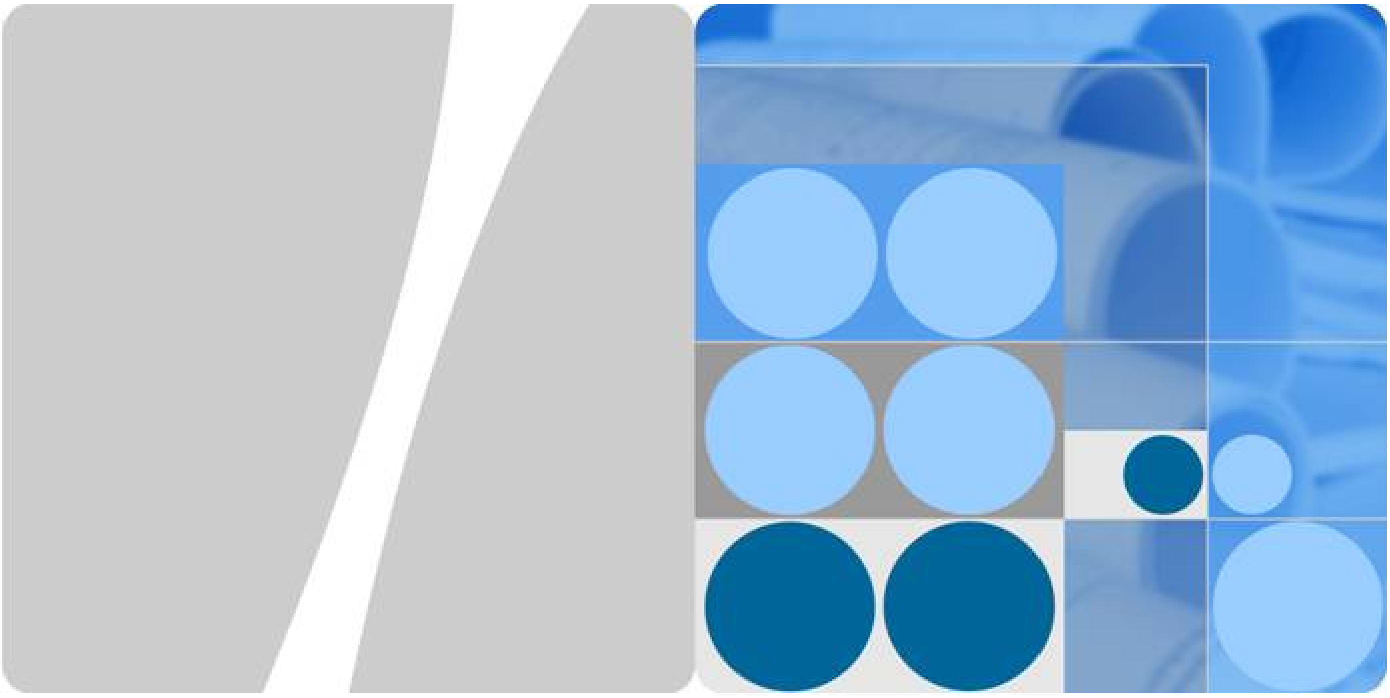


Document Database Service

User Guide

Issue	01
Date	2020-06-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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 Introduction

1.1 What Is DDS?

Document Database Service (DDS), compatible with MongoDB, is a secure, high availability (HA) database service that is reliable, scalable, and easy to use. It provides functions such as one-click deployment, elastic capacity expansion, disaster recovery, backup, restoration, monitoring, and alarm reporting.

DDS has the following features:

- DDS automatically establishes the three-node replica set architecture. It supports fast deployment, high availability (HA) data storage, data redundancy, and failover.
- DDS provides sharded cluster instances comprised of a config node paired with multiple shards and mongos nodes. These clusters can be easily scaled up to enhance read and write performance.
- DDS lets you back up or restore a database from the management console in just a few clicks.
- DDS provides performance metrics and you can configure alarm notifications to make it easier to keep abreast of instance statuses.

For details about the compatible DB engines and versions, see **Database Engine and Version**.

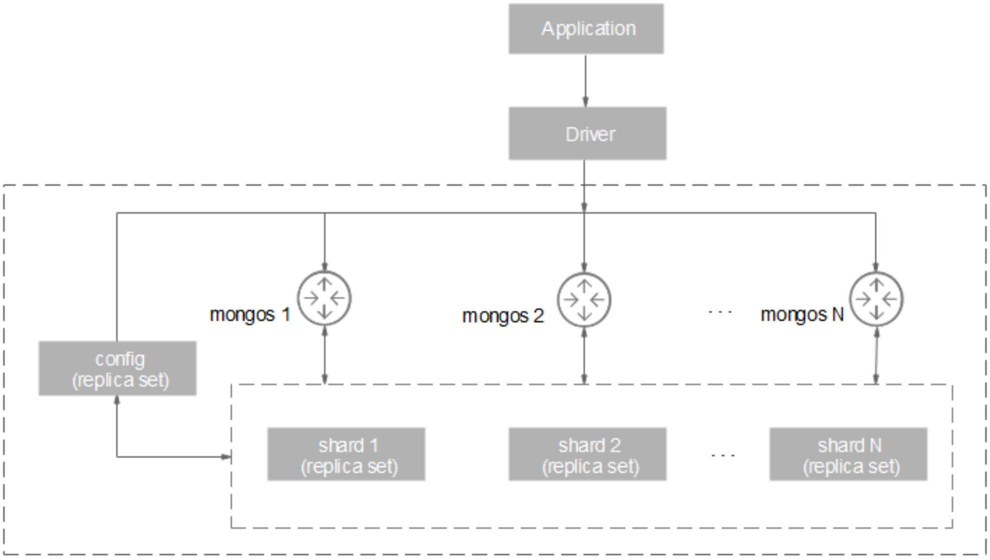
1.2 System Architecture

1.2.1 Cluster

Each DDS cluster is an independent document database. A sharded cluster consists of a config node, and multiple mongos and shard nodes.

Data read and write requests are forwarded by the mongos nodes, which read configuration settings from config, and then allocate the read and write requests to the shards, making it easy to cope with high concurrency scenarios. In addition, each config node, along with the shards in its cluster, are replicated in triplicate to ensure high availability. The following figure shows the DDS cluster architecture.

Figure 1-1 Cluster architecture



- Each mongos is a single node, but you can provision multiple mongos nodes for load balancing and failovers. A single cluster can contain 2 to 12 mongos nodes.
- Each shard is a three-node replica set. A single cluster instance supports 2 to 12 shards.
- A config node is a necessary part of a cluster instance, and is also deployed as a replica set. The config node stores instance configuration data.
- The number of mongos and shard nodes can be increased from the management console. You do not need to use native commands.
- You cannot access data on the config or shard nodes directly. You can only manage the data on the shards though the mongos nodes.
- Currently, a three-node replica set cannot be directly upgraded to a cluster.

1.2.2 Replica Set

A replica set consists of a set of mongod processes. It is a collection of nodes that help ensure data redundancy and reliability.

NOTE

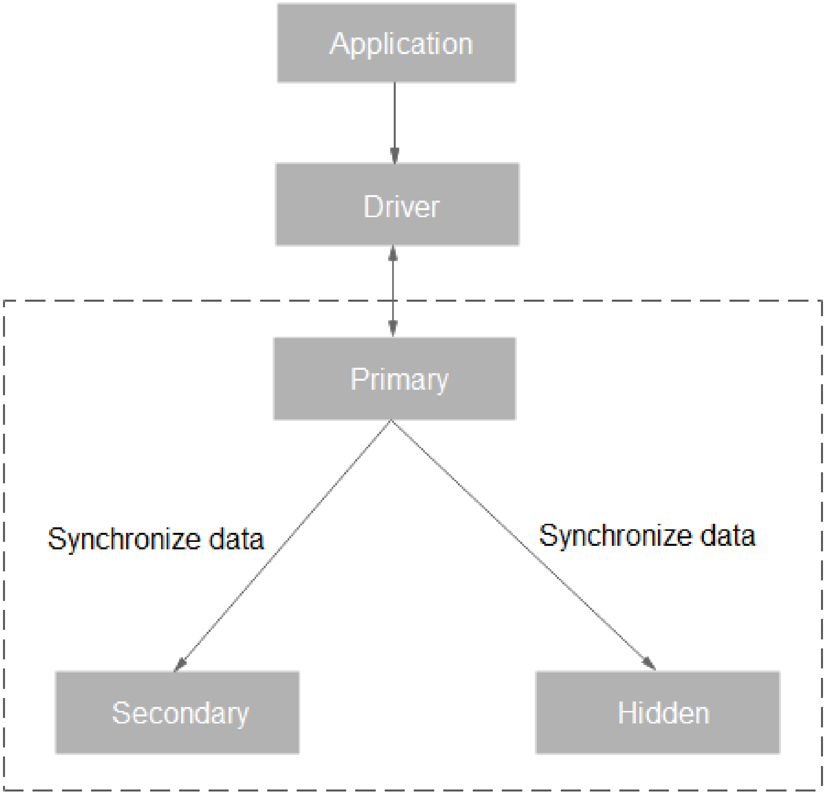
For details about the mongod process, see the MongoDB **official document**.

A replica set consists of three nodes: primary, secondary, and hidden. The three-node architecture is set up automatically, and the three nodes automatically synchronize data with each other to ensure data reliability.

- Primary node: Primary nodes are used to process both read and write requests.
- Secondary node: Secondary nodes are used to process read requests only.
- Hidden node: Hidden nodes are used to back up data.

You can perform operations on the primary and secondary nodes. If a primary node goes down or becomes faulty, the replica set elects a secondary node as a new primary node and continues normal operations. If there are no secondary nodes available, a hidden node will take over the role of the secondary node to ensure high availability. The following figure shows the replica set architecture.

Figure 1-2 Replica set architecture



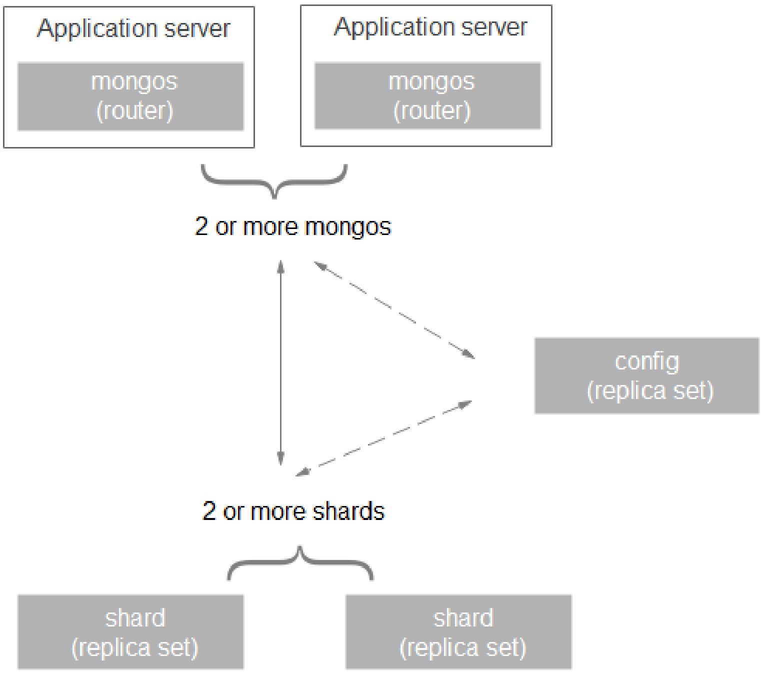
- You can perform multiple management and control tasks, such as creating instances, changing configurations, and backing up instances. The system flexibly controls and tracks tasks, and manages exceptions based on the operations delivered by you.
- DDS collects slow query logs and access control logs, recording DB instance running status.
- You can back up DB instance data and upload it to OBS buckets. Currently, DDS supports automated and manual backup. Automated backups are kept for seven days by default.

1.3 Basic Concepts

1.3.1 Cluster

Each DDS cluster consists of a config node, and multiple mongos and shard nodes. The following diagram shows the node relationships.

Figure 1-3 Diagram of node relationships



mongos

A mongos is a router for reading and writing data, providing a unified interface for accessing DB instances.

- Each cluster instance has 2 to 12 mongos. You can specify the quantity.
- A mongos reads configuration settings from configs and allocates read and write requests to shards. You can connect to a mongos directly.

config

A config stores configuration settings for DB instances and consists of one replica set.

- The availability of a config is the prerequisite to deploying a DB instance or modifying the instance information.
- You cannot connect to a config directly.

shard

In the cluster instance, shards are used to store user data.

- Each cluster instance has 2 to 12 shards. You can specify the quantity.
- Each shard is deployed as a replica set to ensure data redundancy and high reliability.
- You cannot connect to a shard directly.

1.3.2 Database Parameter Group

A database parameter group is a collection of configuration parameters and values and can be applied to multiple DB instances.

1.3.3 Regions and AZs

A region is a geographic area in which resources used by DDS are located.

Each region comprises one or more AZs and is completely isolated from other regions. AZs within the same region can communicate with one another through an internal network, while those in different regions cannot communicate with one another through an internal network.

Cloud service platform data centers are deployed worldwide. DDS applies to different regions. Provisioning DDS to specific regions can better meet user requirements. For example, applications can be designed to better meet specific user requirements or comply with local laws and other demands.

