and the backup method is **Physical**.

----End

8.4 Restoring a Cluster Instance from a Backup

Scenarios

This section describes how to restore a cluster instance from a backup.

Restoration Precautions

- Currently, DDS cluster instances of Community Edition can be restored to a new DB instance or the original DB instance.
- When you restore the DB instance from a backup file, a full backup file is downloaded from OBS and then restored to the DB instance at an average speed of 40 MB/s.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, locate the target backup and click **Restore** in the **Operation** column.

Step 5 Select either of the following restoration methods and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, storage type, and shard quantity must be the same as those of the original and cannot be changed.
  - The number of mongos nodes is 2 by default and ranges from 2 to 32. You can specify the quantity.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Cluster Instance (Community Edition)**.

- Restore to Original

  NOTICE

  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
  - The administrator password of the database instance remains unchanged after the restoration.

  Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, locate the target backup on the **Clusters** tab and click **Restore** in the **Operation** column.

- If you use an automated backup, go to **Step 4**.
- If you use a manual backup, check whether the original instance of the manual backup exists:
  - If yes, go to **Step 4**.
  - If no, you can only restore the backup to a new DB instance. Go to **Create New Instance**.

Step 4 Select either of the following restoration methods and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, storage type, and shard quantity must be the same as those of the original and cannot be changed.
  - The number of mongos nodes is 2 by default and ranges from 2 to 32. You can specify the quantity.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Cluster Instance (Community Edition)**.

- Restore to Original

  NOTICE

  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
  - The administrator password of the database instance remains unchanged after the restoration.

  Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

8.5 Restoring a Replica Set Instance from a Backup

Scenarios

This section describes how to restore a replica set instance from a backup.

Restoration Precautions

- Currently, you can restore a replica set instance to a new or original DB instance.
- When you restore the DB instance from a backup file, a full backup file is downloaded from OBS and then restored to the DB instance at an average speed of 40 MB/s.

Method 1

Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target replica set instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, locate the target backup and click **Restore** in the **Operation** column.

Step 5 Select either of the following restoration methods and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Replica Set DB Instance**.

- Restore to Original

  NOTICE

  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
  - The administrator password of the database instance remains unchanged after the restoration.
  - If the backup method is logical backup, the backup cannot be restored to the original instance.

  Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

Method 2

Step 1 Log in to the DDS console.

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, locate the target backup on the **Replica Sets** tab and click **Restore** in the **Operation** column.

- If you use an automated backup, go to **Step 4**.
- If you use a manual backup, check whether the original instance of the manual backup exists:
  - If yes, go to **Step 4**.
  - If no, you can only restore the backup to a new DB instance. Go to **Create New Instance**.

Step 4 Select either of the following restoration methods and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Replica Set DB Instance**.

- Restore to Original

  NOTICE

  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
  - The administrator password of the database instance remains unchanged after the restoration.
  - If the backup method is logical backup, the backup cannot be restored to the original instance.

  Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

8.6 Restoring Replica Set Instance to a Point in Time

Scenarios

You can restore the data of a replica set instance to a specified time point.

Restoration Precautions

- Currently, you can restore a replica set instance to a new or original DB instance at a point in time.
- The local database is not included in the databases that can be restored to a specified time point.
- When you enter the time point that you want to restore the DB instance to, DDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Constraints

You can use backup files to create a new DB instance only when your account balance is greater than or equal to ¥0.

Data can be restored to a specified time point only after the automated backup policy is enabled.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target replica set instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, click **Restore to Point In Time**.

Step 5 Select the date and time range, select or enter a time point within the acceptable range, and select **Create New Instance** or **Restore to Original**.

Step 6 On the displayed page, the DB instance is restored based on the restoration method you selected in **Step 5**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Replica Set Instance**.

- Restore to Original

NOTICE

- Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
- The administrator password of the database instance remains unchanged after the restoration.
- If the backup method is logical backup, the backup cannot be restored to the original instance.

Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

8.7 Restoring Replica Set Database and Table to a Point in Time

Scenarios

To ensure data integrity and reduce impact on the original instance performance, the system restores the full and incremental data at the selected time point to a temporary DB instance, automatically exports the databases and tables to be restored, and then restores the databases and tables to the original DB instance. The time required depends on the amount of data to be backed up and restored on the DB instance. Please wait.

Restoring databases and tables will not overwrite data in the DB instance. You can select databases and tables to be restored.

Restoration Precautions

- After a successful restoration, a new table named ***Original table name*_bak_*Timestamp*** is generated in the DB instance by default. If the table contains an index, the namespace of the index is changed to ***Original database name.Original table name*_bak_*Timestamp***. You can rename the table later as required.
- New databases and tables will be generated in the original DB instance. Ensure that sufficient storage space is available.
- The length of *Database name.Table name* is no more than 120 characters. The length of *Database name.Table name.Index name* is no more than 128 characters. To avoid restoration failure, ensure that the name length meets the requirements.
- Ensure that the name of the restored table is different from that of the existing table. Otherwise, the restoration may fail.
- If you perform a table-level restore and the table does not exist at the required time point, an empty table is automatically created. If you perform a database-level restore, the empty table is not created.

Constraints

Currently, only DDS Community Editions 3.2 and 3.4 support the point-in-time recovery at the database and table level for replica set instances.

Before performing the restoration, you need enable the automated backup policy.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target replica set instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, click **Restore Database and Table**.

Step 5 In the displayed dialog box, configure parameters as required.

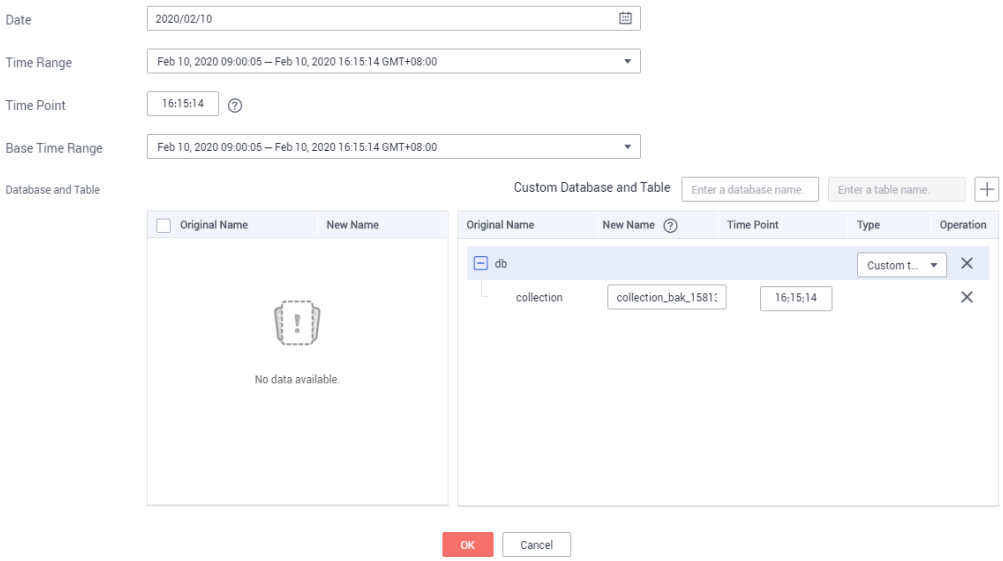
Table 8-1 Database information

Parameter	Description
Date	Date when the automated backup of the DB instance is generated.
Time Range	Time range during which the automated backup can be restored.
Time Point	Time point when the automated full backup is generated in the specified time range.
Base Time Range	Time range during which the database and table can be restored based on the automated full backup.
Database and Table	Databases and tables that have been automatically backed up in the base time range are displayed in the left area. Select the databases and tables on the left to sync information to the area on the right.
Time Point	Time point in the base time range.

Parameter	Description
Custom Database and Table	You can add custom databases and tables as required. • The system database cannot be restored. Therefore, the database name cannot be **admin** or **local**. • The database name cannot contain spaces and the following special characters: ".\/$ • The table name cannot contain the dollar sign ($) or "system." in prefix. • The length of *Database name.Table name* is no more than 120 characters. The length of *Database name.Table name.Index name* is no more than 128 characters. To avoid restoration failure, ensure that the name length meets the requirements. • Ensure that the name of the restored table is different from that of the existing table. Otherwise, the restoration may fail. • After a successful restoration, a new table named ***Original table name*_bak_*Timestamp*** is generated in the DB instance by default. If the table contains an index, the namespace of the index is changed to ***Original database name.Original table name*_bak_*Timestamp***. You can rename the table later as required. Distinct the time point of the custom databases and tables from those synchronized to the right area. You are advised to set the time point to different value. The system restores data to the custom databases and tables based on the time point.
Type	You can restore data to a database or table. If you perform a table-level restore and the table does not exist at the required time point, an empty table is automatically created. If you perform a database-level restore, data will be restored to the database separately, and the table will not be created.

Click **OK** to start the restoration. The data in the new database and table is the same as that in the database and table at the selected time point.

Figure 8-3 Selecting database and table



Step 6 On the **Instance Management** page, the DB instance status is **Restoring**. During the restoration process, services are not interrupted.

Step 7 After the restoration is successful, manage data in the database and table as required.

If you need to use the original database and table names, you can perform the rename operation to back up the original database and table and switch your service to the restored database and table. Then, delete the original database and table after ensuring that your services are normal.

Example:

db.adminCommand({renameCollection: "db1.test1", to: "db2.test2"})

The above command is used to move the **test1** table from the **db1** database to the **db2** database and rename the table to **test2**.

----End

8.8 Restoring Replica Set Instance to a Local Self-Built Database

Scenarios

This section uses the Linux operating system as an example to describe how to restore the downloaded backup files of a replica set instance to your self-built databases. To download backup files, see **Downloading Backup Files**.

NOTE

This method applies only to replica set instances.

Prerequisites

The client tool of version 3.4 has been installed on your local self-built MongoDB database.

Procedure

Step 1 Log in to the server on which self-built databases are deployed.

Assume that **/path/to/mongo** is the directory for restoration, and **/path/to/mongo/data** is the directory for storing the backup file.

Step 2 Before the restoration, ensure that the **/path/to/mongo/data** directory is empty.

cd /path/to/mongo/data/

rm -rf *

Step 3 Copy and paste the downloaded backup file package to **/path/to/mongo/data/** and decompress it.

lz4 -d xxx_.tar.gz |tar -xC /path/to/mongo/data/

Step 4 Create the **mongod.conf** configuration file in **/path/to/mongo**.

touch mongod.conf

Step 5 Start the database in single-node mode.