Each region contains many AZs where power and networks are physically isolated. AZs in the same region can communicate with each other over an intranet. Each AZ provides cost-effective and low-latency network connections that are unaffected by faults in other AZs. As a result, provisioning DDS in separate AZs protects your applications against local faults that occur in a single location.

1.4 Functions and Features

Two Architectures

DDS supports two deployment architectures: cluster and replica set, meeting requirements of different service scenarios.

- Cluster

  A cluster consists of three types of nodes: mongos, shard, and config. You can select the number and configuration of mongos and shard nodes to create cluster instances with different levels of service performance.
- Replica set

  DDS automatically builds the replica set architecture, and you can directly operate the primary and secondary nodes. DDS provides you with advanced functions such as high availability (HA) and disaster recovery (DR) switchover, and is invisible to applications.

Elastic Scaling

With the development of your services, you can change CPU and memory specifications of instances, expand storage space, and add mongos and shard nodes in real time. You are advised to perform the change during off-peak hours to avoid the impact of changes on your services.

Key Features

Table 1-1 Key feature description

Features	Description
SLA	99.95%
Instant availability	You can create a DB instance on the management console and access DDS through an Elastic Cloud Server (ECS) to reduce the application response time. If you need to access a DB instance from your local devices, you can bind an elastic IP address (EIP) to the instance.
High compatibility	DDS is a document-oriented NoSQL database. It is fully compatible with MongoDB.
Visualized operation and maintenance (O&M)	You can easily perform restart, backup, and data recovery operations on instances using a graphical user interface (GUI)
Data security	● A security protection system consists of VPCs, subnets, security groups, storage encryption, and DDoS protection, which is capable of defending against various malicious attacks and ensuring data security. ● DDS supports fine-grained permission control.
High availability	The cluster and replica set support high availability. If the primary node is faulty, the secondary node takes over services in a short time. The switchover process is invisible to applications.
Metric monitoring	DDS monitors key performance metrics of DB instances and DB engines in real time, including the CPU usage, memory usage, storage space usage, command execution frequency, delete statement execution frequency, insert statement execution frequency, and number of active connections.
Backups and restorations	● DDS supports automated backup and manual backup. The maximum retention period of an automated backup is 732 days. The manual backup can be retained for a long time. ● DB instances can be restored using backup data.
Setting parameters	DDS allows you to manage parameter groups and modify configuration parameters on the console.

1.5 Typical Application Scenarios

Games

Game players' information generated from game applications, such as players' equipment and bonus points, are stored in DDS databases. During peak hours, DDS cluster instances can handle large amounts of concurrent requests. DDS cluster and replica set provide high availability to ensure the stable running of games in high-concurrency scenarios.

In addition, DDS is compatible with MongoDB and provides the non-schema mode, which frees you from changing table structure when the game play mode changes. DDS can totally meet the flexible gaming requirements. You can store structured data with fixed patterns in RDS, services with flexible patterns in DDS, and hot data in Distributed Cache Service (DCS) to facilitate access to service data and reduce data storage costs.

Advantages:

- Supports embedded documents that eliminate the need to use joins to reduce application development complexity. Flexible schemas also facilitate rapid development and iteration.
- Sharded clusters provide enough capacity to store data into the TB range.

IoT

DDS is compatible with MongoDB and provides the high-performance and asynchronous data write function. In certain scenarios, DDS can process data in the memory database. In addition, cluster instances can dynamically add the number of mongos and shard nodes or upgrade specifications. The performance and storage space can be quickly expanded, making cluster instances suitable for IoT scenarios with high concurrent writes.

Intelligent IoT terminals need to collect various types of data, store device logs, and analyze information in multiple dimensions. In recent years, IoT services have grown rapidly, with huge volumes of data and increasing access traffic that require horizontal expansion capabilities for data storage.

DDS provides the secondary index to meet dynamic query requirements and uses the MapReduce aggregation framework that is compatible with MongoDB to analyze data from multiple dimensions.

Advantages:

- **High Write Performance**: DDS sharded cluster provides high write performance to meet the requirements of terabyte-scale databases.
- **High Performance and Scalability**: DDS supports applications with high QPS rates, and its sharding architecture can be scaled in or out to flexibly cope with application changes.

Internet

DDS replica set uses the three-node HA architecture. Three data nodes form an anti-affinity group and are deployed on different physical servers to automatically synchronize data. The primary and secondary nodes provide services. Each node has a private IP address and works with Driver to allocate read workloads.

Many organizations need to process and store data into the TB range, requiring data to be written to databases in real time and dynamic analysis capabilities in big data computing.

Advantages:

- **MapReduce:** With a complete data analysis utility, you can query statements or scripts, and distribute requests to DDS.
- **Excellent Scalability**: DDS DB instances can be scaled up to support growing services and data volumes in content management systems.

1.6 DDS DB Instance

1.6.1 DB Instance Specifications

DB instance specifications are listed in the following table.

NOTE

The DB instance specifications depend on service requirements.

You can change the maximum number of connections of a DB instance by modifying the **net.maxIncomingConnections** parameter. For details about how to change the parameter value, see **Editing a Parameter Group**.

Cluster

For details about the cluster instance specifications, see **Table 1-2** and **Table 1-3**.

Table 1-2 config specifications

vCPUs	Memory (GB)	Default Maximum Connections
2	4	1000

Table 1-3 shard and mongos specifications

vCPUs	Memory (GB)	Default Maximum Connections
2	4	400
2	8	400
4	8	1000
4	16	1000
8	16	4000

vCPUs	Memory (GB)	Default Maximum Connections
8	32	4000
16	32	8000
16	64	8000

Replica Set

The specifications supported by a replica set are the same as those supported by shard and mongos. For details, see **Table 1-3**.

1.6.2 Database Engine and Version

Currently, DDS is compatible with MongoDB 3.4 and 4.0 Community Edition and supports the WiredTiger storage engine, so you need to use a driver compatible with MongoDB 3.0 or later to access DDS.

1.6.3 DB Instance Status

DB Instance Status

Table 1-4 Status and description

Status	Description
Available	A DB instance is running properly.
Abnormal	A DB instance is faulty.
Creating	A DB instance is being created.
Creation failed	A DB instance fails to be created.
Restarting	A DB instance is being restarted because of a customer request or a modification that requires restarting it for the modification to take effect.
Switchover in progress	The primary and secondary nodes of a replica set instance are being switched.
Adding node	shards or mongos are being added to a DDS cluster instance.
Deleting node	The node that failed to be added is being deleted.
Scaling up	The storage space of instance nodes is being expanded.
Changing instance class	The CPU or memory of a DB instance is being changed.
Backing up	A backup file is being created.

Status	Description
Checking restoration	The backup of the current DB instance is being restored to a new DB instance.
Changing private IP address	The private IP address of a node is being changed.
Changing port	The DB instance port is being changed.
Changing a security group	The security group is being changed.

Parameter Group Status

Table 1-5 Status and description

Status	Description
In-Sync	A database parameter change has taken effect.
Pending restart	A database parameter is waiting for the DB instance to be restarted before its modification takes effect.

1.7 Constraints

To improve the stability and security of DB instances, there are some constraints on the use of DDS. For details, see **Table 1-6**.

Table 1-6 Function constraints

Operation	Constraints
Connecting to a DB instance	● When connecting to a DB instance over private networks, you need to bind an EIP to the prepared ECS and ensure that the DB instance and the ECS are in the same region, AZ, and VPC subnet. ● By default, DDS is not accessible from ECSs that are not in the same security group. If the ECS is not in the same group, you need to add an inbound rule to enable access. ● The default DDS port is 8635, but this port can be modified if necessary.
Deployment	ECSs in which DB instances are deployed are not visible to you. Your applications can access the database only through an IP address and port.

Operation	Constraints
Obtaining permissions of user **rwuser**	Only the **rwuser** user permissions are provided on the instance creation page. For details about the related commands, see **Which Commands are Supported or Restricted by DDS?**
Migrating data	You can use command line tools, including mongoexport and mongoimport, to migrate data. For details, see section **Migrating Data**.
Storage engine	Currently, DDS supports the WiredTiger storage engine.
Restarting a DB instance or a node	DB instances cannot be restarted using commands. They must be restarted on the RDS console.

1.8 Related Services

Table 1-7 Related services

Service Name	Function
Elastic Cloud Service (ECS)	ECS provides DDS with elastic computing resources and a running environment for DB instances.
Virtual Private Cloud (VPC)	VPC provides DDS with elastic network resources and implements network isolation and access control for your DB instances.
Object Storage Service (OBS)	OBS stores your DDS DB instance backup files.
Identity and Access Management (IAM)	IAM provides the permission management function for DDS.

2 Logging In to the DDS Console

Prerequisites

You have registered an account.

For the first time you use DDS, apply for an account at the official website. After the application is successful, your account has permissions to access the DDS service, as well as all other cloud services.

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 **Database**, click **Document Database Service** to go to the DDS console.

----End

3 Getting Started with Clusters

3.1 Connection Methods

You can access DDS over private or public networks.

Table 3-1 Connection methods

Method	IP Address	Scenario	Description
Private network	Private IP address	DDS provides a private IP address by default. ● If your applications are running on an ECS that is in the same region, AZ, and VPC subnet as your DDS DB instance, you are advised to use a private IP address to connect the ECS to your DDS DB instances. ● By default, DDS is not accessible from ECSs that are not in the same security group. If the ECS is not in the same group, you need to add an inbound rule to enable access. ● The default DDS port is 8635, but this port can be modified if necessary.	Secure and excellent performance

Method	IP Address	Scenario	Description
Public network	EIP	● If your applications are running on an ECS that is in a different region from the one where the DB instance is located, you are advised to use an EIP to connect the ECS to your DDS DB instances. ● If your applications are deployed on another cloud platform, EIP is recommended.	● Low security ● For faster transmission and improved security, you are advised to migrate your applications to an ECS that is in the same subnet as your DDS instance and use a private IP address to access the instance.

3.2 Connecting to a Cluster Instance Over Private Networks

3.2.1 Overview

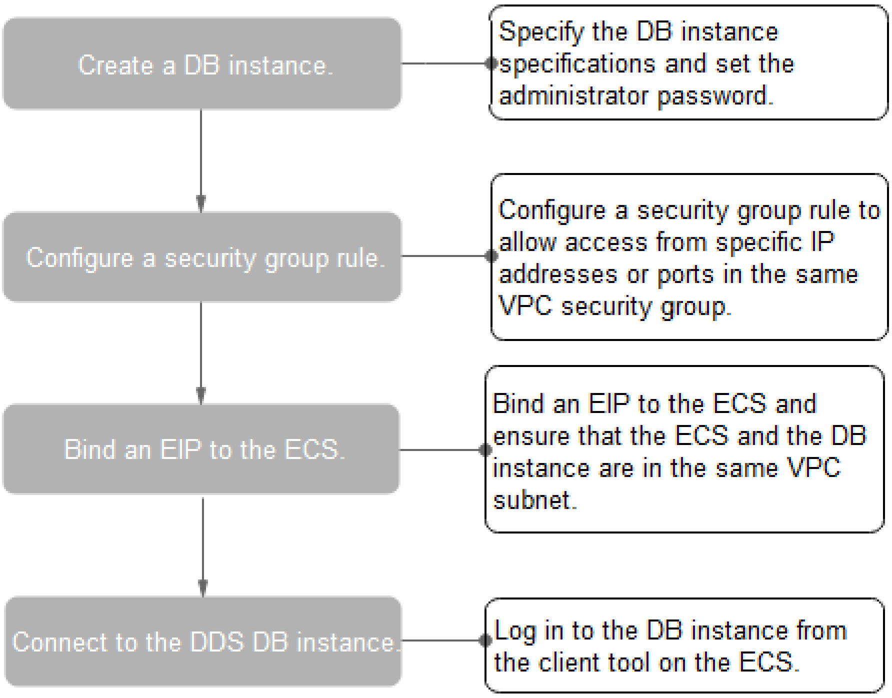
Scenarios

This section describes how to create a cluster instance on the management console, set a security group, and connect to a cluster instance over private networks.

Process

The following describes the steps from creating a DB instance to using it.

Figure 3-1 Accessing DB instances from a private network


Create a DB instance.
Specify the DB instance specifications and set the administrator password.
Configure a security group rule.
Configure a security group rule to allow access from specific IP addresses or ports in the same VPC security group.
Bind an EIP to the ECS.
Bind an EIP to the ECS and ensure that the ECS and the DB instance are in the same VPC subnet.
Connect to the DDS DB instance.
Log in to the DB instance from the client tool on the ECS.

3.2.2 Creating a Cluster Instance

Scenarios

This section describes how to create a Community Edition cluster instance on the DDS management console. DDS allows you to tailor your compute resources and storage space to your business needs.

You can use your account to create up to 10 cluster instances.

You can use your account to create up to 10 cluster instances. To create more cluster instances, click ◔ in the upper right corner of the management console. On the **Service Quota** page, click **Increase Quota** to apply for quotas.

Procedure

Step 1 Log in to the DDS console.

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

Step 3 On the displayed page, select your DB instance specifications and click **Create Now**.