1. Modify the **mongod.conf** file to meet the backup startup configuration requirements.

   The following is a configuration template for backup startup:

```
systemLog:
    destination: file
    path: /path/to/mongo/mongod.log
    logAppend: true
security:
    authorization: enabled
storage:
    dbPath: /path/to/mongo/data
    directoryPerDB: true
    engine: wiredTiger
    wiredTiger:
        collectionConfig: {blockCompressor: snappy}
        engineConfig: {directoryForIndexes: true, journalCompressor: snappy}
        indexConfig: {prefixCompression: true}
net:
    http:
        enabled: false
    port: 27017
    bindIp: xxx.xxx.xxx.xxx,xxx.xxx.xxx.xxx
    unixDomainSocket:
        enabled: false
processManagement:
    fork: true
    pidFilePath: /path/to/mongo/mongod.pid
```

   NOTE

   bindIp indicates the IP address bound to the database. This field is optional. If it is not specified, your local IP address is bound by default.

2. Run the **mongod.conf** command to start the database.

   */usr/bin/***mongod -f /path/to/mongo/mongod.conf**

   NOTE

   /usr/bin/ is the directory that stores the **mongod** file of the installed MongoDB client.

3. After the database is started, log in to the database using mongo shell to verify the restoration result.

   mongo --host *<DB_HOST>* **-u** *<DB_USER>* **-p** *<PASSWORD>* **--authenticationDatabase admin**

   NOTE

   - **DB_HOST** indicates the IP address bound to the database.
   - **DB_USER** indicates the username of the database account. The default value is **rwuser**.
   - **PASSWORD** indicates the password of the database account, which is the password used for backing up the DB instance.

----End

Starting the Database in Replica Set Mode

By default, the physical backup of the DDS DB instance contains the replica set configuration of the original DB instance. You need to start the database in single-node mode. Otherwise, the database cannot be accessed.

If you want to start the database in replica set mode, perform step **Step 5** and then perform the following steps:

Step 1 Log in to the database using mongo shell.

Step 2 Remove the original replica set configuration.

use local

db.system.replset.remove({})

Step 3 Stop the database process.

use admin

db.shutdownServer()

Step 4 Add the replication configuration in the **mongod.conf** file in the **/path/to/mongo/** directory. For details about the command usage, see **Deploy a Replica Set**.

Step 5 Run the **mongod.conf** command to start the database.

*/usr/bin/***mongod -f /path/to/mongo/mongod.conf**

NOTE

/usr/bin/ is the directory that stores the **mongod** file of the installed MongoDB client.

Step 6 Add the replica set members and initialize the replica set.

NOTE

Use the rs.initiate() command to perform the preceding step. For details, see **rs.initiate()**.

----End

8.9 Restoring a Single Node Instance from a Backup

Scenarios

This section describes how to restore a single node instance from a backup.

Restoration Precautions

- Currently, a single node instance can be restored only to a new or original instance.
- When you restore the DB instance from a backup file, a full backup file is downloaded from OBS and then restored to the DB instance at an average speed of 40 MB/s.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target single node instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, locate the target backup and click **Restore** in the **Operation** column.

Step 5 In the displayed window, select a restoration mode and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.
  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Single Node Instance**.

- Restore to Original

NOTICE

- Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
- The administrator password of the database instance remains unchanged after the restoration.
- If the backup method is logical backup, the backup cannot be restored to the original instance.

Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

Step 6 On the displayed page, create a DB instance using the same configurations as the backup. The new DB instance is independent from the original one.

- You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
- The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
- The storage space is the same as that of the original instance by default. You can only increase the storage space.
- Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Single Node Instance**.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, locate the target backup on the **Single Nodes** tab and click **Restore** in the **Operation** column.

Step 4 In the displayed window, select a restoration mode and click **OK**.

- Create New Instance

  The **Create New Instance** page is displayed for you to create a DB instance using the backup data. The new DB instance is independent from the original one.

  - You are recommended to deploy the restored DB instance in a different AZ to ensure that applications will not be adversely affected by the failure in any single AZ.
  - The database type, DB instance type, compatible MongoDB version, storage engine, and storage type must be the same as those of the original and cannot be changed.
  - The storage space is the same as that of the original instance by default. You can only increase the storage space.

  - Other settings are the same as those of the original DB instance by default and can be modified. For details, see **Buying a Single Node Instance**.

- Restore to Original

  NOTICE

  - Restoring to the original DB instance will overwrite all existing data and the DB instance will be temporarily unavailable during the restoration.
  - The administrator password of the database instance remains unchanged after the restoration.
  - If the backup method is logical backup, the backup cannot be restored to the original instance.

  Check that the status of the DB instance on the **Instance Management** page is **Restoring**.

----End

8.10 Restoring Single Node Instance to a Local Self-Built Database

Scenarios

This section uses the Linux operating system as an example to describe how to restore the downloaded backup files of a single node instance to your self-built databases. To download backup files, see **Downloading Backup Files**.

NOTE

This method applies only to single node instances.

Prerequisites

The client tool of version 3.4 has been installed on your local self-built MongoDB database.

Procedure

Step 1 **Download the backup file of the single node**.

Step 2 Log in to the device that can access the self-built database.

Step 3 Upload the single-node backup file to the device that can access the self-built database.

Select an uploading method based on the OS you are using. In Linux, for example, run the following command:

scp -r *<IDENTITY_DIR> <REMOTE_USER>@<REMOTE_ADDRESS>:<REMOTE_DIR>*

- **IDENTITY_DIR** indicates the directory that stores the backup file.
- **REMOTE_USER** indicates the username for logging in to the device that can access the self-built database.
- **REMOTE_ADDRESS** indicates the IP address of the host that can access the self-built database.
- **REMOTE_DIR** indicates the destination directory to which the backup file is imported.

In Windows, upload the backup file using file transfer tools.

Step 4 Import the backup files in the self-built database.

./mongorestore --host *<DB_HOST>* **--port** *<DB_PORT>* **-u** *<DB_USER>* **--authenticationDatabase** *<AUTH_DB>* **--drop --gzip --archive=***<Backup directory>* **-vvvv --stopOnError**

- **DB_HOST** indicates the self-built database address.
- **DB_PORT** indicates the self-built database port.
- **DB_USER** indicates the self-built database username.
- **AUTH_DB** indicates the database that authenticates DB_USER. Generally, this value is **admin**.
- **Backup directory** indicates the backup file name.

Enter the self-built database account password when prompted:

```
Enter password:
```

Example:

./mongorestore --host 192.168.6.187 --port 8635 -u rwuser --authenticationDatabase admin --drop --gzip --archive=xxx_tar.gz -vvvv --stopOnError

----End

8.11 Downloading Backup Files

Scenarios

This section describes how to download manual or automated backup files for local data backup or restoration.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, locate the available backup you want to download and click **Download** in the **Operation** column.

Step 4 Download and install the client. For details, see **Downloading OBS Browser**.

Step 5 Log in to the OBS Browser.

For details on how to log in to OBS Browser, see section **Logging In to OBS Browser** in the *Object Storage Service Tools Guide*.

Step 6 Disable certificate verification on OBS Browser.

For details on how to configure OBS Browser, see section **Configuring the System** in the *Object Storage Service Tools Guide*.

NOTE

The OBS bucket names displayed on the **Download Backup File** page on the DDS console do not support certificate verification. Therefore, you need to disable OBS Browser certificate verification before adding external buckets and enable it after the backup files are downloaded.

Step 7 Add an external bucket.

In the **Create Bucket** dialog box of OBS Browser, select **Add external bucket** and enter the bucket name displayed on **Download Backup File** of the DDS console.

For details about how to add external buckets, see section **Adding External Buckets** in the *Object Storage Service Tools Guide*.

Step 8 Download the backup files.

In the search box on the right of OBS Browser, enter the backup file name displayed on **Download Backup File** of the DDS console.

Step 9 Enable OBS Browser certificate verification after the backup files are downloaded.

----End

8.12 Deleting a Manual Backup

Scenarios

This section describes how to delete manual backups to release the storage space.

NOTICE

The deletion operation is irreversible. Exercise caution when performing this operation.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** page, locate the manual backup to be deleted and click **Delete**.

Backups being used to recover instances cannot be deleted.

Step 5 In the displayed dialog box, click **Yes**.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup Management** page, locate the manual backup to be deleted and click **Delete** in the **Operation** column.

Backups being used to recover instances cannot be deleted.

Step 4 In the displayed dialog box, click **Yes**.

----End

8.13 Deleting an Automated Backup

Scenarios

This section describes how to delete an automated backup. If the automated backup policy is disabled, DDS allows you to delete stored automated backups to release storage space.

If the automated backup policy is enabled, DDS will delete automated backups as they expire. You cannot delete them manually.

NOTICE

The deletion operation is irreversible. Exercise caution when performing this operation.

Method 1

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Backups & Restorations**.

Step 4 On the **Backups & Restorations** tab, locate the automated backup to be deleted and click **Delete**.

Backups being used to recover instances cannot be deleted.

Step 5 In the displayed dialog box, click **Yes**.

----End

Method 2

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Backup Management**.

Step 3 On the **Backup & Restorations** page, locate the automated backup to be deleted and click **Delete** in the **Operation** column.

Backups being used to recover instances cannot be deleted.

Step 4 In the displayed dialog box, click **Yes**.

----End

9 Parameter Group Settings

9.1 What Is a Parameter Group?

DB parameter groups act as a container for engine configuration values that are applied to one or more DB instances. You can customize the parameter settings to manage DB engine configurations.

Type

When creating a DB instance, you can associate a default parameter group or a customized parameter group with the DB instance. After a DB instance is created, you can also change the associated parameter group.

- Default parameter groups

  The DB engine parameter values and system service parameter values in the default parameter group are designed for optimizing the database performance.
- Custom parameter groups

  If you need a DB instance with customized parameter settings, you can create a parameter group and change the parameter values as required.

  If you change the parameter values of the parameter group associated with several DB instances, the changes will apply to all these DB instances.