Table 3-2 Basic information

Parameter	Description
Region	A region where the tenant is located. It can be changed in the upper left corner. For details, see section **Regions and AZs**. **NOTE** DB instances deployed in different regions cannot communicate with each other through a private network, and you cannot change the region of a DB instance once it is created. Exercise caution when selecting a region.
DB Instance Name	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 (_). After the DB instance is created, you can change the DB instance name. For details, see section **Changing a DB Instance Name**.
Database Type	Community Edition
DB Instance Type	Select **Cluster**. A cluster instance includes three types of nodes: mongos, shard, and config. Each shard and config is a three-node replica set to ensure high availability.
Compatible MongoDB Version	● 4.0 ● 3.4
Storage Type	Ultra-high I/O
Storage Engine	WiredTiger
Disk Encryption	● **Disabled**: Disable the encryption function. ● **Enabled**: Enable the encryption function. This feature improves data security but slightly affects read/write performance. **Key Name**: Select or create a private key, which is the tenant key. **NOTE** – After a DB instance is created, the disk encryption status and the key cannot be changed. The backup data stored in OBS is not encrypted. – The key cannot be disabled or deleted when being used. Otherwise, the database becomes unavailable. – For details about how to create a key, see the "Creating a CMK" section in the *Key Management Service User Guide*.

Table 3-3 Specifications

Parameter	Description
Specifications	In the x86 CPU architecture, the following specifications can be selected to suit different application scenarios: General-purpose (s6), Enhanced (c3), and Enhanced II (c6).
mongos class	For details about the mongos CPU and memory, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Cluster DB Instance**.
mongos quantity	The value ranges from 2 to 12. After a DB instance is created, you can add nodes. For details, see section **Adding Cluster Instance Nodes**.
mongos parameter group	The parameters that apply to the mongos nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.
shard class	For details about the shard CPU and memory, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Cluster DB Instance**.
shard storage space	The value ranges from 10 GB to 2000 GB and must be a multiple of 10. After a DB instance is created, you can scale up its storage space. For details, see section **Scaling Up Storage Space**.
shard quantity	The number of shard nodes. The shard node stores user data but cannot be accessed directly. The value ranges from 2 to 12. After a DB instance is created, you can add nodes. For details, see section **Adding Cluster Instance Nodes**.
shard parameter group	The parameters that apply to the shard nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.
config class	The CPU and memory of a config node. The config node stores the DB instance configurations but cannot be accessed directly. For details, see **DB Instance Specifications**.
config storage space	The storage space is 20 GB and cannot be scaled up.

Parameter	Description
config parameter group	The parameters that apply to the config nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.

Table 3-4 Network

Parameter	Description
VPC	The VPC where your DB instances are located. A VPC isolates networks for different services, so you can easily manage and configure internal networks and change network configuration. For details about how to create a VPC, see section "Creating a VPC" in the *Virtual Private Cloud User Guide*. For details about the constraints on the use of VPCs, see **Connection Methods**. **NOTE** After the DDS instance is created, the VPC cannot be changed.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. After the instance is created, you can change the private IP address assigned by the subnet. For details, see **Changing a Private IP Address**.
Security Group	A security group controls access between DDS and other services for security. **NOTE** Ensure that the security group rule you set allows clients to access DB instances. For example, select the TCP protocol with inbound direction, input the default port number **8635**, and enter a subnet IP address or select a security group that the DB instance belongs to.
SSL	Secure Sockets Layer (SSL) certificates set up encrypted connections between clients and servers, preventing data from being tampered with or stolen during transmission. You can enable SSL to improve data security. After a DB instance is created, you can connect to it using SSL.

Table 3-5 Database configuration

Parameter	Description
Administrator	The default account is **rwuser**.

Parameter	Description
Administrator Password	Set a password for the administrator. 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: ~!@#%^*-_=+? Keep this password secure. If lost, the system cannot retrieve it for you.
Confirm Password	Enter the administrator password again.

Table 3-6 Tag

Parameter	Description
Tags	This setting is optional. Adding tags helps you better identify and manage your DB instances. Up to 10 tags can be added for a DB instance. A tag is composed of a key-value pair. ● Key: Mandatory if the DB instance is going to be tagged – 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 (-). ● Value: Optional if the DB instance is going to be tagged – The value consists of up to 43 characters. – The value can only consist of digits, letters, underscores (_), dots (.), and hyphens (-). After a DB instance is created, you can view its tag details on the **Tags** tab. In addition, you can add, modify, and delete tags for existing DB instances. For details, see **Tag**.

NOTE

DB instance performance is determined by how you configure it during the creation. The hardware configuration items that can be selected include the node class and storage space.

Step 4 On the displayed page, confirm the DB instance information.

- If you need to modify the specifications, click **Previous** to return to the previous page.
- If you do not need to modify the specifications, click **Submit** to start the instance creation.

Step 5 After a DDS DB instance is created, you can view and manage it on the **Instance Management** page.

- When a DB instance is being created, the status displayed in the **Status** column is **Creating**. This process takes about 15 minutes. After the creation is complete, the status changes to **Available**.
- DDS enables the automated backup policy by default. After a DB instance is created, you can modify or disable the automated backup policy. An automated full backup is immediately triggered after the creation of a DB instance.
- The default DDS port is 8635, but this port can be modified if necessary. If you change the port, you need to add the security group rule to enable access.

----End

3.2.3 Setting a Security Group

Scenarios

This section explains how to add a security group rule to control access to and from the DDS DB instances in a security group.

Precautions

The default security group rule allows all outgoing data packets. ECSs and DDS DB instances in the same security group can access each other. After a security group is created, you can create different rules for that security group, which allows you to control access to the DB instances that are in it.

To access a DB instance in a security group from a source outside of that group, you need to create an inbound rule.

For details about the constraints on the using security groups, see "Security Group Overview" in the *Virtual Private Cloud User Guide*.

Procedure

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Security Group** area, on the **Inbound Rules** tab, click **Add Rule**. In the displayed **Add Inbound Rule** dialog box, set required parameters to add inbound rules. On the **Outbound Rules** tab, click **Add Rule**. In the displayed **Add Outbound Rule** dialog box, set required parameters to add outbound rules.

You can click ⊕ to add more rules.

Step 4 Add a security group rule as prompted.

Table 3-7 Parameter description

Parameter	Description	Value Example
Protocol	The network protocol required for access. You can allow all protocols or specify a specific protocol, TCP, UDP, ICMP, and SSH.	TCP
Port	Specifies the port that allows the access to ECSs or external devices.	8635
Source/ Destination	Specifies the supported IP address and security group that the rule applies to. ● **IP address**: The IP address or subnet that the rule applies to. Single IP addresses must be expressed using slash notation. – Single IP address: xxx.xxx.xxx.xxx/32 (IPv4) – Subnet: xxx.xxx.xxx.0/24 – All IP addresses: 0.0.0.0/0 ● **Security group**: A security group that access will be allowed from. ECSs in this security group will be granted access to DDS instance in the current security group.	● 192.168.10.0/24 ● default

Step 5 Click **OK**.

----End

3.2.4 Connecting to a Cluster Instance Over Private Networks

Scenarios

This section describes how to connect to a cluster instance using the MongoDB client over private networks.

The MongoDB client can connect to a DB instance with a common connection or an encrypted connection (SSL). To improve data transmission security, you are advised to connect to DB instances using the SSL connection.

Different OS scenarios: The following uses Linux ECS and Window client as an example.

Constraints

For details about constraints on connecting to a DB instance, see **Constraints**.

Prerequisites

1. For details on how to create and log in to an ECS, see "Creating and Logging In to a Windows ECS" or "Creating and Logging In to a Linux ECS" in the *Elastic Cloud Server User Guide*.
2. Install the MongoDB client on the ECS.

   For details on how to install a MongoDB client, see section **How Can I Install a MongoDB Client?**

   > **NOTE**
   >
   > If you use a **connection address** to connect to a cluster instance, download the MongoDB client of version later than 4.0.

Connecting to a DB Instance Using the MongoDB Client (SSL)

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, click ⤓ next to the **SSL** field.

Step 4 Upload the root certificate to the ECS to be connected to the DB instance.

The following describes how to upload the certificate to a Linux and Window ECS:

- In Linux, run the following command:

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

  > **NOTE**
  >
  > - **IDENTITY_FILE** indicates the directory where the root certificate locates. 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 root certificate is uploaded.

- In Windows, upload the root certificate using the remote connection tool.

Step 5 Connect to the DB instance in the directory where the MongoDB client is located.

- Method 1: Using Linux commands

  ```
  ./mongo --host <DB_HOST> --port <DB_PORT> -u <DB_USER> -p --authenticationDatabase admin --ssl --sslCAFile <FILE_PATH> --sslAllowInvalidHostnames
  ```

  Enter the database account password when prompted:

  ```
  Enter password:
  ```

- Method 2: Using the private connection address. Enter the IP addresses and ports based on the number of DB instance nodes.

  ```
  ./mongo mongodb://rwuser:****@<DB_HOST>:<DB_PORT>,<DB_HOST>:<DB_PORT>/test?authSource=admin --ssl --sslCAFile <FILE_PATH> --sslAllowInvalidHostnames
  ```

The connection information can be obtained in the **Address** column on the **Instance Management** page.

A connection address indicates that one of the mongos nodes will be randomly connected. If you use this method to connect to a DB instance, use the MongoDB client of version later than 4.0.

NOTE

- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **Private IP Address** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- ****** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).
- **FILE_PATH** indicates the path where the root certificate is stored.

- Connect to the instance using Linux commands. The following is an example command:

  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames

- Connect to the DB instance using the private connection address. The following is an example command:

  ./mongo mongodb://rwuser:**@192.168.1.6:8635/test?authSource=admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames**

Step 6 Check the connection result. If the following information is displayed, the connection is successful.

```
mongos>
```

----End

Connecting to a DB Instance Using the MongoDB Client (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Connect to the ECS.

Step 2 Connect to the DB instance in the directory where the MongoDB client is located.

- Method 1: Using Linux commands

  ./mongo --host <*DB_HOST*> **--port** <*DB_PORT*> **-u** <*DB_USER*> **-p --authenticationDatabase admin**

  Enter the database account password when prompted:

```
Enter password:
```

- Method 2: Using the private connection address. Enter the IP addresses and ports based on the number of DB instance nodes.

  ```
  ./mongo mongodb://rwuser:****@<DB_HOST1>:<DB_PORT1>,<DB_HOST2>:<DB_PORT2>/test?authSource=admin
  ```

  The connection information can be obtained in the **Address** column on the **Instance Management** page.

  A connection address indicates that one of the mongos nodes will be randomly connected. If you use this method to connect to a DB instance, use the MongoDB client of version later than 4.0.

NOTE

- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **Private IP Address** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- ********** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).

- Connect to the instance using Linux commands. The following is an example command:

  ```
  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin
  ```

- Connect to the DB instance using the private connection address. The following is an example command:

  ```
  ./mongo mongodb://rwuser:****@192.168.1.6:8635/test?authSource=admin
  ```

Step 3 Check the connection result. If the following information is displayed, the connection is successful.

```
mongos>
```

----End

3.3 Connecting to a Cluster Instance Over Public Networks

3.3.1 Overview

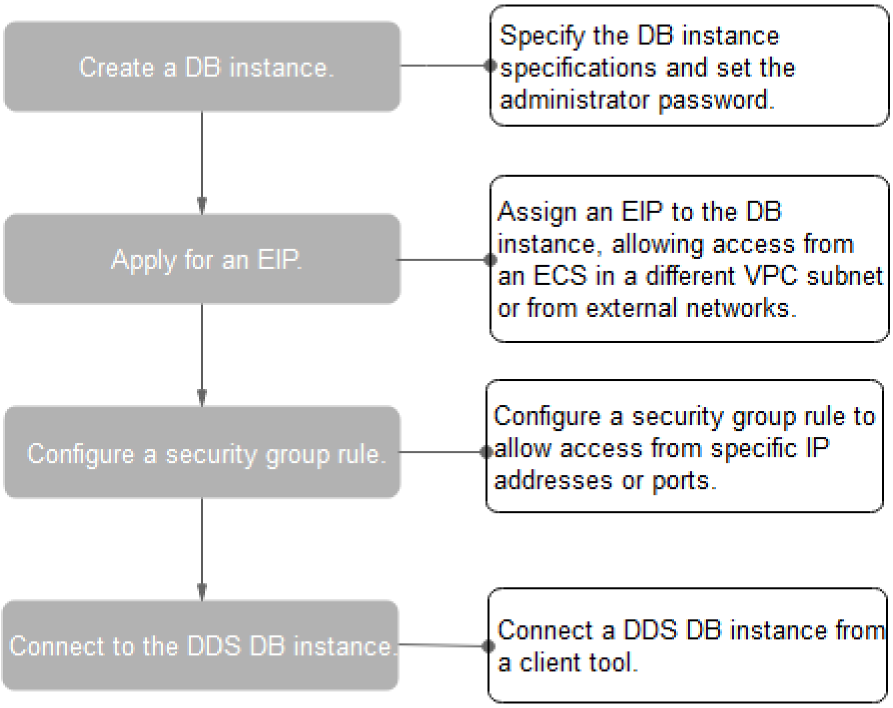
Scenarios

This section describes how to create a cluster instance on the management console, set a security group, bind an EIP, and connect to a cluster instance over public networks.

Process

The following describes the steps from creating a DB instance to using it.

Figure 3-2 Accessing DB instances from a public network


Create a DB instance.
Specify the DB instance specifications and set the administrator password.
Apply for an EIP.
Assign an EIP to the DB instance, allowing access from an ECS in a different VPC subnet or from external networks.
Configure a security group rule.
Configure a security group rule to allow access from specific IP addresses or ports.
Connect to the DDS DB instance.
Connect a DDS DB instance from a client tool.

3.3.2 Creating a Cluster Instance

Scenarios

This section describes how to create a Community Edition cluster instance on the DDS management console. DDS allows you to tailor your compute resources and storage space to your business needs.

You can use your account to create up to 10 cluster instances.

You can use your account to create up to 10 cluster instances. To create more cluster instances, click in the upper right corner of the management console. On the **Service Quota** page, click **Increase Quota** to apply for quotas.

Procedure

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

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

Step 3 On the displayed page, select your DB instance specifications and click **Create Now**.

Table 3-8 Basic information

Parameter	Description
Region	A region where the tenant is located. It can be changed in the upper left corner. For details, see section **Regions and AZs**. **NOTE** DB instances deployed in different regions cannot communicate with each other through a private network, and you cannot change the region of a DB instance once it is created. Exercise caution when selecting a region.
DB Instance Name	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 (_). After the DB instance is created, you can change the DB instance name. For details, see section **Changing a DB Instance Name**.
Database Type	Community Edition
DB Instance Type	Select **Cluster**. A cluster instance includes three types of nodes: mongos, shard, and config. Each shard and config is a three-node replica set to ensure high availability.
Compatible MongoDB Version	● 4.0 ● 3.4
Storage Type	Ultra-high I/O
Storage Engine	WiredTiger
Disk Encryption	● **Disabled**: Disable the encryption function. ● **Enabled**: Enable the encryption function. This feature improves data security but slightly affects read/write performance. **Key Name**: Select or create a private key, which is the tenant key. **NOTE** – After a DB instance is created, the disk encryption status and the key cannot be changed. The backup data stored in OBS is not encrypted. – The key cannot be disabled or deleted when being used. Otherwise, the database becomes unavailable. – For details about how to create a key, see the "Creating a CMK" section in the *Key Management Service User Guide*.

Table 3-9 Specifications

Parameter	Description
Specifications	In the x86 CPU architecture, the following specifications can be selected to suit different application scenarios: General-purpose (s6), Enhanced (c3), and Enhanced II (c6).
mongos class	For details about the mongos CPU and memory, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Cluster DB Instance**.
mongos quantity	The value ranges from 2 to 12. After a DB instance is created, you can add nodes. For details, see section **Adding Cluster Instance Nodes**.
mongos parameter group	The parameters that apply to the mongos nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.
shard class	For details about the shard CPU and memory, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Cluster DB Instance**.
shard storage space	The value ranges from 10 GB to 2000 GB and must be a multiple of 10. After a DB instance is created, you can scale up its storage space. For details, see section **Scaling Up Storage Space**.
shard quantity	The number of shard nodes. The shard node stores user data but cannot be accessed directly. The value ranges from 2 to 12. After a DB instance is created, you can add nodes. For details, see section **Adding Cluster Instance Nodes**.
shard parameter group	The parameters that apply to the shard nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.
config class	The CPU and memory of a config node. The config node stores the DB instance configurations but cannot be accessed directly. For details, see **DB Instance Specifications**.
config storage space	The storage space is 20 GB and cannot be scaled up.

Parameter	Description
config parameter group	The parameters that apply to the config nodes. After a DB instance is created, you can change the parameter group of a node to bring out the best performance. For details, see **Parameter Group Settings**.

Table 3-10 Network

Parameter	Description
VPC	The VPC where your DB instances are located. A VPC isolates networks for different services, so you can easily manage and configure internal networks and change network configuration. For details about how to create a VPC, see section "Creating a VPC" in the *Virtual Private Cloud User Guide*. For details about the constraints on the use of VPCs, see **Connection Methods**. **NOTE** After the DDS instance is created, the VPC cannot be changed.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. After the instance is created, you can change the private IP address assigned by the subnet. For details, see **Changing a Private IP Address**.
Security Group	A security group controls access between DDS and other services for security. **NOTE** Ensure that the security group rule you set allows clients to access DB instances. For example, select the TCP protocol with inbound direction, input the default port number **8635**, and enter a subnet IP address or select a security group that the DB instance belongs to.
SSL	Secure Sockets Layer (SSL) certificates set up encrypted connections between clients and servers, preventing data from being tampered with or stolen during transmission. You can enable SSL to improve data security. After a DB instance is created, you can connect to it using SSL.

Table 3-11 Database configuration

Parameter	Description
Administrator	The default account is **rwuser**.

Parameter	Description
Administrator Password	Set a password for the administrator. 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: ~!@#%^*-_=+? Keep this password secure. If lost, the system cannot retrieve it for you.
Confirm Password	Enter the administrator password again.

Table 3-12 Tag

Parameter	Description
Tags	This setting is optional. Adding tags helps you better identify and manage your DB instances. Up to 10 tags can be added for a DB instance. A tag is composed of a key-value pair. ● Key: Mandatory if the DB instance is going to be tagged – 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 (-). ● Value: Optional if the DB instance is going to be tagged – The value consists of up to 43 characters. – The value can only consist of digits, letters, underscores (_), dots (.), and hyphens (-). After a DB instance is created, you can view its tag details on the **Tags** tab. In addition, you can add, modify, and delete tags for existing DB instances. For details, see **Tag**.

NOTE

DB instance performance is determined by how you configure it during the creation. The hardware configuration items that can be selected include the node class and storage space.

Step 4 On the displayed page, confirm the DB instance information.

- If you need to modify the specifications, click **Previous** to return to the previous page.
- If you do not need to modify the specifications, click **Submit** to start the instance creation.

Step 5 After a DDS DB instance is created, you can view and manage it on the **Instance Management** page.

- When a DB instance is being created, the status displayed in the **Status** column is **Creating**. This process takes about 15 minutes. After the creation is complete, the status changes to **Available**.
- DDS enables the automated backup policy by default. After a DB instance is created, you can modify or disable the automated backup policy. An automated full backup is immediately triggered after the creation of a DB instance.
- The default DDS port is 8635, but this port can be modified if necessary. If you change the port, you need to add the security group rule to enable access.

----End

3.3.3 Binding an EIP

Scenarios

After you create a DB instance, you can bind it to an EIP to allow external access. If later you want to prohibit external access, you can also unbind the EIP from the DB instance.

Precautions

- Before accessing a database, you need to apply for an EIP on the VPC console. Then, add an inbound rule to allow the IP addresses or IP address ranges of ECSs. For details, see section **Setting a Security Group**.
- In the cluster instance, only mongos can be bound to an EIP. To change the EIP that has been bound to a node, you need to unbind it from the node first.

Binding an EIP

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

Step 2 In the navigation pane on the left, choose **Connections**. In the **Basic Information** area, locate the target mongos node and click **Bind EIP** in the **Operation** column.

In the **Node Information** area on the **Basic Information** page, locate the target mongos node and choose **More** > **Bind EIP** in the **Operation** column.

Step 3 In the displayed dialog box, all available unbound EIPs are listed. Select the required EIP and click **OK**. If no available EIPs are displayed, click **View EIP** and create an EIP on the VPC console.

Step 4 In the **EIP** column on the **mongos** tab, view the EIP that is successfully bound.

To unbind an EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

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

Step 2 In the navigation pane on the left, choose **Connections**. In the **Basic Information** area, locate the target mongos node and click **Unbind EIP** in the **Operation** column.

In the **Node Information** area on the **Basic Information** page, locate the target mongos node and choose **More** > **Unbind EIP** in the **Operation** column.

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

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

3.3.4 Setting a Security Group

Scenarios

This section explains how to add a security group rule to control access to and from the DDS DB instances in a security group.

Precautions

The default security group rule allows all outgoing data packets. ECSs and DDS DB instances in the same security group can access each other. After a security group is created, you can create different rules for that security group, which allows you to control access to the DB instances that are in it.

To access a DB instance in a security group from a source outside of that group, you need to create an inbound rule.

For details about the constraints on the using security groups, see "Security Group Overview" in the *Virtual Private Cloud User Guide*.

Procedure

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Security Group** area, on the **Inbound Rules** tab, click **Add Rule**. In the displayed **Add Inbound Rule** dialog box, set required parameters to add inbound rules. On the **Outbound Rules** tab, click **Add Rule**. In the displayed **Add Outbound Rule** dialog box, set required parameters to add outbound rules.

You can click ⊕ to add more rules.

Step 4 Add a security group rule as prompted.

Table 3-13 Parameter description

Parameter	Description	Value Example
Protocol	The network protocol required for access. You can allow all protocols or specify a specific protocol, TCP, UDP, ICMP, and SSH.	TCP
Port	Specifies the port that allows the access to ECSs or external devices.	8635
Source/ Destination	Specifies the supported IP address and security group that the rule applies to. ● **IP address**: The IP address or subnet that the rule applies to. Single IP addresses must be expressed using slash notation. – Single IP address: xxx.xxx.xxx.xxx/32 (IPv4) – Subnet: xxx.xxx.xxx.0/24 – All IP addresses: 0.0.0.0/0 ● **Security group**: A security group that access will be allowed from. ECSs in this security group will be granted access to DDS instance in the current security group.	● 192.168.10.0/24 ● default

Step 5 Click **OK**.

----End

3.3.5 Connecting to a Cluster Instance Over Public Networks

Scenarios

This section describes how to connect to a cluster instance using the MongoDB client and Robo 3T over public networks.

The MongoDB client and Robo 3T can connect to a DB instance with a common connection or an encrypted connection (SSL). To improve data transmission security, you are advised to connect to DB instances using the SSL connection.

Different OS scenarios: The following uses Linux ECS and Window client as an example.

Prerequisites

1. **Bind an EIP** to the cluster instance and **set security group rules** to ensure that the EIP can be accessed through the ECS or Robo 3T.
2. Install the MongoDB client or Robo 3T.

   MongoDB client

a. For details on how to create and log in to an ECS, see "Creating and Logging In to a Windows ECS" or "Creating and Logging In to a Linux ECS" in the *Elastic Cloud Server User Guide*.

b. Install the MongoDB client on the ECS.

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

NOTE

If you use a **connection address** to connect to a cluster instance, download the MongoDB client of version later than 4.0.

Robo 3T

For details on how to install Robo 3T, see **How Do I Install Robo 3T?**

3. If you select the SSL mode, download the SSL certificate on the DDS console.

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

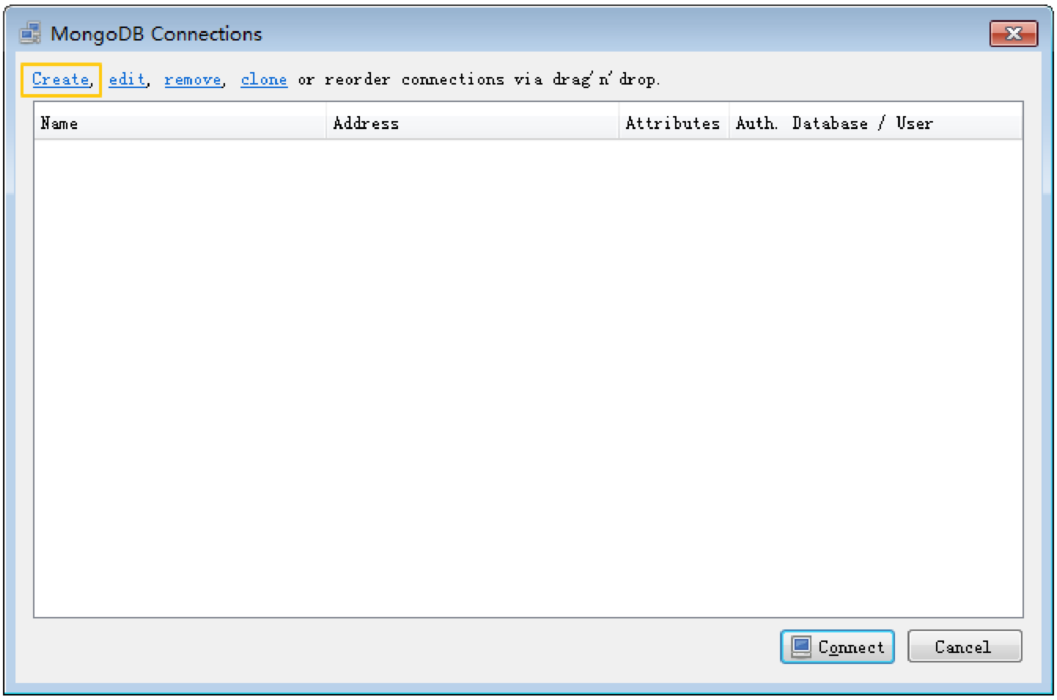
    b. In the navigation pane on the left, choose **Connections**.

    c. In the **Basic Information** area, click ⤓ next to the **SSL** field.

Connecting to a DB Instance Using Robo 3T (SSL)

Step 1 Run the installed Robo 3T. On the displayed dialog box, click **Create**.