Application Scenarios

- If you want to use a customized parameter group, you only need to create a parameter group in advance and select the parameter group when creating a DB instance. For details about how to create a parameter group, see **Creating a Parameter Group**.
- If you need to change the associated parameter group, see **Changing Associated Parameter Group**.
- When you have already created a parameter group and want to include most of the custom parameters and values from that group in a new parameter group, you can replicate that parameter group following the instructions provided in section **Replicating a Parameter Group**.

Precautions

- Default parameter groups are unchangeable. You can only view them by clicking their names. If inappropriate settings of customized parameter groups lead to a database startup failure, you can reset the customized parameter group by referring to the settings of the default parameter group.
- After modifying a parameter, you need to view the associated instance status in the instance list. If **Pending restart** is displayed, you need to restart the instance for the modification to take effect.
- Improperly setting parameters in a parameter group may have unintended adverse effects, including degraded performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter group. Before applying parameter changes to a production DB instance, you should try out these changes on a test DB instance.

9.2 Creating a Parameter Group

Scenarios

DB parameter groups act as a container for engine configuration values that are applied to one or more DB instances. This section describes how to create a parameter group to manage your DB instance configurations.

NOTE

- DDS does not share parameter group quotas with RDS.
- Each account can create up to 100 DDS parameter groups for the cluster, replica set, and single node instances.

Cluster

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, click **Create Parameter Group**.

Step 4 Specify **DB Engine Version**, **DB Instance Type**, **Node Type**, **Parameter Group Name**, and **Description** and then click **OK**.

- **Node Type**: specifies the node type that this parameter group will apply to. For example, to create a parameter group applying to config, select **config**.
- **Parameter Group Name**: specifies the parameter group name, which is a string of 1 to 64 characters composed of only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.).
- **Description**: contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 5 On the **Parameter Group Management** page, view and manage parameter groups on the **Clusters** tab.

----End

Replica Set

Step 1 Log in to the DDS console.

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, click **Create Parameter Group**.

Step 4 Specify **DB Engine Version**, **DB Instance Type**, **Parameter Group Name**, and **Description** and then click **OK**.

- **Parameter Group Name**: specifies the parameter group name, which is a string of 1 to 64 characters composed of only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.).
- **Description**: contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 5 On the **Parameter Group Management** page, view and manage parameter groups on the **Replica Sets** tab.

----End

Single Node

Step 1 Log in to the DDS console.

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, click **Create Parameter Group**.

Step 4 Specify **DB Engine Version**, **DB Instance Type**, **Parameter Group Name**, and **Description** and then click **OK**.

- **Parameter Group Name**: specifies the parameter group name, which is a string of 1 to 64 characters composed of only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.).
- **Description**: contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 5 On the **Parameter Group Management** page, view and manage parameter groups on the **Single Nodes** tab.

----End

9.3 Editing a Parameter Group

Scenarios

This section describes how to edit parameters in the parameter groups that you have created to meet your service requirements and achieve optimal performance.

NOTE

Default parameter groups are unchangeable. You can only view them by clicking their names. If inappropriate settings of customized parameter groups lead to a database startup failure, you can reset the customized parameter group by referring to the settings of the default parameter group.

Procedure

Step 1 Log in to the DDS console.

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate and click the target parameter group.

Step 4 Modify the required parameters.

Related parameters are described as follows:

- For details on parameter descriptions, visit **MongoDB official website**.
- The default value of the **net.maxIncomingConnections** parameter varies according to DB instance specifications. Therefore, this parameter is set to **default** before being specified.
- **disableJavaScriptJIT** and **security.javascriptEnabled** are used together to set the statistical function.
  - **disableJavaScriptJIT**: The default value is **true**, indicating that the JavaScriptJIT compiler is disabled.
  - **security.javascriptEnabled**: The default value is **false**, indicating that JavaScript cannot be executed on mongod and the mapReduce and group commands cannot be used.

Possible operations are as follows:

- If you want to save the modifications, click **Save**.
- If you want to cancel the modifications, click **Cancel**.
- If you want to preview the modifications, click **Preview**.

NOTE

For details on the description of parameter group status, see **Parameter Group Status**.

After modifying a parameter, you need to view the associated instance status in the instance list. If **Pending restart** is displayed, you need to restart the instance for the modification to take effect.

----End

9.4 Comparing Two Parameter Groups

Scenarios

This section describes how to compare two parameter groups of the same node type and DB engine version.

Procedure

Step 1 Log in to the DDS console.

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate the target parameter group, and click **Compare**.

Step 4 In the displayed **Compare Parameter Group** dialog box, select a parameter group for **Group2** and click **OK**.

If the settings of the two parameter groups are different, the parameter names and values of group 1 and group 2 parameter groups are displayed. If the settings are the same, no data is displayed.

----End

9.5 Replicating a Parameter Group

Scenarios

This section describes how to replicate a parameter group you created and assign it a name different from that of original group.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate the target parameter group, and click **Replicate**.

Step 4 Enter the new parameter group name and description and click **OK**.

- **Parameter Group Name**: specifies the parameter group name, which is a string of 1 to 64 characters composed of only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.).
- **Description**: contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

Step 5 After the creation is complete, you can manage the parameter group in the parameter group list on the corresponding tab.

----End

9.6 Changing Associated Parameter Group

Scenarios

After a DB instance is created, you can change the parameter group associated with the DB instance to achieve optimal performance. The parameter group associated with the DB instance cannot be changed in any of the following cases:

- The account is frozen.
- The user is not authorized.
- A DB instance is being restarted.
- A backup file is being created.
- Cluster instance nodes are being added.

- The storage space is being expanded.
- The instance class is being changed.
- An SSL connection is being enabled or disabled.
- The port is being changed.

Cluster

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target cluster instance.

Step 3 In the **Node Information** area on the **Basic Information** page, click **mongos**, **shard**, or **config**, locate the target node, and click **Change Parameter Group** or choose **More** > **Change Parameter Group**.

Step 4 On the displayed dialog box, select the parameter group to be modified and click **OK**.

- Changes to certain parameters take effect only after you restart the DB instance. Other changes take effect immediately.
- If no parameter groups are available for **New Parameter Group**, create a parameter group. For details, see section **Creating a Parameter Group**.

----End

Replica Set

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target replica set instance, and choose **More** > **Change Parameter Group** in the **Operation** column.

Step 3 On the displayed dialog box, select the parameter group to be modified and click **OK**.

- Changes to certain parameters take effect only after you restart the DB instance. Other changes take effect immediately.
- If no parameter groups are available for **New Parameter Group**, create a parameter group. For details, see section **Creating a Parameter Group**.

----End

Single Node

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, locate the target single node instance, and choose **More** > **Change Parameter Group** in the **Operation** column.

Step 3 On the displayed dialog box, select the parameter group to be modified and click **OK**.

- Changes to certain parameters take effect only after you restart the DB instance. Other changes take effect immediately.

- If no parameter groups are available for **New Parameter Group**, create a parameter group. For details, see section **Creating a Parameter Group**.

----End

9.7 Resetting a Parameter Group

Scenarios

This section describes how to reset all parameters in a parameter group you create to the default settings as needed.

NOTICE

Resetting the parameter group will restore the default values. Exercise caution when performing this operation.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate the target parameter group, and choose **More** > **Reset**.

Step 4 In the displayed dialog box, click **Yes**.

----End

9.8 Changing the Parameter Group Description

The section describes how to modify the description of the parameter group you created so that you can distinguish and identify parameter groups.

NOTE

The description of a default parameter group cannot be modified.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate the target parameter group, and click  in the **Description** column.

Step 4 Enter new description information. The parameter group description contains a maximum of 256 characters and cannot contain the carriage return character and the following special characters: >!<"&'=

- To submit the change, click **OK**. After the modification is successful, you can view the new description in the **Description** column of the parameter group list.
- To cancel the change, click **Cancel**.

----End

9.9 Deleting a Parameter Group

Scenarios

This section describes how to delete a parameter group. The following parameter groups cannot be deleted.

- Default parameter groups
- Parameter groups associated with DB instances

NOTICE

Deleted parameter groups cannot be restored. Exercise caution when performing this operation.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Parameter Group Management**.

Step 3 On the **Parameter Group Management** page, locate the target parameter group, and choose **More** > **Delete**.

Step 4 In the displayed dialog box, click **Yes**.

----End

10 Task Center

Scenarios

This section describes how to view the progress and result of asynchronous tasks on the **Task Center** page.

Tasks Overview

- Creating a DB instance

  Creating a Community Edition cluster instance, replica set instance, or single node instance.
- Scaling up storage space

  Scaling up the storage space of the shard node of a Community Edition cluster instance, or the storage space of a replica set instance, or a single node instance.
- Changing instance class

  Changing the class of a Community Edition cluster instance, replica set instance, or a single node instance.
- Adding nodes

  Adding nodes to a Community Edition cluster instance.
- Restarting DB instances

  Restarting a cluster instance, one or more cluster instance nodes, a replica set instance, or a single-node instance.
- Restoring to a new DB instance

  Restoring data to a new Community Edition cluster instance, replica set instance, or single node instance.
- Restoring to a point in time

  Restoring a replica set instance to a point in time.

NOTE

Tasks that fail to be executed will be retained for seven days by default.

Procedure

Step 1 **Log in to the DDS console.**

Step 2 In the navigation pane on the left, click **Task Center**.

Step 3 In the navigation pane on the left, choose **Task Center**. Then, view the task progresses and results.

- You can view tasks in a specified period.
- The tasks can be located by DB instance name and ID or by task status or type from the drop-down list in the upper right corner.

----End

11 Monitoring and Alarm Reporting

11.1 DDS Metrics

Function

This section describes metrics reported by Document Database Service (DDS) to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for DDS.