Figure 3-3 Connections



MongoDB Connections

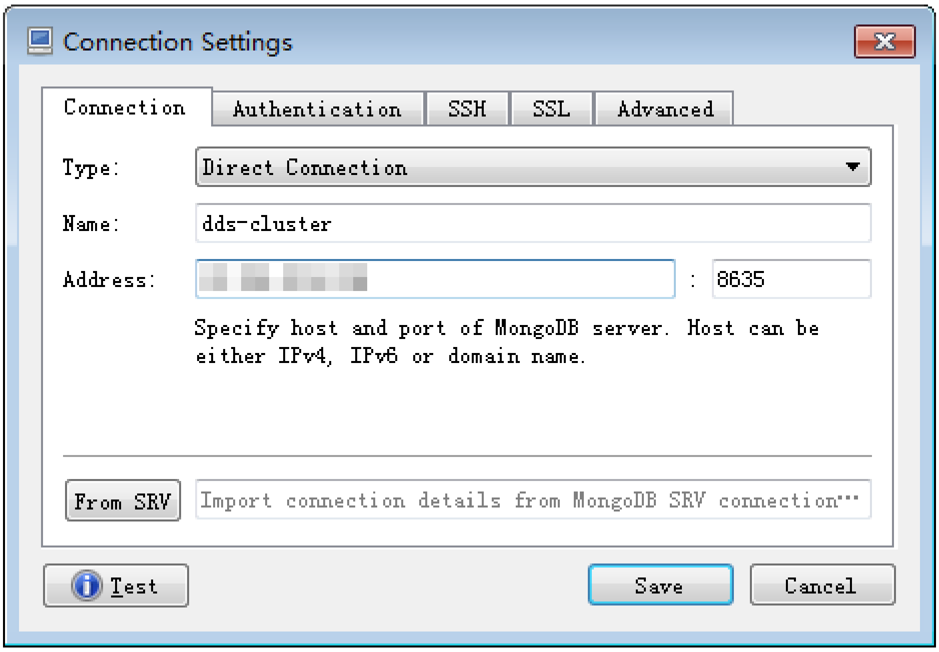
Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User

Step 2 In the **Connection Settings** dialog box, set the parameters of the new connection.

1. On the **Connection** tab, enter the name of the new connection in the **Name** text box and enter the EIP and database port that are bound to the DDS DB instance in the **Address** text box.

Figure 3-4 Connection



Type: Direct Connection

Name: dds-cluster

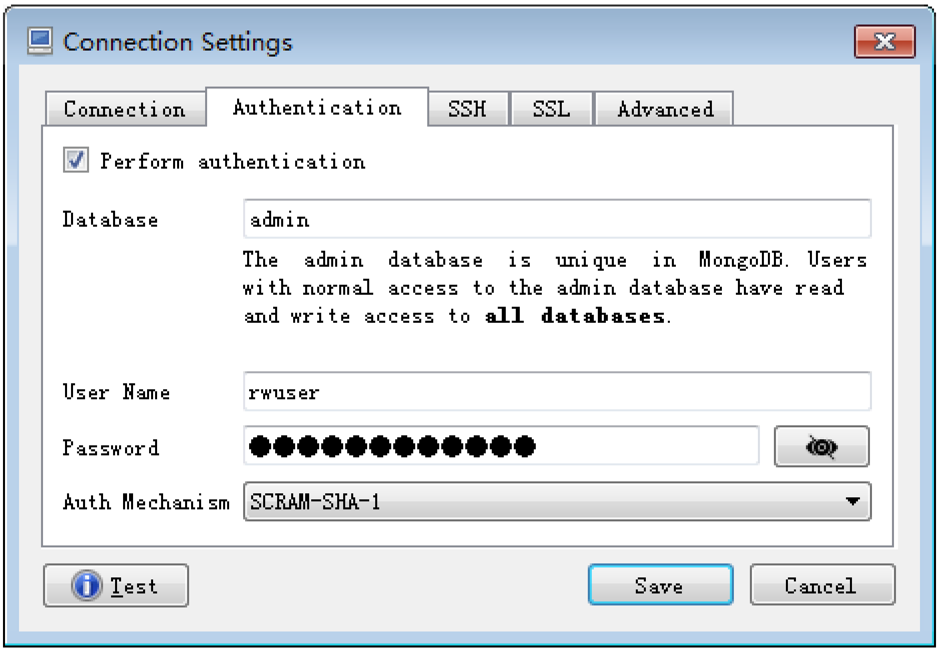
Address: [redacted] : 8635

Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

Import connection details from MongoDB SRV connection…

2. On the **Authentication** tab, set **Database** to **admin**, **User Name** to **rwuser**, and **Password** to the administrator password you set during the creation of the cluster instance.

Figure 3-5 Authentication



☑ Perform authentication

Database: admin

The admin database is unique in MongoDB. Users with normal access to the admin database have read and write access to **all databases**.

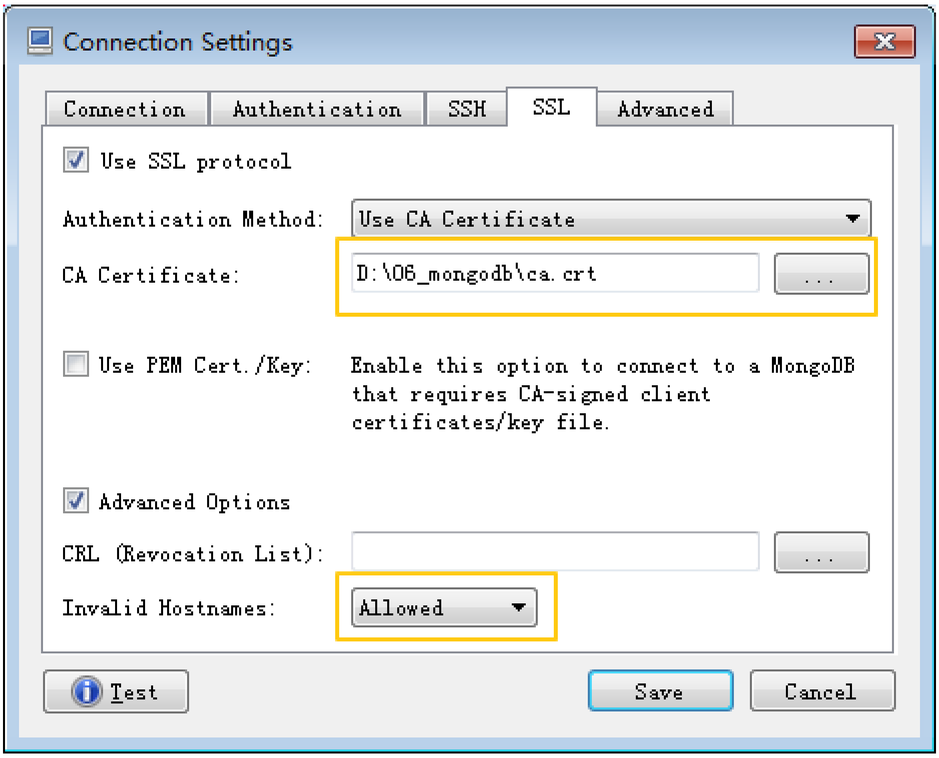
User Name: rwuser

Password: ●●●●●●●●●●●●

Auth Mechanism: SCRAM-SHA-1

3. On the **SSL** tab, upload the SSL certificate and select **Allowed** for **Invalid Hostnames**.

Figure 3-6 SSL



Connection Settings

☑ Use SSL protocol

Authentication Method: Use CA Certificate

CA Certificate: D:\06_mongodb\ca.crt ...

□ Use PEM Cert./Key: Enable this option to connect to a MongoDB that requires CA-signed client certificates/key file.

☑ Advanced Options

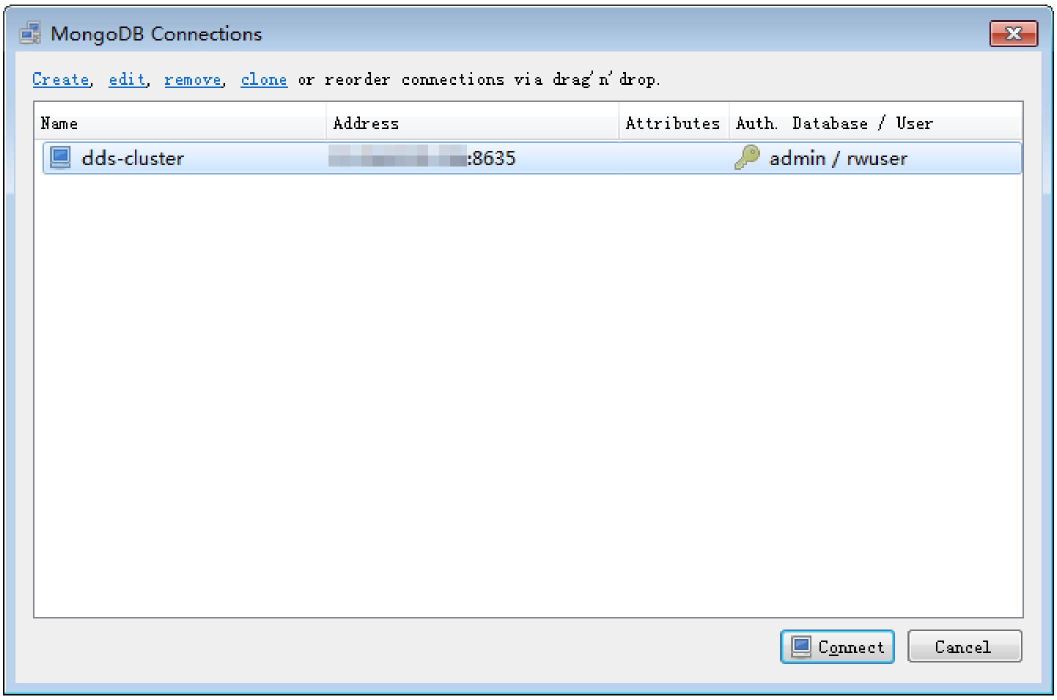
CRL (Revocation List): ...

Invalid Hostnames: Allowed

4. Click **Save**.

Step 3 On the **MongoDB Connections** page, click **Connect** to connect to the cluster instance.

Figure 3-7 Connections



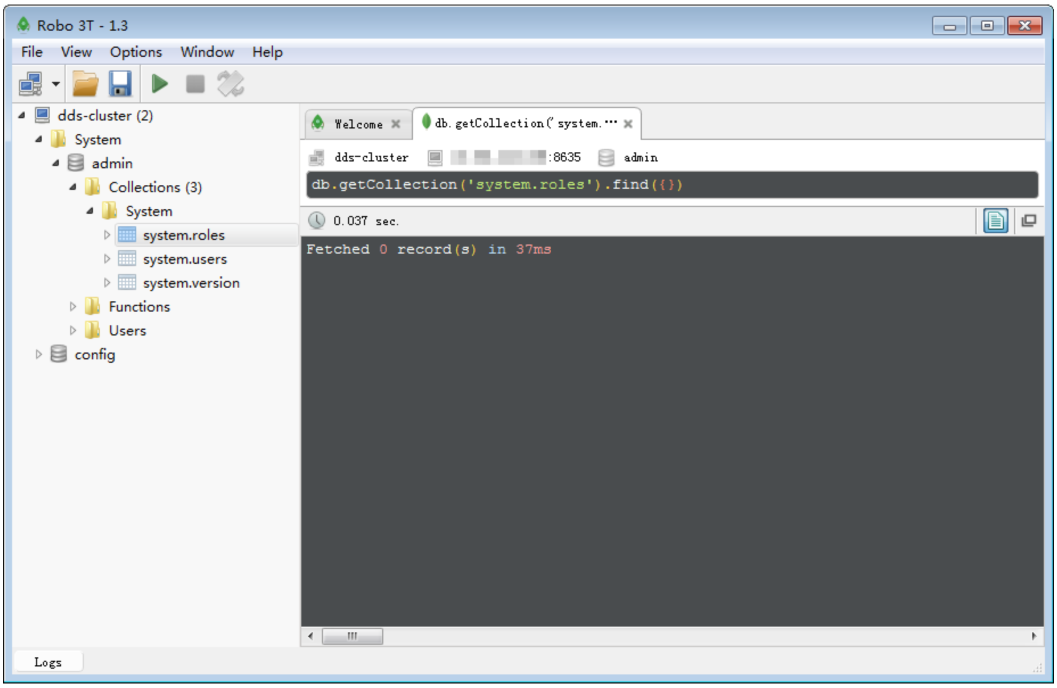
MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User
dds-cluster	[redacted]:8635		admin / rwuser

Step 4 If the cluster instance is successfully connected, the page shown in **Figure 3-8** is displayed.

Figure 3-8 Connection succeeded



```
db.getCollection('system.roles').find({})
```

```
Fetched 0 record(s) in 37ms
```

----End

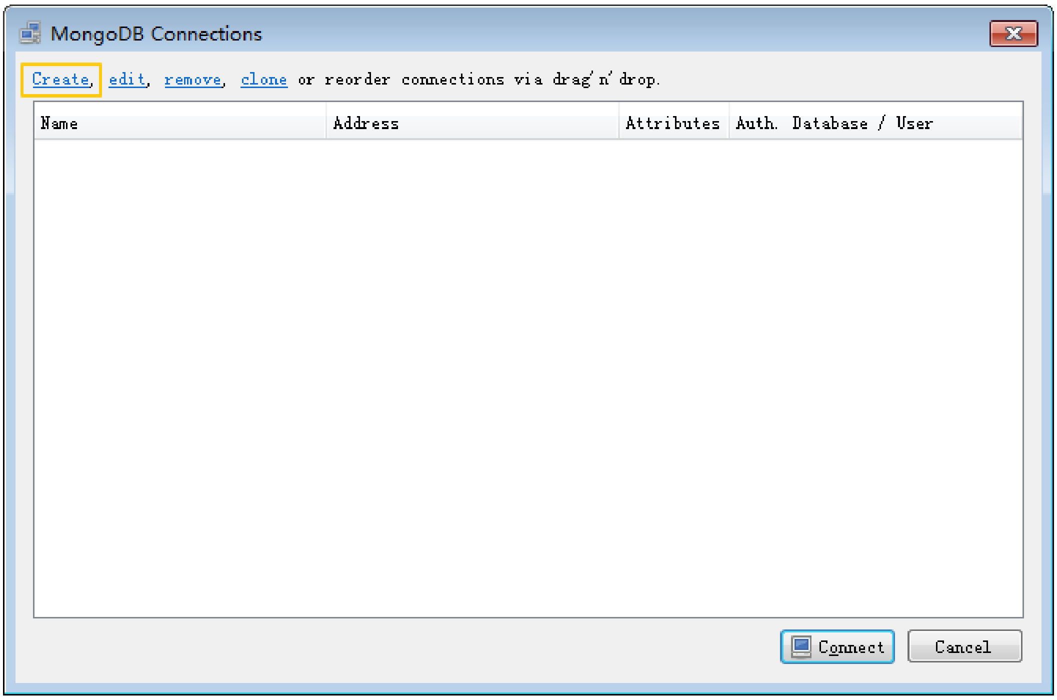
Connecting to a DB Instance Using Robo 3T (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Run the installed Robo 3T. On the displayed dialog box, click **Create**.

Figure 3-9 Connections

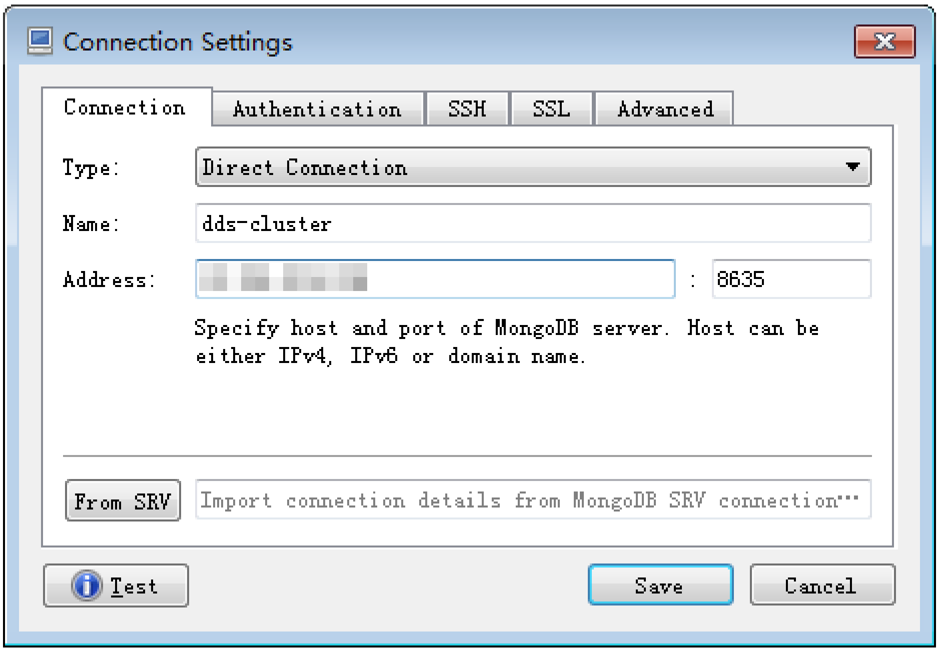


Create, edit, remove, clone or reorder connections via drag'n'drop.

Step 2 In the **Connection Settings** dialog box, set the parameters of the new connection.

1. On the **Connection** tab, enter the name of the new connection in the **Name** text box and enter the EIP and database port that are bound to the DDS DB instance in the **Address** text box.

Figure 3-10 Connection



Type: Direct Connection

Name: dds-cluster

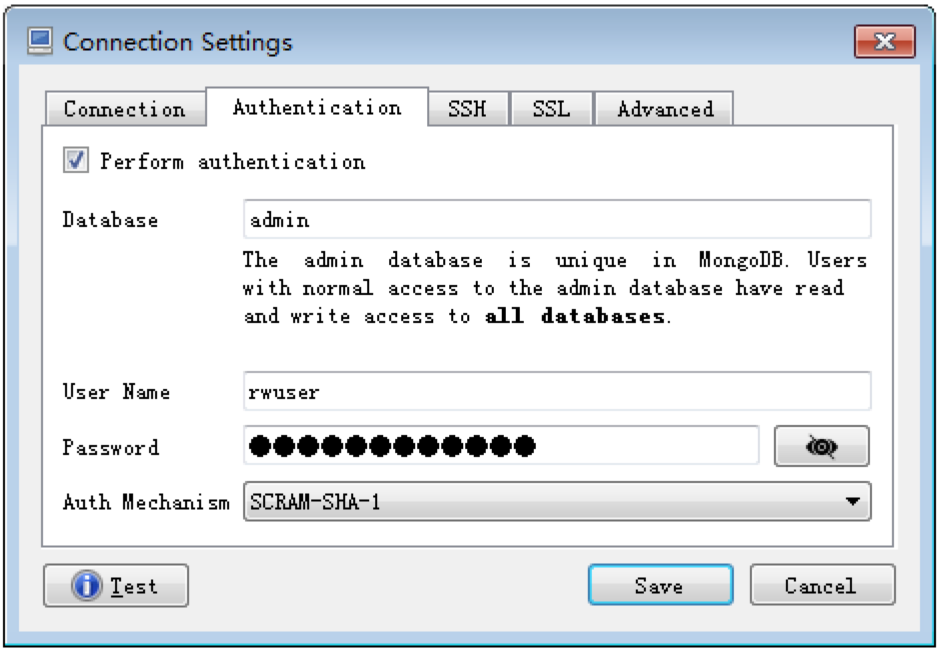
Address: [redacted] : 8635

Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

Import connection details from MongoDB SRV connection…

2. On the **Authentication** tab, set **Database** to **admin**, **User Name** to **rwuser**, and **Password** to the administrator password you set during the creation of the cluster instance.

Figure 3-11 Authentication



Connection Settings

☑ Perform authentication

Database: admin

The admin database is unique in MongoDB. Users with normal access to the admin database have read and write access to **all databases**.

User Name: rwuser

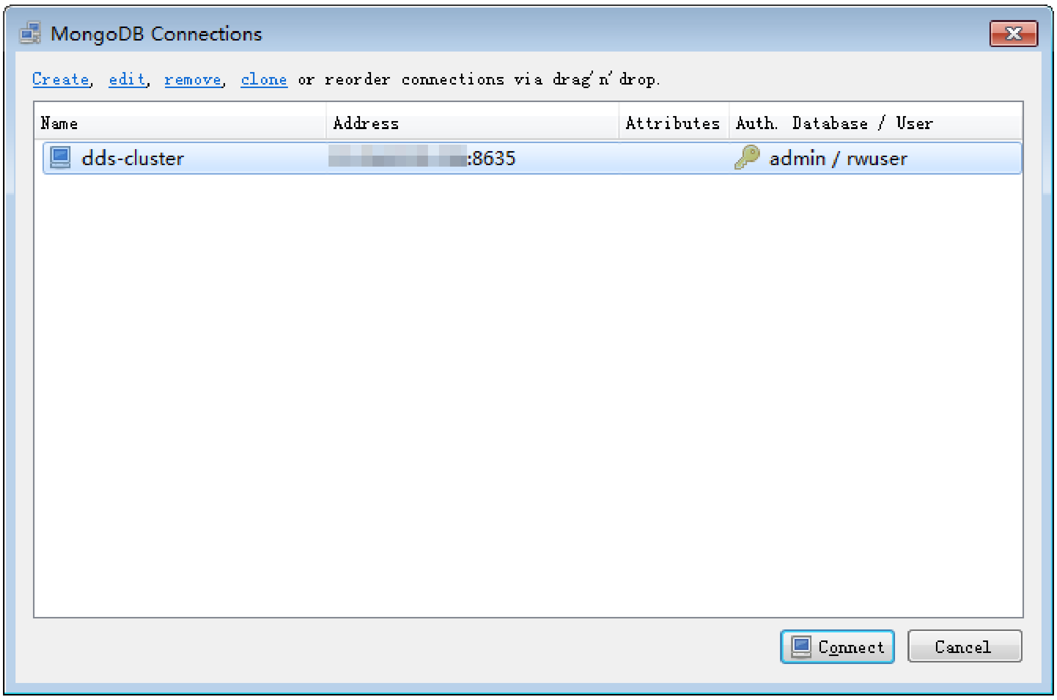
Password: ●●●●●●●●●●●●

Auth Mechanism: SCRAM-SHA-1

3. Click **Save**.

Step 3 On the **MongoDB Connections** page, click **Connect** to connect to the cluster instance.

Figure 3-12 Connections



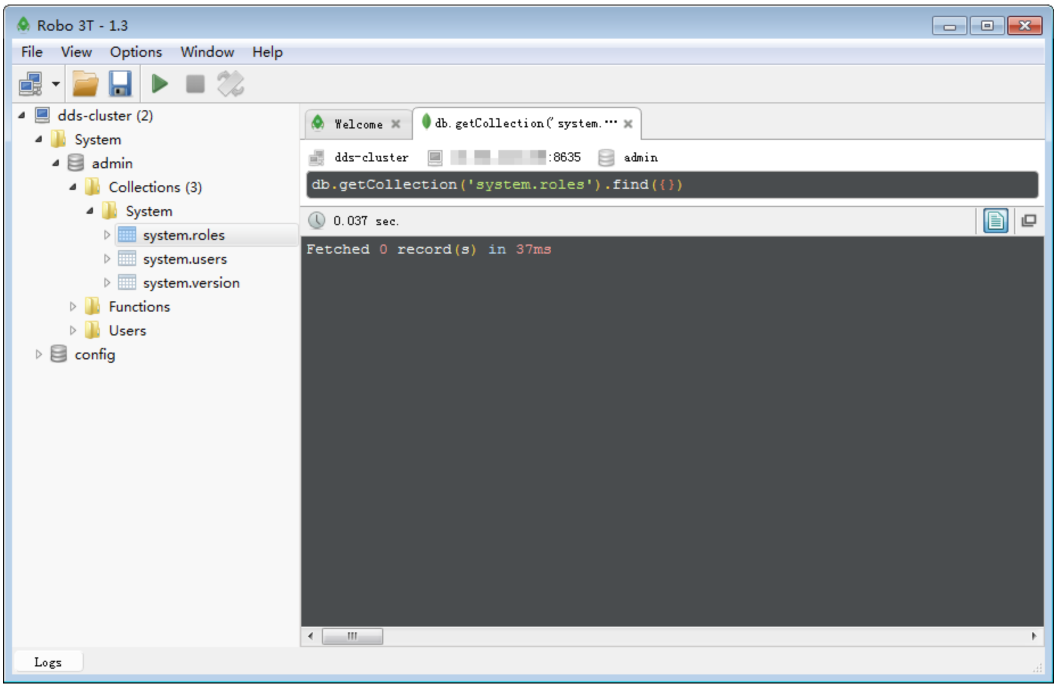
MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User
dds-cluster	[redacted]:8635		admin / rwuser

Step 4 If the cluster instance is successfully connected, the page shown in **Figure 3-13** is displayed.

Figure 3-13 Connection succeeded



```
db.getCollection('system.roles').find({})
```

0.037 sec.

```
Fetched 0 record(s) in 37ms
```

----End

Connecting to a DB Instance Using the MongoDB Client (SSL)

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, click ⤓ next to the **SSL** field.

Step 4 Upload the root certificate to the ECS to be connected to the DB instance.

The following describes how to upload the certificate to a Linux and Window ECS:

- In Linux, run the following command:

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

  NOTE

  – **IDENTITY_FILE** indicates the directory where the root certificate locates. 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 root certificate is uploaded.

- In Windows, upload the root certificate using the remote connection tool.

Step 5 Connect to the DB instance in the directory where the MongoDB client is located.

- Method 1: Using Linux commands

```
./mongo --host <DB_HOST> --port <DB_PORT> -u <DB_USER> -p --authenticationDatabase admin --ssl --sslCAFile <FILE_PATH> --sslAllowInvalidHostnames
```

  Enter the database account password when prompted:

```
Enter password:
```

- Method 2: Using the public connection address

```
./mongo mongodb://rwuser:****@<DB_HOST>:<DB_PORT>/test?authSource=admin --ssl --sslCAFile <FILE_PATH> --sslAllowInvalidHostnames
```

  To obtain the public connection address, click the instance name and choose **Connections**. The address is displayed in **Public Network Connection Address** field on the **Public Connection** tab.

NOTE

- A cluster instance uses the management IP address to generate SSL certificate. **--sslAllowInvalidHostnames** is needed for the SSL connection in a public network.
- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **EIP** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- **** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).
- **FILE_PATH** indicates the path where the root certificate is stored.

- Connect to the instance using Linux commands. The following is an example command:

```
./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames
```

- Connect to the DB instance using the public connection address. The following is an example command:

```
./mongo mongodb://rwuser:****@192.168.1.80:8635/test?authSource=admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames
```

Step 6 Check the connection result. If the following information is displayed, the connection is successful.

```
mongos>
```

----End

Connecting to a DB Instance Using the MongoDB Client (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Connect to the ECS.

Step 2 Connect to the DB instance in the directory where the MongoDB client is located.