Namespace

SYS.DDS

Monitoring Metrics

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo001_command_ps	COMMAND Statements per Second	Number of COMMAND statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo002_delete_ps	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute
mongo003_insert_ps	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute
mongo004_query_ps	QUERY Statements per Second	Number of QUERY statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo005_update_ps	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute
mongo006_getmore_ps	GETMORE Statements per Second	Number of GETMORE statements executed per second	≥ 0 Count/s	Monitored object: database Monitored object type: • DDS DB instance • mongos node • Primary node • Secondary node	1 minute
mongo007_chunk_num1	Chunks of Shard 1	Number of chunks in shard 1	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num2	Chunks of Shard 2	Number of chunks in shard 2	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo007_chunk_num3	Chunks of Shard 3	Number of chunks in shard 3	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num4	Chunks of Shard 4	Number of chunks in shard 4	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num5	Chunks of Shard 5	Number of chunks in shard 5	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num6	Chunks of Shard 6	Number of chunks in shard 6	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num7	Chunks of Shard 7	Number of chunks in shard 7	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num8	Chunks of Shard 8	Number of chunks in shard 8	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo007_chunk_num9	Chunks of Shard 9	Number of chunks in shard 9	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num10	Chunks of Shard 10	Number of chunks in shard 10	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num11	Chunks of Shard 11	Number of chunks in shard 11	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo007_chunk_num12	Chunks of Shard 12	Number of chunks in shard 12	0–64 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo008_connections	Active Instance Connections	Total number of connections attempting to connect to a DDS DB instance	0–200 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute
mongo009_migFail_num	Chunk Migration Failures in Last 24 hrs	Number of chunk migration failures in the last 24 hours	≥ 0 Counts	Monitored object: database Monitored object type: DDS DB instance	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo007_connections	Active Node Connections	Total number of connections attempting to connect to a DDS DB instance node	0–200 Counts	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo007_connections_usage	Percentage of Active Node Connections	Percentage of the number of connections that attempt to connect to the instance node to the total number of available connections	0~100 %	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo008_mem_resident	Resident Memory	Size of resident memory in MB	≥ 0 MB	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo009_mem_virtual	Virtual Memory	Size of virtual memory in MB	≥ 0 MB	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo010_regular_asserts_ps	Regular Asserts per Second	Number of regular asserts per second	≥ 0 Count/s	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo011_warning_asserts_ps	Warning Asserts per Second	Number of warning asserts per second	≥ 0 Count/s	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo012_msg_asserts_ps	Message Asserts per Second	Number of message asserts per second	≥ 0 Count/s	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo013_user_asserts_ps	User Asserts per Second	Number of user asserts per second	≥ 0 Count/s	Monitored object: database Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo014_queues_total	Operations Queued Waiting for a Lock	Number of operations queued waiting for a lock	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo015_queues_readers	Operations Queued Waiting for a Read Lock	Number of operations queued waiting for a read lock	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo016_queues_writers	Operations Queued Waiting for a Write Lock	Number of operations queued waiting for a write lock	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo017_page_faults	Page Faults	Number of page faults on the monitored nodes	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo018_porfling_num	Slow Queries	Number of slow queries on the monitored nodes	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo019_cursors_open	Maintained Cursors	Number of maintained cursors on the monitored nodes	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo020_cursors_timeOut	Timeout Cursors	Number of timed out cursors on the monitored nodes	≥ 0 Counts	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo021_wt_cahe_usage	Bytes in WiredTiger Cache	Size of data in the WiredTiger cache in MB	≥ 0 MB	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo022_wt_cahe_dirty	Tracked Dirty Bytes in WiredTiger Cache	Size of tracked dirty data in the WiredTiger cache in MB	≥ 0 MB	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo023_wInto_wtCache	Bytes Written Into Cache per Second	Bytes written into WiredTiger cache per second	≥ 0 bytes/s	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo024_wFrom_wtCache	Bytes Written From Cache per Second	Bytes written from the WiredTiger cache to the disk per second	≥ 0 bytes/s	Monitored object: database Monitored object type: • Primary node • Secondary node	1 minute
mongo025_repl_oplog_win	Oplog Window	Available time in hour in the monitored primary node's oplog	≥ 0 Hours	Monitored object: database Monitored object type: primary node	1 minute
mongo026_oplog_size_ph	Oplog Growth Rate	Speed in MB/hour at which oplogs are generated on the monitored primary node	≥ 0 MB/Hour	Monitored object: database Monitored object type: primary node	1 minute
mongo025_repl_headroom	Replication Headroom	Time difference in seconds between the primary's oplog window and the replication lag of the secondary	≥ 0 Seconds	Monitored object: database Monitored object type: secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo026_repl_lag	Replication Lag	A delay in seconds between an operation on the primary and the application of that operation from the oplog to the secondary	≥ 0 Seconds	Monitored object: database Monitored object type: secondary node	1 minute
mongo027_repl_command_ps	Replicated COMMAND Statements per Second	Number of replicated COMMAND statements executed on the secondary node per second	≥ 0 Count/s	Monitored object: database Monitored object type: secondary node	1 minute
mongo028_repl_update_ps	Replicated UPDATE Statements per Second	Number of replicated UPDATE statements executed on the secondary node per second	≥ 0 Count/s	Monitored object: database Monitored object type: secondary node	1 minute
mongo029_repl_delete_ps	Replicated DELETE Statements per Second	Number of replicated DELETE statements executed on the secondary node per second	≥ 0 Count/s	Monitored object: database Monitored object type: secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo030_repl_insert_ps	Replicated INSERT Statements per Second	Number of replicated INSERT statements executed on the secondary node per second	≥ 0 Count/s	Monitored object: database Monitored object type: secondary node	1 minute
mongo031_cpu_usage	CPU Usage	CPU usage of the monitored object	0–1	Monitored object: ECS Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo032_mem_usage	Memory Usage	Memory usage of the monitored object	0–1	Monitored object: ECS Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo033_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored object type: • mongos node • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo034_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored object type: • mongos node • Primary node • Secondary node	1 minute
mongo035_disk_usage	Disk Utilization	Disk usage of the monitored object	0–1	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo036_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 Count/s	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo037_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute

Metrics	Metrics Name	Description	Value Range	Remarks	Monitoring Interval (Raw Data)
mongo038_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo039_avg_disk_sec_per_read	Average Time per Disk Read	Average time required for each disk read in a specified period	≥ 0 Seconds	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo040_avg_disk_sec_per_write	Average Time per Disk Write	Average time required for each disk write in a specified period	≥ 0 Seconds	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo042_disk_total_size	Total Storage Space	Total storage space of the monitored object	0–1000 GB	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute
mongo043_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–1000 GB	Monitored object: ECS Monitored object type: • Primary node • Secondary node	1 minute

Dimensions

Key	Value
mongodb_cluster_id	DDS DB instance ID Supports cluster instances of Community Edition, replica set instances, and single node instances.
mongos_instance_id	mongos node ID
mongod_primary_instance_id	Primary node ID Includes the primary config and shard nodes in a cluster instance and the primary nodes in a replica set instance.
mongod_secondary_instance_id	Secondary node ID Includes the secondary config and shard nodes of cluster instances and the secondary nodes of replica set instances.

11.2 Setting Alarm Rules

Scenarios

- You can enable the alarm reporting function in one click. After alarms are triggered, Simple Message Notification (SMN) can send notifications to specified cloud account.
- You can set DDS alarm rules to customize the monitored objects and notification policies. Then, you can learn DDS running status in a timely manner.

  The DDS alarm rules include alarm rule name, instance, metric, threshold, monitoring interval and whether to send notification.

NOTE

For more information about DDS alarm rules, see *Cloud Eye User Guide*.

Enabling Alarm Reporting

Step 1 **Log in to the DDS console.**

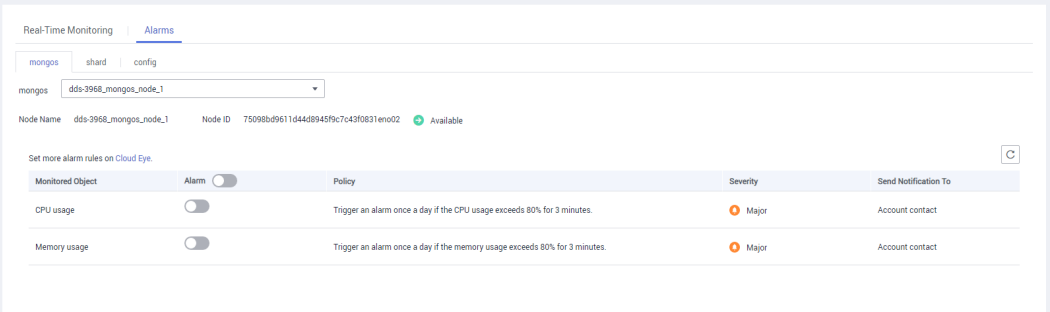
Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Advanced O&M**.

Step 4 On the **Advanced O&M** page, click the **Alarms** tab and click  next to the target alarm policy to enable alarm reporting function.

The following figure uses a cluster instance as an example.

Figure 11-1 Cluster instance alarm settings



NOTE

The basic alarm function is free of charge. SMN sends you the alarm messages and charges you for that. For pricing details, see **Pricing Details**.

Step 5 If you want to disable the alarm reporting function, click  .

----End

Setting Alarm Rules

Step 1 Log in to the management console.

Step 2 Under **Management & Deployment**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.

Step 4 On the **Alarm Rules** page, click **Create Alarm Rule** to create an alarm rule, or modify an existing alarm rule.

The following operations use the modification of an existing alarm rule as an example.

Locate the alarm rule to be modified and choose **More** > **Modify**.

Click **OK**.

Step 5 After the alarm rule is set, the system automatically notifies you when an alarm is triggered.

----End

11.3 Viewing DDS Metrics

Scenarios

Cloud Eye monitors DDS running statuses. You can obtain the monitoring metrics of DDS on the management console.

Monitored data requires a period of time for transmission and display. The status of DDS displayed on the Cloud Eye page is the status obtained 5 to 10 minutes before. You can view the monitored data of a newly created DB instance 5 to 10 minutes later.

Prerequisites

- The DDS DB instance is running properly.

  Cloud Eye does not display the metrics of a faulty or deleted DB instance or node. You can view the monitoring information only after the instance is restarted or recovered.

- The DB instance has been properly running for at least 10 minutes.

  For a newly created DB instance, you need to wait for a while before viewing the monitoring metrics.