- Method 1: Using Linux commands

  ```
  ./mongo --host <DB_HOST> --port <DB_PORT> -u <DB_USER> -p --authenticationDatabase admin
  ```

  Enter the database account password when prompted:

  ```
  Enter password:
  ```

- Method 2: Using the public connection address

  ```
  ./mongo mongodb://rwuser:****@<DB_HOST>:<DB_PORT>/test?authSource=admin
  ```

  To obtain the public connection address, click the instance name and choose **Connections**. The address is displayed in **Public Network Connection Address** field on the **Public Connection** tab.

NOTE

- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **EIP** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- **** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).

- Connect to the instance using Linux commands. The following is an example command:

  ```
  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin
  ```

- Connect to the DB instance using the public connection address. The following is an example command:

  ```
  ./mongo mongodb://rwuser:****@192.168.1.80:8635/test?authSource=admin
  ```

Step 3 Check the connection result. If the following information is displayed, the connection is successful.

```
mongos>
```

----End

4 Getting Started with Replica Sets

4.1 Connection Methods

You can access DDS over private or public networks.

Table 4-1 Connection methods

Method	IP Address	Scenario	Description
Private network	Private IP address	DDS provides a private IP address by default. ● If your applications are running on an ECS that is in the same region, AZ, and VPC subnet as your DDS DB instance, you are advised to use a private IP address to connect the ECS to your DDS DB instances. ● By default, DDS is not accessible from ECSs that are not in the same security group. If the ECS is not in the same group, you need to add an inbound rule to enable access. ● The default DDS port is 8635, but this port can be modified if necessary.	Secure and excellent performance

Method	IP Address	Scenario	Description
Public network	EIP	● If your applications are running on an ECS that is in a different region from the one where the DB instance is located, you are advised to use an EIP to connect the ECS to your DDS DB instances. ● If your applications are deployed on another cloud platform, EIP is recommended.	● Low security ● For faster transmission and improved security, you are advised to migrate your applications to an ECS that is in the same subnet as your DDS instance and use a private IP address to access the instance.

4.2 Connecting to a Replica Set Instance Over Private Networks

4.2.1 Overview

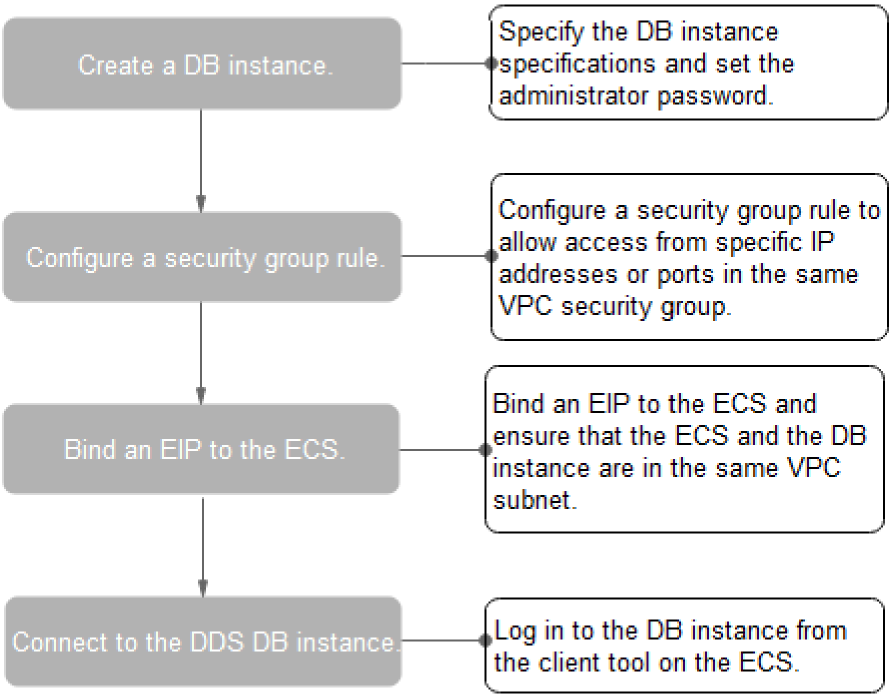
Scenarios

This section describes how to create a replica set instance on the management console, set a security group, and connect to a replica set instance over private networks.

Process

The following describes the steps from creating a DB instance to using it.

Figure 4-1 Accessing DB instances from a private network


Create a DB instance.
Specify the DB instance specifications and set the administrator password.
Configure a security group rule.
Configure a security group rule to allow access from specific IP addresses or ports in the same VPC security group.
Bind an EIP to the ECS.
Bind an EIP to the ECS and ensure that the ECS and the DB instance are in the same VPC subnet.
Connect to the DDS DB instance.
Log in to the DB instance from the client tool on the ECS.

4.2.2 Creating a Replica Set Instance

Scenarios

This section describes how to create a replica set instance on the DDS management console. DDS allows you to tailor your computing resources and storage space to your business needs.

You can use your account to create up to 50 replica set instances.

You can use your account to create up to 50 replica set instances. To create more replica set instances, click ◔ in the upper right corner of the management console. On the **Service Quota** page, click **Increase Quota** to apply for quotas.

Procedure

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

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

Step 3 On the displayed page, select your DB instance specifications and click **Create Now**.

Table 4-2 Basic information

Parameter	Description
Region	The region where the tenant is located. It can be changed in the upper left corner. For details, see section **Regions and AZs**. **NOTE** DB instances deployed in different regions cannot communicate with each other through a private network, and you cannot change the region of a DB instance once it is created. Exercise caution when selecting a region.
DB Instance Name	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 (_). After a DB instance is created, you can change the DB instance name. For details, see section **Changing a DB Instance Name**.
Database Type	Community Edition
DB Instance Type	Select **Replica set**. A replica set consists of the primary node, secondary node, and hidden node. If a primary node goes down or becomes faulty, a secondary node is automatically assigned to the primary role and continues normal operation. If a secondary node is unavailable, a hidden node will take the role of the secondary to ensure high availability.
Compatible MongoDB Version	● 4.0 ● 3.4
Storage Type	Ultra-high I/O
Storage Engine	WiredTiger
AZ	An AZ is a part of a region with its own independent power supplies and networks. AZs are physically isolated but can communicate through an internal network connection. Currently, instances can be deployed in a single AZ or three AZs. ● If you want to deploy an instance in a single AZ, select one AZ. ● If you want to deploy an instance across AZs for disaster recovery, select three AZs. In this deployment mode, the nodes are evenly distributed in the three AZs.

Parameter	Description
Disk Encryption	● **Disabled**: Disable the encryption function. ● **Enabled**: Enable the encryption function. This feature improves data security but slightly affects read/write performance. **Key Name**: Select or create a private key, which is the tenant key. **NOTE** – After a DB instance is created, the disk encryption status and the key cannot be changed. The backup data stored in OBS is not encrypted. – The key cannot be disabled or deleted when being used. Otherwise, the database becomes unavailable. – For details about how to create a key, see the "Creating a CMK" section in the *Key Management Service User Guide*.

Table 4-3 Specifications

Parameter	Description
Specifications	In the x86 CPU architecture, the following specifications can be selected to suit different application scenarios: General-purpose (s6), Enhanced (c3), and Enhanced II (c6).
Node Class	For details about the DB instance specifications, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Replica Set DB Instance**.
Storage Space	The value ranges from 10 GB to 2000 GB and must be a multiple of 10.

Table 4-4 Network

Parameter	Description
VPC	The VPC where your DB instances are located. A VPC isolates networks for different services, so you can easily manage and configure internal networks and change network configuration. You need to create or select the required VPC. For details about how to create a VPC, see section "Creating a VPC" in the *Virtual Private Cloud User Guide*. For details about the constraints on the use of VPCs, see **Connection Methods**.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. After the instance is created, you can change the private IP address assigned by the subnet. For details, see **Changing a Private IP Address**.
Security Group	A security group controls access between DDS and other services for security. **NOTE** Ensure that the security group rule you set allows clients to access DB instances. For example, select the TCP protocol with inbound direction, input the default port number **8635**, and enter a subnet IP address or select a security group that the DB instance belongs to.
SSL	Secure Sockets Layer (SSL) certificates set up encrypted connections between clients and servers, preventing data from being tampered with or stolen during transmission. You can enable SSL to improve data security. After a DB instance is created, you can connect to it using SSL.
Cross-CIDR Access	● Configure Add the VPC subnet of your client. Ensure that the ECS where your client is installed can connect to the DB instance. **NOTE** – To ensure the ECS and the DB instance can communicate with each other, configure the connection by referring to section "VPC Peering Connection Overview" in the *Virtual Private Cloud User Guide*. – VPC CIDR blocks can only be added, but not modified or deleted. – Up to 9 CIDR blocks can be configured, and each of them does not overlap. ● Skip Configure the CIDR block of the client later. After a DB instance is created, you can configure cross-subnet access by referring to **Configuring Cross-CIDR Access**.

Table 4-5 Database configuration

Parameter	Description
Administrator	The default account is **rwuser**.

Parameter	Description
Administrator Password	Set a password for the administrator. 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: ~!@#%^*-_=+? Keep this password secure. If lost, the system cannot retrieve it for you.
Confirm Password	Enter the administrator password again.
Replica Set Parameter Group	The parameters that apply to the replica set instances. After a DB instance is created, you can change the parameter group you configured for the DB instance to bring out the best performance. For details, see **Parameter Group Settings**.

NOTE

DB instance performance is determined by how you configure it during the creation. The hardware configuration items that can be selected include the class and storage space of the replica set.

Step 4 On the displayed page, confirm the DB instance information.

- If you need to modify the specifications, click **Previous** to return to the previous page.
- If you do not need to modify the specifications, click **Submit** to start the instance creation.

Step 5 After a DDS DB instance is created, you can view and manage it on the **Instance Management** page.

- When a DB instance is being created, the status displayed in the **Status** column is **Creating**. This process takes about 15 minutes. After the creation is complete, the status changes to **Available**.
- DDS enables the automated backup policy by default. After a DB instance is created, you can modify or disable the automated backup policy. An automated full backup is immediately triggered after the creation of a DB instance.
- The default DDS port is 8635, but this port can be modified if necessary. If you change the port, you need to add the security group rule to enable access.

----End

4.2.3 Setting a Security Group

Scenarios

This section guides you on how to add a security group rule to control access from and to DDS DB instances in a security group. The following describes how to set security groups.

Precautions

The default security group rule allows all outgoing data packets. ECSs and DDS DB instances in the same security group can access each other. After a security group is created, you can create different rules for that security group, which allows you to control access to the DB instances that are in it.

To access a DB instance in a security group from a source outside of that group, you need to create an inbound rule.

For details about the constraints on the using security groups, see "Security Group Overview" in the *Virtual Private Cloud User Guide*.

Procedure

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Security Group** area, on the **Inbound Rules** tab, click **Add Rule**. In the displayed **Add Inbound Rule** dialog box, set required parameters to add inbound rules. On the **Outbound Rules** tab, click **Add Rule**. In the displayed **Add Outbound Rule** dialog box, set required parameters to add outbound rules.

You can click ⊕ to add more rules.

Step 4 Add a security group rule as prompted.

Table 4-6 Parameter description

Parameter	Description	Value Example
Protocol	The network protocol required for access. You can allow all protocols or specify a specific protocol, TCP, UDP, ICMP, and SSH.	TCP
Port	Specifies the port that allows the access to ECSs or external devices.	8635

Parameter	Description	Value Example
Source/Destination	Specifies the supported IP address and security group that the rule applies to. ● **IP address**: The IP address or subnet that the rule applies to. Single IP addresses must be expressed using slash notation. – Single IP address: xxx.xxx.xxx.xxx/32 (IPv4) – Subnet: xxx.xxx.xxx.0/24 – All IP addresses: 0.0.0.0/0 ● **Security group**: A security group that access will be allowed from. ECSs in this security group will be granted access to DDS instance in the current security group.	● 192.168.10.0/24 ● default

Step 5 Click **OK**.

----End

4.2.4 Connecting to a Replica Set Instance Over Private Networks

Scenarios

This section describes how to connect to a replica set instance using the MongoDB client over private networks.

You can directly perform operations on the primary and secondary nodes. Primary nodes are used for processing read and write requests. Secondary nodes replicate data from the primary and are used for processing read requests only.

The MongoDB client can connect to a DB instance with a common connection or an encrypted connection (SSL). To improve data transmission security, you are advised to connect to DB instances using the SSL connection.

Different OS scenarios: The following uses Linux ECS and Window client as an example.

Constraints

For details about constraints on connecting to a replica set instance over private networks, see **Constraints**.

Prerequisites

1. For details on how to create and log in to an ECS, see "Creating and Logging In to a Windows ECS" or "Creating and Logging In to a Linux ECS" in the *Elastic Cloud Server User Guide*.
2. Install the MongoDB client on the ECS.

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

Connecting to a DB Instance Using the MongoDB Client (SSL)

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, click ⭳ next to the **SSL** field.

Step 4 Upload the root certificate to the ECS to be connected to the DB instance.

The following describes how to upload the certificate to a Linux and Window ECS:

- In Linux, run the following command:

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

  NOTE

  - **IDENTITY_FILE** indicates the directory where the root certificate locates. 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 root certificate is uploaded.

- In Windows, upload the root certificate using the remote connection tool.

Step 5 Connect to a DDS DB instance.