Procedure

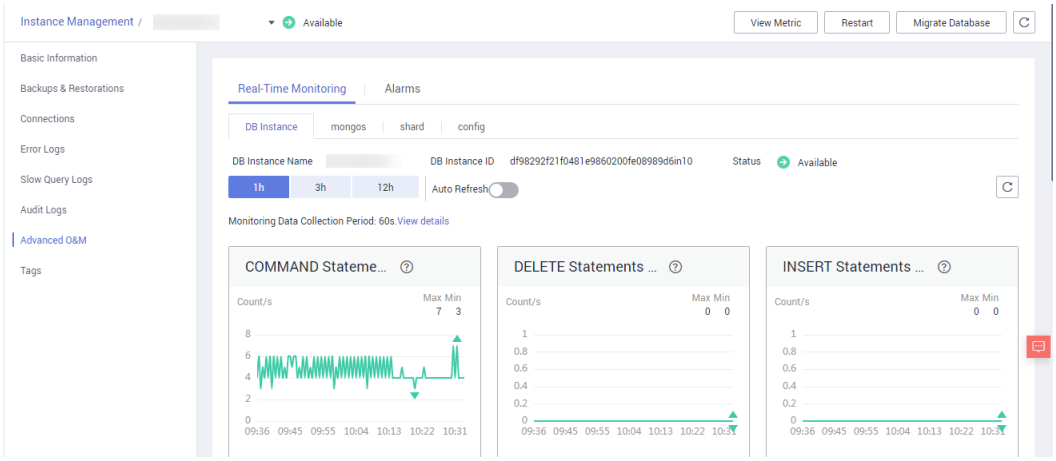
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, choose **Advanced O&M**.

Step 4 View metrics of cluster instance, nodes in cluster instances, nodes in replica set instances, and single-node instances. The following figure uses a cluster instance as an example.

Figure 11-2 Cluster instance monitoring information



Step 5 In the DDS monitoring area, you can select a duration to view the monitoring data.

- You can view the monitoring data of the last 1 hour, 3 hours, and 12 hours.
- After the automatic refresh function is enabled, monitoring data is automatically refreshed at an interval of 60s.
- For more metric information, click **View details** to switch to the Cloud Eye console.

----End

12 Auditing

12.1 Key Operations Recorded by CTS

With Cloud Trace Service (CTS), you can record operations associated with DDS for later query, audit, and backtrack operations.

Table 12-1 Key operations on DDS

Operation	Resource	Trace Name
Restoring data to a new DB instance	instance	ddsRestoreToNewInstance
Creating a DB instance	instance	ddsCreateInstance
Deleting a DB instance	instance	ddsDeleteInstance
Restarting a DB instance	instance	ddsRestartInstance
Scaling up a DB instance	instance	ddsGrowInstance
Scaling up storage space	instance	ddsExtendInstanceVolume
Resetting the database password	instance	ddsResetPassword
Renaming a DB instance	instance	ddsRenameInstance
Switching SSL	instance	ddsSwitchSsl
Modifying a DB instance port	instance	ddsModifyInstancePort
Creating a backup	backup	ddsCreateBackup
Deleting a backup	backup	ddsDeleteBackup
Setting a backup policy	backup	ddsSetBackupPolicy

Operation	Resource	Trace Name
Applying a parameter group	parameterGroup	ddsApplyConfigurations
Replicating a parameter group	parameterGroup	ddsCopyConfigurations
Resetting a parameter group	parameterGroup	ddsResetConfigurations
Creating a parameter group	parameterGroup	ddsCreateConfigurations
Deleting a parameter group	parameterGroup	ddsDeleteConfigurations
Updating a parameter group	parameterGroup	ddsUpdateConfigurations
Binding an EIP	instance	ddsBindEIP
Unbinding an EIP	instance	ddsUnbindEIP
Adding a tag	tag	ddsAddTag
Deleting a tag	tag	ddsDeleteTag
Editing a tag	tag	ddsModifyTag
Deleting an instance tag	tag	ddsDeleteInstanceTag
Adding an instance tag	tag	ddsAddInstanceTag
Rolling back upon scaling-up failure	instance	ddsDeleteExtendedDdsNode
Changing DB instance classes	instance	ddsResizeInstance
Unfreezing a DB instance	instance	ddsUnfreezeInstance
Freezing a DB instance	instance	ddsFreezeInstance

12.2 Querying Traces

Scenarios

After CTS is enabled, the tracker starts recording operations on cloud resources. Operation records for the last 7 days are stored on the CTS console.

This section describes how to query operation records for the last 7 days on the CTS console.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Management & Deployment**, click **Cloud Trace Service**.

Step 4 Choose **Trace List** in the navigation pane on the left.

Step 5 Specify the filters used for querying traces. The following four filters are available:

- **Trace Source**, **Resource Type**, **Search By**, and **Operator**

  Select the filter from the drop-down list.

  When you select **Trace name** for **Search By**, you also need to select a specific trace name.

  When you select **Resource ID** for **Search By**, you also need to select or enter a specific resource ID.

  When you select **Resource name** for **Search By**, you also need to select or enter a specific resource name.
- **Operator**: Select a specific operator (a user rather than tenant).
- **Trace Status**: Available options include **All trace statuses**, **normal**, **warning**, and **incident**. You can only select one of them.
- Start time and end time: You can specify the time period for query traces.

Step 6 Click  on the left of the record to be queried to extend its details.

Step 7 Locate a trace and click **View Trace** in the **Operation** column.

----End

13 Log Management

13.1 Error Log

Scenarios

DDS log management allows you to view database-level logs, including warning- and error-level logs generated during database running, which help you analyze system problems.

Constraints

Community Edition DB instances allow you to view and export log details, and download log files on the management console.

Viewing and Exporting Log Details

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Error Logs**.

Step 4 On the displayed page, click **Error Logs**. Then, view the log details on the **Log Details** tab.

- Log records of different node types of a cluster instance in batches
  - If you select **All nodes**, the logs of all nodes in the cluster instance are displayed.
  - If you select **All mongos**, the logs of all mongos in the cluster instance are displayed.
  - If you select **All shards**, the logs of all shards in the cluster instance are displayed.
  - If you select **All configs**, the logs of all configs in the cluster instance are displayed.
- Error logs of all nodes of a replica set instance

- Error logs of a node in different time periods
- Error logs of the following level
  - All log levels
  - WARNING
  - ERROR

Step 5 On the **Log Details** tab, click in the upper right corner of the log list to export log details.

- View the .csv file exported to your local PC.
- A maximum of 10,000 log details can be exported at a time.

----End

Downloading Logs

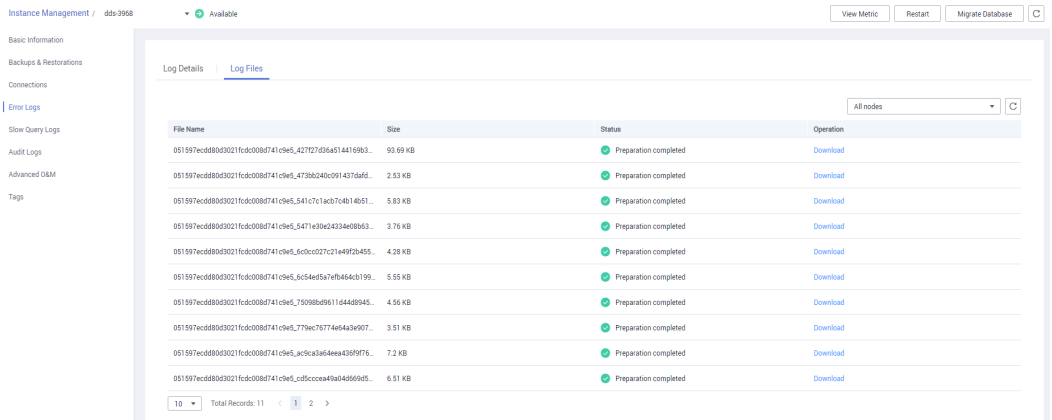
Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target Community Edition instance.

Step 3 In the navigation pane on the left, click **Error Logs**.

Step 4 On the **Error Logs** page, click the **Log Files** tab. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 13-1 Error Logs



- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - During the downloading preparation, the log status is **Preparing**.
  - After the downloading preparation is complete, the log status is **Preparation completed**.
  - If the downloading preparation fails, the log status is **Abnormal**.
- You can download only one log file from a node. The maximum size of a log file to be downloaded is 40 MB.

- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

----End

13.2 Slow Query Log

Scenarios

Slow query logs record statements whose execution period exceeds the value of **operationProfiling.slowOpThresholdMs** (100 ms by default). With slow query logs, you can identify and optimize slowly executed statements.

Constraints

- Community Edition DB instances allow you to view and export log details, enable the Show Original Log function, and download log files on the management console.
- The Show Original Log function cannot be enabled when you delete DB instances, add nodes, change DB instance class, rebuild secondary node, or the DB instance is frozen.
- When the Show Original Log function is being enabled, you cannot delete DB instances, add nodes, change DB instance class, or rebuild secondary node.

Related Parameters

Table 13-1 Parameter description

Parameter	Description
operationProfiling.mode	Specifies database profiling (analysis) level. The parameter value is **slowOp** by default. • **off**: Disables the analyzer and data collection. • **slowOp**: Collects data related to queries that exceed a given threshold of execution time. • **all**: Collects all operations data.
operationProfil-ing.slowOpThresholdMs	Queries that exceed the threshold in the unit of ms are deemed slow. The parameter value is **100 ms** by default. Unless otherwise specified, setting the value to 100 ms is recommended.

Showing Original Logs

NOTE

- After Show Original Log is enabled, it cannot be disabled.
- After Show Original Log, original logs are displayed. The original slow query logs are displayed for your query and retained for 30 days.
- If the DB instance to which the slow query log belongs is deleted, related logs are also deleted after Show Original Log is enabled.

Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Slow Query Logs**.

Step 4 On the displayed page, click **Slow Query Logs**. Then, click  on the **Log Details** tab.

Step 5 In the displayed dialog box, click **Yes** to enable the function of slowing original logs.

----End

Viewing and Exporting Log Details

Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Slow Query Logs**.

Step 4 On the **Slow Query Logs** page, set search criteria on the **Log Details** tab and click **Search** to view log information.

- Log records of all shards of a cluster instance
- Log records of all nodes of a replica set instance
- Slow query logs of a node in different time periods
- Slow query statements of the following level
  - All statement type
  - INSERT
  - QUERY
  - UPDATE
  - REMOVE
  - GETMORE
  - COMMAND
  - KILLCURSORS

Step 5 On the **Log Details** tab, click  in the upper right corner of the log list to export log details.

- View the .csv file exported to your local PC.
- A maximum of 10,000 log details can be exported at a time.

----End

Downloading Logs

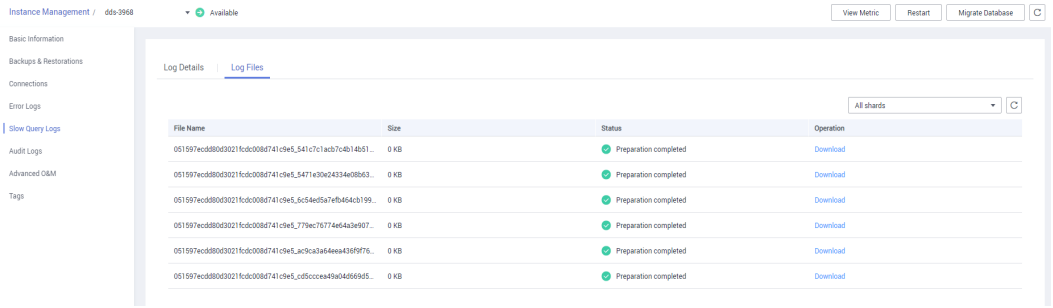
Step 1 Log in to the DDS console.

Step 2 On the **Instance Management** page, click the target Community Edition instance.

Step 3 In the navigation pane on the left, click **Slow Query Logs**.

Step 4 On the **Slow Query Logs** page, click the **Log Files** tab. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 13-2 Slow Query Logs



- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - During the downloading preparation, the log status is **Preparing**.
  - After the downloading preparation is complete, the log status is **Preparation completed**.
  - If the downloading preparation fails, the log status is **Abnormal**.
- You can download only one log file from a node. The maximum size of a log file to be downloaded is 40 MB.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

----End

13.3 Audit Log

Scenarios

DDS supports log auditing. The feature is disabled by default because enabling it may have a slight impact on your service performance.

An audit log records operations performed on your databases and collections. The generated log files are stored in OBS. Auditing logs can enhance your database security and help you analyze the cause of failed operations.

Precautions

- DDS checks generated audit logs. If the retention period of logs exceeds the period you set, DDS will delete the logs.

- After the audit policy is modified, DDS audits logs according to the new policy and the retention period of the original audit logs is subject to the modified retention period.
- You will be charged for enabling SQL audit log. For details, see **Service Pricing**.

Querying Audit Logs

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Audit Logs**.

Step 4 On the **Audit Logs** page, view the log details.

You can view log records of a node in different time periods.

----End

Setting the Audit Policy

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Audit Logs**.

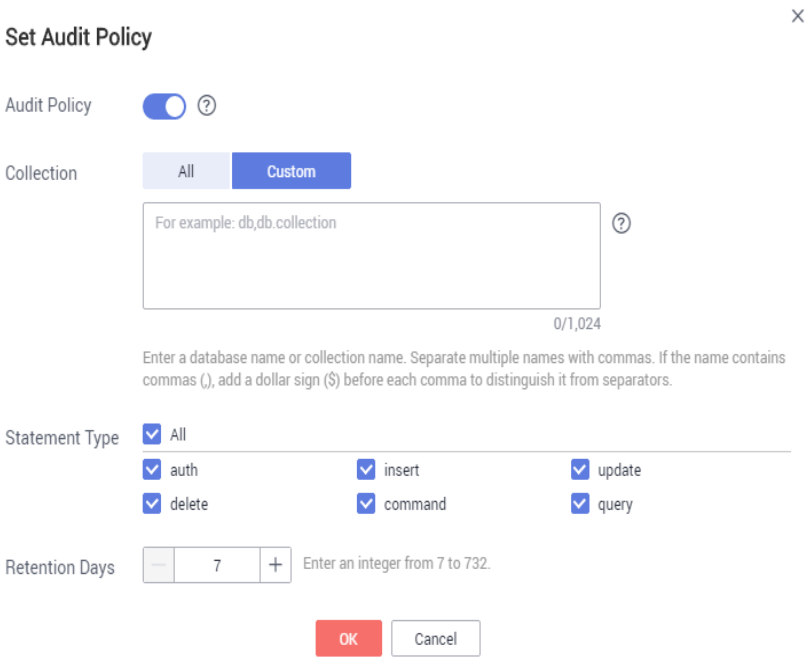
Step 4 On the **Audit Logs** page, click **Set Audit Policy**.

Step 5 On the displayed page, enable, modify, or disable audit policies.

- Enable or modify the audit policy.

  To enable the audit policy, click  . Once enabled, the audit policy can be modified.

Figure 13-3 Modifying the audit policy



  - **All**: audits all collections of the DB instance.
  - **Custom**: audits specified databases or collections of the DB instance.

    The database or collection name cannot contain spaces or the following special characters: /\' : "[]{}() The dollar sign ($) can be used only as an escape character.

    The database name can contain a maximum of 64 characters.

    If you enter a combined database and collection name, the total name length is 120 characters with the database name length of no more than 64 characters and the collection name cannot be blank, contain **null**, or use **system.** in prefix.
  - **Statement Type**: audits specified statements, including auth, insert, update, delete, command, and query statements.
  - **Retention Days**: indicates the number of days to retain audit logs and can range from 7 to 732 days.

  Click **OK** to save the modification. After the modification, logs are generated according to the new policy and the retention period of the original logs is subject to the modified retention period.

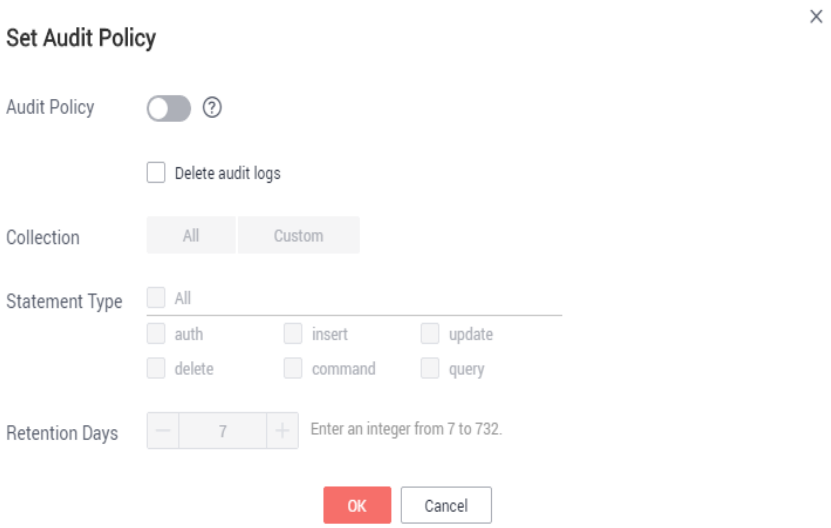
- Disable the audit policy.

  NOTE

  After the audit policy is disabled, no audit log is generated.

  To disable the audit policy, click  .

Figure 13-4 Disabling the audit policy



You can determine whether to delete all automated backup files:

- If you do not select **Delete automated backups**, all backup files within the retention period will be retained. You can manually delete them later.
- If you select **Delete automated backups**, all backup files within the retention period will be deleted.

Click **OK**.

----End

Downloading Logs

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Audit Logs**.

Step 4 On the **Audit Logs** page, click **Download** to download audit logs.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

----End

14 Tag

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally and other cloud services manage their own tags.

Adding tags to DDS DB instances helps you better identify and manage them. A DB instance can be tagged during or after it is created.

- You are advised to set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key. For details about the naming rules of tag keys and tag values, see **Table 14-1**.
- Up to 20 tags can be added for a DB instance.

Table 14-1 Naming rules

Parameter	Requirement	Example
Tag key	• The key cannot be left blank. • Each tag key must be unique for each DB instance. • A tag key consists of up to 36 characters. • The key can only consist of digits, letters, underscores (_), and hyphens (-).	Organization
Tag value	• This tag value can be left blank. • The value consists of up to 43 characters. • The value can only consist of digits, letters, underscores (_), dots (.), and hyphens (-).	dds_01

Adding a Tag

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Tags**.

Step 4 On the **Tags** page, click **Add Tag**. In the displayed dialog box, enter a tag key and value, and click **OK**.

Step 5 View and manage tags on the **Tags** page.

----End

Editing a Tag

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Tags**.

Step 4 On the **Tags** page, locate the tag to be edited and click **Edit** in the **Operation** column. In the displayed dialog box, change the tag value and click **OK**.

Only the tag value can be edited when editing a tag.

Step 5 View and manage tags on the **Tags** page.

----End

Deleting a Tag

Step 1 **Log in to the DDS console.**

Step 2 On the **Instance Management** page, click the target DB instance.

Step 3 In the navigation pane on the left, click **Tags**.

Step 4 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Step 5 After a tag has been deleted, it will not be displayed on the **Tags** page.

----End

15 Quotas

Scenarios

Quotas are enforced for service resources on the platform to prevent unforeseen spikes in resource usage. For example, the maximum number of DDS DB instances that can be created varies depending on the DB instance type. You can apply for increasing quotas if necessary.

This section describes how to view the usage of each type of DDS resource and the total quotas in a specified region.

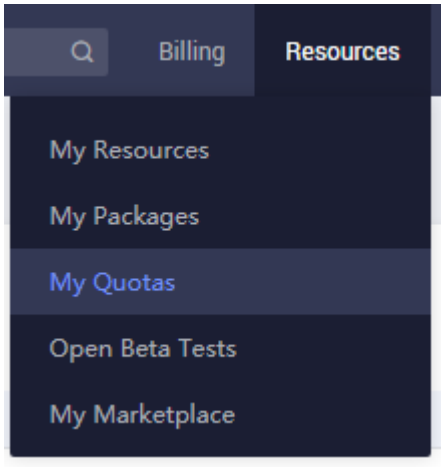
Viewing Quotas

Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select a region and a project.

Step 3 In the upper right corner of the DDS console, choose **Resources** > **My Quota**.

Figure 15-1 My Quota



Step 4 View the used and total quota of each type of DDS resource.

Step 5 If a quota cannot meet service requirements, click **Increase Quota** to adjust it.

----End

Increasing Quotas

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 In the upper right corner of the DDS console, choose **Resources** > **My Quota**.

Step 4 Click **Increase Quota**.

Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for adjustment.

Step 6 After all necessary parameters are configured, select **I have read and agree to the Tenant Authorization Letter and Privacy Statement** and click **Submit**.

----End

16 Troubleshooting

16.1 Overview

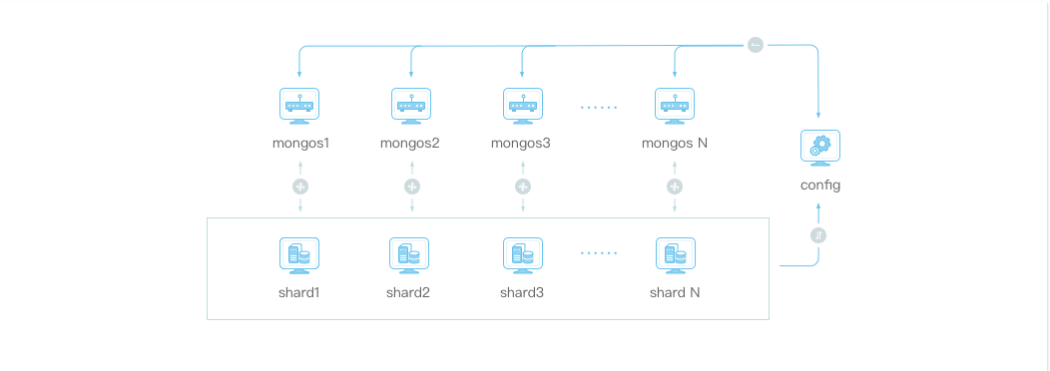
You can use the Data Admin Service (DAS), Mongo Shell, program code, or other tools to connect to a DDS DB instance. This chapter describes typical problems you may encounter when connecting to a DDS DB instance and how to solve the problems.

16.2 DDS Instance Node Fault Handling Mechanism

Cluster Instance

The shard and config nodes of the cluster instance use the three-node replica set architecture. When a node is faulty, the system uses another normal node to replace the faulty node to continue providing services. In addition, the system checks and rectifies the faulty node. This process is transparent to users and may cause intermittent disconnection within 30 seconds. You are advised to enable automatic reconnection on your application.

Figure 16-1 Cluster instance diagram



The mongos node of a cluster instance uses the single-node architecture. When a mongos node is faulty, services on the node become unavailable. You are advised

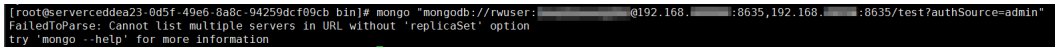
to use Connection String URI to connect to all mongos nodes instead of a single mongos node. If a mongos node becomes faulty, the client can automatically perform failover and distribute requests to normal mongos nodes. Command example:

mongo "mongodb://
rwuser:xxxxxxxx@192.168.95.167:8635,192.168.92.43:8635/test?
authSource=admin"

NOTICE

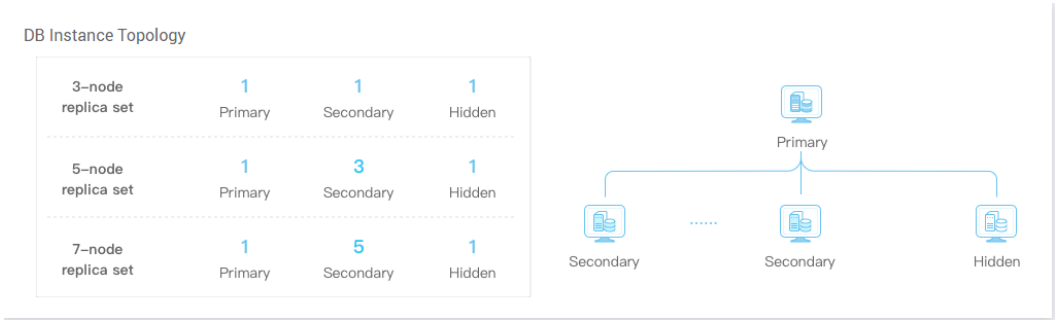
If you use Connection String URI to connect to a cluster instance that is compatible with MongoDB 3.4, you are advised to use a MongoDB client later than 4.0. Otherwise, an error is reported.

Figure 16-2 Error message



Replica Set Instance

Figure 16-3 Replica set instance diagram



A replica set instance consists of two nodes. When a node is faulty, the system uses the other node to continue providing services. In addition, the system checks and rectifies the faulty node. This process is transparent to users and may cause intermittent disconnection within 30 seconds. You are advised to enable automatic reconnection on your application.

You are advised to use Connection String URI to connect to the nodes. Do not directly connect to the primary node of a replica set instance. If Connection String URI is used and a node becomes faulty, the read and write operations will not be affected by the failover. Command example:

mongo "mongodb://
rwuser:xxxxxxxx@192.168.168.116:8635,192.168.200.147:8635/test?
authSource=admin&replicaSet=replica"

For best practices of connecting replica set instances, see **Connecting to a Replica Set Instance for Read and Write Separation and High Availability**.

Single-Node Instance

Figure 16-4 Single-node instance diagram

As the name implies, single-node instance provides only one node for users to access. When a node is faulty, the system checks and rebuilds the faulty node. In the meanwhile, services on the node become unavailable.

Single-node instances apply to scenarios such as testing, training, and non-core services. You are advised to use cluster instances or replica set instances in the production environment to ensure high service availability.

16.3 Mongo Shell Fails to Connect to the DB Instance, Leaving Message "isMaster"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance:

./mongo --host 192.168.168.182 --port 8635 -u rwuser -p *xxxxxxxxxx* **--authenticationDatabase admin**

Figure 16-5 Connection failed

```
[root@ecs-liuwai bin]# ./mongo --host 192.168.168.182 --port 8635 -u rwuser -p --authenticationDatabase admin
MongoDB shell version v3.4.0
Enter password:
connecting to: mongodb://192.168.168.182:8635/
2019-01-10T09:28:27.002+0800 E QUERY    [main] Error: network error while attempting to run command 'isMaster' on host '192.168.
168.182:8635'  :
connect@src/mongo/shell/mongo.js:234:13
@(connect):1:6
exception: connect failed
[root@ecs-liuwai bin]#
```

Possible Cause

This preceding command is used to connect to a DB instance in non-SSL mode. If the SSL connection is enabled, an error is reported when you run this command.

Solution

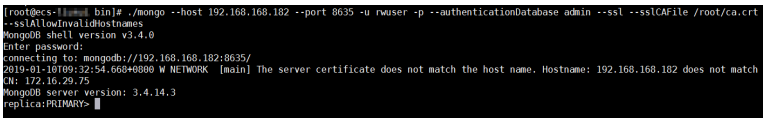
- Run the following command disable SSL connection and use a common connection to connect to the DB instance:
- Download the SSL certificate, upload the certificate to the ECS directory (for example, **/root/ca.crt**), and run the command in SSL mode to connect to the instance.

  NOTE

  You can click the DB instance name on the DDS console, click next to **SSL** on the **Connections** page, and download the SSL certificate.

./mongo --host 192.168.168.182 --port 8635 -u rwuser -p *xxxxxxxxxx* --authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --sslAllowInvalidHostnames

Figure 16-6 Connection succeeded



16.4 Mongo Shell Fails to Connect to the DB Instance, Leaving Message "No route to host" and "connection attempt failed"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance:

mongo --host 192.168.1.6 --port 8635 -u rwuser -p *xxxxxxxxx* --authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --sslAllowInvalidHostnames

Error message:

```
MongoDB shell version v3.4.17
connecting to: mongodb://192.168.1.6:8635/
2019-09-19T09:38:36.954+0800 W NETWORK  [thread1] Failed to connect to 192.168.1.6:8635, in(checking socket for error after poll), reason: No route to host
2019-09-19T09:38:36.954+0800 E QUERY    [thread1] Error: couldn't connect to server 192.168.1.6:8635, connection attempt failed :
connect@src/mongo/shell/mongo.js:240:13
@(connect):1:6
exception: connect failed
```

Possible Cause

- The DB instance port is incorrect.
- The DDS instance and ECS are not in the same region.
- The DDS DB instance and ECS are not in the same subnet.

For details about how to connect to a DB instance through a private network, see **Connecting to a DB Instance Through an ECS**.

Solution

You can run the curl command to check whether the network connection to the DDS instance is normal. The following is a command example:

curl 192.168.1.6:8635

If the message "It looks like you are trying to access MongoDB over HTTP on the native driver port." is displayed, the IP address can be connected and port 8635 can be used for communication.

Figure 16-7 Command output

```
[root@serverceddea23-0d5f-49e6-8a8c-94259dcf09cb ycsb]# curl 192.168.[redacted]:8635
It looks like you are trying to access MongoDB over HTTP on the native driver port.
```

16.5 Mongo Shell Fails to Connect to the DB Instance, Leaving Message "Authentication failed"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance:

mongo --host 192.168.168.116 --port 8635 -u rwuser -p *xxxxxxxxx* **--authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --sslAllowInvalidHostnames**

Error message:

```
MongoDB shell version v3.4.17
connecting to: mongodb://192.168.168.116:8635/
2019-09-19T09:39:24.306+0800 W NETWORK  [thread1] The server certificate does not match the host name. Hostname: 192.168.168.116 does not match CN: 172.16.2.65
MongoDB server version: 4.0.3
WARNING: shell and server versions do not match
2019-09-19T09:39:24.329+0800 E QUERY    [thread1] Error: Authentication failed. :
DB.prototype._authOrThrow@src/mongo/shell/db.js:1461:20
@(auth):6:1
@(auth):1:2
exception: login failed
```

Possible Cause

The administrator password in the command for connecting to the DDS instance is incorrect.

Solution

- Enter the correct administrator password.
- If you forget the password, reset the password by referring to **Resetting the Administrator Password**.

16.6 Mongo Shell Fails to Connect to the DB Instance, Leaving Message "couldn't connect to server"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance:

mongo --host 192.168.64.201 --port 8635 -u rwuser -p *xxxxxxxxx* **--authenticationDatabase admin --ssl --sslCAFile /root/ca.crt --sslAllowInvalidHostnames**

Error message:


```
MongoDB shell version v3.4.17
connecting to: mongodb://192.168.64.201:8635/
2019-09-19T09:45:48.168+0800 W NETWORK  [thread1] Failed to connect to 192.168.64.201:8635 after
5000ms milliseconds, giving up.
2019-09-19T09:45:48.168+0800 E QUERY    [thread1] Error: couldn't connect to server 192.168.64.201:8635,
connection attempt failed :
connect@src/mongo/shell/mongo.js:240:13
@(connect):1:6
exception: connect failed
```

Possible Cause

The security group policy is not correctly configured. That means the security group rules do not allow the traffic into the DDS DB instance.

Solution

Set the security group policy. For details, see **Setting a Security Group**.

16.7 Mongo Shell Fails to Connect to the DB Instance, Leaving Message "Cannot list multiple servers in URL without 'replicaSet' option"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance that is compatible with MongoDB 3.4:

mongo "mongodb://
rwuser:*xxxxxxxxx*@192.168.95.167:8635,192.168.92.43:8635/test?
authSource=admin"

Error message:

```
FailedToParse: Cannot list multiple servers in URL without 'replicaSet' option
try 'mongo --help' for more information.
```

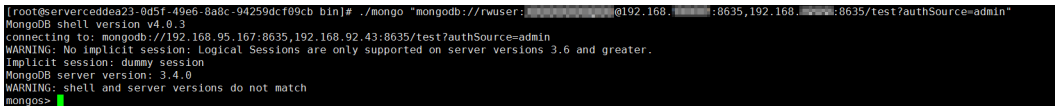
Possible Cause

The MongoDB client version is too early.

Solution

Use a MongoDB client of version later than 4.0 to connect to the instance.

Figure 16-8 Connection succeeded



16.8 Mongo Shell Fails to Connect to the Replica Set Instance, Leaving Message "Cannot list multiple servers in URL without 'replicaSet' option"

Symptom

An error is reported when you run the following command to connect to a DDS replica set instance:

mongo mongodb://
rwuser:*xxxxxxxxxx***@192.168.168.116:8635,192.168.200.147:8635/test?**
authSource=admin&replicaSet=replica

Error message:

```
[1] 8302
[root@serverceddea23-0d5f-49e6-8a8c-94259dcf09cb ycsb]# FailedToParse: Cannot list multiple servers in URL without 'replicaSet' option
try 'mongo --help' for more information
```

Possible Cause

When you use Connection String URI to connect replica set instances, the URI command line is not enclosed in double quotation marks.

Solution

Add double quotation marks to the command line and then connect the replica set instance.

mongo "mongodb://
rwuser:*xxxxxxxxx***@192.168.168.116:8635,192.168.200.147:8635/test?**
authSource=admin&replicaSet=replica"

16.9 Java Driver Fails to Connect to the DB Instance, Leaving Message "Timeout while receiving message"

Symptom

An error is reported when you run the following command to connect to a DDS DB instance through Java driver:

Error message:

```
org.springframework.data.mongodb.UncategorizedMongoDbException: Timeout while receiving message; nested exception is com.mongodb.MongoSocketReadTimeoutException: Timeout while receiving message
```

Possible Cause

- Abnormal slow queries occupy DB instance resources, causing the CPU usage to increase sharply or even reach the peak value.

- The connection pool of the application is incorrectly configured. For example, the timeout interval is incorrectly set.

Solution

- Check whether **slow queries** exist. You are advised to add indexes for optimization.
- Check whether the connection pool of the application is correctly configured. For details, see **How Do I Query and Limit the Number of Connections?**

A Change History

Released On	Description
2020-10-30	This issue is the thirty-fourth official release, which incorporates the following change: Supported up to 20 tags per instance.
2020-09-30	This issue is the thirty-third official release, which incorporates the following change: • Taken Enhanced Edition offline. • Supported Kunpeng-based instances of Community Edition 4.0.
2020-08-30	This issue is the thirty-second official release, which incorporates the following changes: • Supported up to 32 mongos nodes and 32 shard nodes in each Community Edition cluster instance. • Supported up to 3000 GB of the replica set storage space. • Supported restoring a DB instance of Community Edition to the original DB instance.
2020-07-30	This issue is the thirty-first official release, which incorporates the following changes: Supported cross-CIDR access to replica set instances.
2020-05-30	This issue is the thirtieth official release, which incorporates the following change: Supported the recycle bin.
2020-04-30	This issue is the twenty-ninth official release, which incorporates the following change: Supported the purchase of multi-AZ Community Edition DB instances.

Released On	Description
2020-04-15	This issue is the twenty-eighth official release, which incorporates the following change: Supported cross-subnet access for replica set instances in the same VPC.
2020-03-31	This issue is the twenty-seventh official release, which incorporates the following change: Added the following monitoring metrics: • mongo044_swap_usage • mongo050_top_total_time • mongo051_top_read_time • mongo052_top_write_time • mongo053_wt_flushes_status • mongo054_wt_cache_used_percent • mongo055_wt_cache_dirty_percent
2020-02-14	This issue is the twenty-sixth official release, which incorporates the following changes: • Supported adding five or seven nodes to a replica set instance. • Updated the procedures for restarting and deleting DB instances, adding nodes, scaling up storage space, changing instance class, backing up and restoring data, modifying parameter groups, and monitoring metrics.
2019-12-09	This issue is the twenty-fifth official release, which incorporates the following changes: • Added section **What Is a Parameter Group?** • Supported the deletion of frozen yearly/monthly DB instances. • Supported selecting AZs for deploying new mongos nodes. • Allowed users to view tasks, including changing specifications, adding nodes, restarting instances, or restoring instances to a point in time.
2019-11-11	This issue is the twenty-fourth official release, which incorporates the following changes: • Supported up to 2000 GB of storage of the Community Edition cluster instance. • Modified monitoring metrics of DDS. • Modified section **Which Commands are Supported or Restricted by DDS?** • Added section **MapReduce Commands**.

Released On	Description
2019-10-18	This issue is the twenty-third official release, which incorporates the following changes: • Adjusted the outline. • Added section **Creating a Database Account through DAS**. • Added the precautions for setting an automated backup policy. • Added precautions for data migration. • Modified the naming rules of tags. • Supported exporting error logs and slow query log details. • Added troubleshooting related to DB instance connections.
2019-09-11	This issue is the twenty-second official release, which incorporates the following changes: • Supported a maximum of 16 mongos nodes and 16 shards for a Community Edition cluster instance. • Supported audit logs for Community Edition DB instances. • Supported table-level, point-in-time recovery for Community Edition replica set instances.
2019-08-13	This issue is the twenty-first official release, which incorporates the following changes: • Supported showing original slow query logs for Community Edition DB instances. • Supported the statistical function by setting parameters. • Supported the point-in-time recovery for Enhanced Edition cluster instances.
2019-07-07	This issue is the twentieth official release, which incorporates the following change: Added section **Audit Log**.
2019-06-13	This issue is the nineteenth official release, which incorporates the following change: Added **Permissions Management**.
2019-04-19	This issue is the eighteenth official release, which incorporates the following changes: • Updated the procedures for restarting and deleting DB instances, backing up and restoring data, managing parameter groups, storage space, security groups, tags, and task centers, and monitoring metrics. • Supported switching the primary and secondary nodes of a replica set instance. • Added section **Migrating Data Using mongodump and mongorestore**.

Released On	Description
2019-03-25	This issue is the seventeenth official release, which incorporates the following changes: Supported retaining backups for a maximum of 732 days.
2019-02-15	This issue is the sixteenth official release, which incorporates the following changes: • Supported changing the billing mode of DB instances of Enhanced Edition from pay-per-use to yearly/monthly. • Added section **Restoring Replica Set Instance to a Local Self-Built Database**. • Deleted the specification confirmation page and related description from the pay-per-use DB instance scaling.
2019-01-07	This issue is the fifteenth official release, which incorporates the following changes: • Added a new method to change the DB instance name. • Supported restoring a DB instance of Enhanced Edition from an automated backup to an existing DB instance. • Added section **Troubleshooting**.
2018-11-23	This issue is the fourteenth official release, which incorporates the following change: Supported downloading error logs and slow query logs for DB instances of DDS Community Edition.
2018-11-02	This issue is the thirteenth official release, which incorporates the following changes: • Supported changing the pay-per-use replica set instance to yearly/monthly. • Supported changing pay-per-use DB instances to yearly/monthly in batches. • Supported renewing yearly/monthly DB instances in batches.
2018-09-26	This issue is the twelfth official release, which incorporates the following changes: • Supported changing the CPU or memory of a DB instance without interrupting services. • Interconnected with Tag Management Service (TMS).

Released On	Description
2018-09-06	This issue is the eleventh official release, which incorporates the following changes: ● Supported changing the billing mode of cluster and single node instances from pay-per-use to yearly/monthly. ● Supported case-sensitive manual backup names. ● Supported modifying the description of parameter groups created by users. ● Supported scaling down of Community Edition instance CPU and memory.
2018-08-03	This issue is the tenth official release, which incorporates the following changes: Supported downloading backup files.
2018-07-02	This issue is the ninth official release, which incorporates the following changes: ● Supported DDS Enhanced Edition. ● Supported changing a security group. ● Supported the task center. ● Optimized the automated backup policy. ● Allowed users to delete automated backups. ● Changed the maximum storage capacity of replica sets to 2000 GB.
2018-06-15	This issue is the eighth official release, which incorporates the following change: Supported the enabling or disabling the automated backup policy.
2018-06-01	This issue is the seventh official release, which incorporates the following changes: ● Supported DB instances that are compatible with MongoDB 3.4 Community Edition. ● Supported the second verification when a DB instance is restarted or deleted.
2018-05-04	This issue is the sixth official release, which incorporates the following changes: ● Supported the creation of a replica set in the yearly/monthly mode. ● Supported the open beta test for a single node. ● Put replica set into commercial use.

Released On	Description
2018-04-13	This issue is the fifth official release, which incorporates the following changes: ● Enabled the feature of setting DB instance name on the page for creating a DB instance. ● Supported public accessibility enabling and disabling. ● Supported the feature of viewing slow logs of all shards. ● Supported the changing of DB instance storage space. ● Supported the changing of the minimum capacity of the DB instance to be expanded. ● Supported the creation of a database account for a created DB instance. ● Supported the deletion of node that fails to be added. ● Supported changing the CPU or memory of a DB instance.
2017-12-29	This issue is the fourth official release, which incorporates the following change: Customized the document based on changes made to the DDS console.
2017-11-08	This issue is the third official release, which incorporates the following change: Supported commercial use of DDS.
2017-08-18	This issue is the second official release, which incorporates the following changes: ● Optimized the procedures of buying DB instances based on changes made to the DDS console. ● Optimized the procedures of connecting DB instances using SSL based on changes made to the DDS console.
2017-03-03	This issue is the first official release.