- Method 1: Using Linux commands

  ./mongo --host *<DB_HOST>* **--port** *<DB_PORT>* **-u** *<DB_USER>* **-p --authenticationDatabase admin --ssl --sslCAFile** *<FILE_PATH>* **--sslAllowInvalidHostnames**

  Enter the database account password when prompted:

  ```
  Enter password:
  ```

- Method 2: Using the private connection address

  ./mongo "mongodb://rwuser:`****`**@***<DB_HOST1>***:***<DB_PORT1>***,***<DB_HOST2>***:***<DB_PORT2>***/test?authSource=admin&replicaSet=replica" --ssl --sslCAFile** *<FILE_PATH>* **--sslAllowInvalidHostnames**

  If the DB instance is connected using the connection address, add double quotation marks before and after the connection information. The connection information can be obtained in the **Address** column on the **Instance Management** page.

NOTE

- A replica set instance uses the management IP address to generate SSL certificate. **--sslAllowInvalidHostnames** is needed for the SSL connection through a private network.
- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **Private IP Address** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- ****** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).
- **FILE_PATH** indicates the path where the root certificate is stored.

- Connect to the instance using Linux commands. The following is an example command:

  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames

- Connect to the DB instance using the private connection address. The following is an example command:

  ./mongo "mongodb://rwuser:**@192.168.1.6:8635,192.168.1.80:8635/test?authSource=admin&replicaSet=replica" --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames**

Step 6 Check the connection result. If the following information is displayed, the connection is successful.

- Result from connecting the primary node in a replica set:

```
replica:PRIMARY>
```

- Result from connecting the secondary node in a replica set:

```
replica:SECONDARY>
```

----End

Connecting to a DB Instance Using the MongoDB Client (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Connect to the ECS.

Step 2 Connect to a DDS DB instance.

- Method 1: Using Linux commands

  ./mongo --host *<DB_HOST>* **--port** *<DB_PORT>* **-u** *<DB_USER>* **-p --authenticationDatabase admin**

  Enter the database account password when prompted:

```
Enter password:
```

- Method 2: Using the private connection address

  ./mongo "mongodb://rwuser:**@<DB_HOST1>:<DB_PORT1>,<DB_HOST2>:<DB_PORT2>/test?authSource=admin&replicaSet=replica"**

  If the DB instance is connected using the connection address, add double quotation marks before and after the connection information. The connection information can be obtained in the **Address** column on the **Instance Management** page.

NOTE

- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **Private IP Address** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- **** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).

- Connect to the instance using Linux commands. The following is an example command:

  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin

- Connect to the DB instance using the private connection address. The following is an example command:

  ./mongo "mongodb://rwuser:**@192.168.1.6:8635,192.168.1.80:8635/test?authSource=admin&replicaSet=replica"**

Step 3 Check the connection result. If the following information is displayed, the connection is successful.

- Result from connecting the primary node in a replica set:

```
replica:PRIMARY>
```

- Result from connecting the secondary node in a replica set:

```
replica:SECONDARY>
```

----End

4.3 Connecting to a Replica Set Instance Over Public Networks

4.3.1 Overview

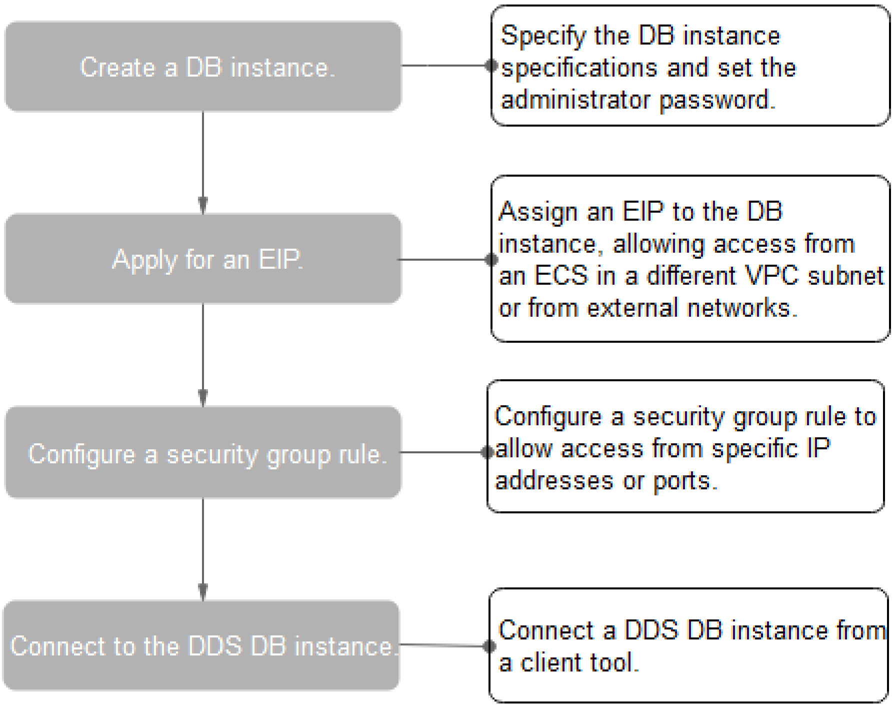
Scenarios

This section describes how to create a replica set instance on the management console, set a security group, bind an EIP, and connect to a replica set instance over public networks.

Process

The following describes the steps from creating a DB instance to using it.

Figure 4-2 Accessing DB instances from a public network


Create a DB instance.
Specify the DB instance specifications and set the administrator password.
Apply for an EIP.
Assign an EIP to the DB instance, allowing access from an ECS in a different VPC subnet or from external networks.
Configure a security group rule.
Configure a security group rule to allow access from specific IP addresses or ports.
Connect to the DDS DB instance.
Connect a DDS DB instance from a client tool.

4.3.2 Creating a Replica Set Instance

Scenarios

This section describes how to create a replica set instance on the DDS management console. DDS allows you to tailor your computing resources and storage space to your business needs.

You can use your account to create up to 50 replica set instances.

You can use your account to create up to 50 replica set instances. To create more replica set instances, click ◔ in the upper right corner of the management console. On the **Service Quota** page, click **Increase Quota** to apply for quotas.

Procedure

Step 1 Log in to the DDS console.

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

Step 3 On the displayed page, select your DB instance specifications and click **Create Now**.

Table 4-7 Basic information

Parameter	Description
Region	The region where the tenant is located. It can be changed in the upper left corner. For details, see section **Regions and AZs**. **NOTE** DB instances deployed in different regions cannot communicate with each other through a private network, and you cannot change the region of a DB instance once it is created. Exercise caution when selecting a region.
DB Instance Name	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 (_). After a DB instance is created, you can change the DB instance name. For details, see section **Changing a DB Instance Name**.
Database Type	Community Edition
DB Instance Type	Select **Replica set**. A replica set consists of the primary node, secondary node, and hidden node. If a primary node goes down or becomes faulty, a secondary node is automatically assigned to the primary role and continues normal operation. If a secondary node is unavailable, a hidden node will take the role of the secondary to ensure high availability.
Compatible MongoDB Version	● 4.0 ● 3.4
Storage Type	Ultra-high I/O
Storage Engine	WiredTiger
AZ	An AZ is a part of a region with its own independent power supplies and networks. AZs are physically isolated but can communicate through an internal network connection. Currently, instances can be deployed in a single AZ or three AZs. ● If you want to deploy an instance in a single AZ, select one AZ. ● If you want to deploy an instance across AZs for disaster recovery, select three AZs. In this deployment mode, the nodes are evenly distributed in the three AZs.

Parameter	Description
Disk Encryption	- **Disabled**: Disable the encryption function. - **Enabled**: Enable the encryption function. This feature improves data security but slightly affects read/write performance. **Key Name**: Select or create a private key, which is the tenant key. **NOTE** – After a DB instance is created, the disk encryption status and the key cannot be changed. The backup data stored in OBS is not encrypted. – The key cannot be disabled or deleted when being used. Otherwise, the database becomes unavailable. – For details about how to create a key, see the "Creating a CMK" section in the *Key Management Service User Guide*.

Table 4-8 Specifications

Parameter	Description
Specifications	In the x86 CPU architecture, the following specifications can be selected to suit different application scenarios: General-purpose (s6), Enhanced (c3), and Enhanced II (c6).
Node Class	For details about the DB instance specifications, see section **DB Instance Specifications**. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing the CPU or Memory of a Replica Set DB Instance**.
Storage Space	The value ranges from 10 GB to 2000 GB and must be a multiple of 10.

Table 4-9 Network

Parameter	Description
VPC	The VPC where your DB instances are located. A VPC isolates networks for different services, so you can easily manage and configure internal networks and change network configuration. You need to create or select the required VPC. For details about how to create a VPC, see section "Creating a VPC" in the *Virtual Private Cloud User Guide*. For details about the constraints on the use of VPCs, see **Connection Methods**.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. After the instance is created, you can change the private IP address assigned by the subnet. For details, see **Changing a Private IP Address**.
Security Group	A security group controls access between DDS and other services for security. **NOTE** Ensure that the security group rule you set allows clients to access DB instances. For example, select the TCP protocol with inbound direction, input the default port number **8635**, and enter a subnet IP address or select a security group that the DB instance belongs to.
SSL	Secure Sockets Layer (SSL) certificates set up encrypted connections between clients and servers, preventing data from being tampered with or stolen during transmission. You can enable SSL to improve data security. After a DB instance is created, you can connect to it using SSL.
Cross-CIDR Access	● Configure Add the VPC subnet of your client. Ensure that the ECS where your client is installed can connect to the DB instance. **NOTE** – To ensure the ECS and the DB instance can communicate with each other, configure the connection by referring to section "VPC Peering Connection Overview" in the *Virtual Private Cloud User Guide*. – VPC CIDR blocks can only be added, but not modified or deleted. – Up to 9 CIDR blocks can be configured, and each of them does not overlap. ● Skip Configure the CIDR block of the client later. After a DB instance is created, you can configure cross-subnet access by referring to **Configuring Cross-CIDR Access**.

Table 4-10 Database configuration

Parameter	Description
Administrator	The default account is **rwuser**.

Parameter	Description
Administrator Password	Set a password for the administrator. 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: ~!@#%^*-_=+? Keep this password secure. If lost, the system cannot retrieve it for you.
Confirm Password	Enter the administrator password again.
Replica Set Parameter Group	The parameters that apply to the replica set instances. After a DB instance is created, you can change the parameter group you configured for the DB instance to bring out the best performance. For details, see **Parameter Group Settings**.

NOTE

DB instance performance is determined by how you configure it during the creation. The hardware configuration items that can be selected include the class and storage space of the replica set.

Step 4 On the displayed page, confirm the DB instance information.

- If you need to modify the specifications, click **Previous** to return to the previous page.
- If you do not need to modify the specifications, click **Submit** to start the instance creation.

Step 5 After a DDS DB instance is created, you can view and manage it on the **Instance Management** page.

- When a DB instance is being created, the status displayed in the **Status** column is **Creating**. This process takes about 15 minutes. After the creation is complete, the status changes to **Available**.
- DDS enables the automated backup policy by default. After a DB instance is created, you can modify or disable the automated backup policy. An automated full backup is immediately triggered after the creation of a DB instance.
- The default DDS port is 8635, but this port can be modified if necessary. If you change the port, you need to add the security group rule to enable access.

----End

4.3.3 Binding an EIP

Scenarios

After you create a DB instance, you can bind it to an EIP to allow external access. If later you want to prohibit external access, you can also unbind the EIP from the DB instance.

Precautions

- Before accessing a database, you need to apply for an EIP on the VPC console. Then, add an inbound rule to allow the IP addresses or IP address ranges of ECSs. For details, see section **Setting a Security Group**.
- In the replica set instance, only primary and secondary nodes can be bound to an EIP. To change the EIP that has been bound to a node, you need to unbind it from the node first.

Binding an EIP

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, locate the target node and click **Bind EIP** in the **Operation** column.

Step 4 In the displayed dialog box, all available unbound EIPs are listed. Select the required EIP and click **OK**. If no available EIPs are displayed, click **View EIP** and create an EIP on the VPC console.

Step 5 Locate the target node, in the **EIP** column, view the EIP that is successfully bound.

To unbind an EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

Step 1 On the **Instance Management** page, click the replica set instance that has been bound with an EIP.

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, locate the target node and click **Unbind EIP** in the **Operation** column.

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

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

4.3.4 Setting a Security Group

Scenarios

This section guides you on how to add a security group rule to control access from and to DDS DB instances in a security group. The following describes how to set security groups.

Precautions

The default security group rule allows all outgoing data packets. ECSs and DDS DB instances in the same security group can access each other. After a security group is created, you can create different rules for that security group, which allows you to control access to the DB instances that are in it.

To access a DB instance in a security group from a source outside of that group, you need to create an inbound rule.

For details about the constraints on the using security groups, see "Security Group Overview" in the *Virtual Private Cloud User Guide*.

Procedure

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Security Group** area, on the **Inbound Rules** tab, click **Add Rule**. In the displayed **Add Inbound Rule** dialog box, set required parameters to add inbound rules. On the **Outbound Rules** tab, click **Add Rule**. In the displayed **Add Outbound Rule** dialog box, set required parameters to add outbound rules.

You can click ⊕ to add more rules.

Step 4 Add a security group rule as prompted.

Table 4-11 Parameter description

Parameter	Description	Value Example
Protocol	The network protocol required for access. You can allow all protocols or specify a specific protocol, TCP, UDP, ICMP, and SSH.	TCP
Port	Specifies the port that allows the access to ECSs or external devices.	8635

Parameter	Description	Value Example
Source/Destination	Specifies the supported IP address and security group that the rule applies to. ● **IP address**: The IP address or subnet that the rule applies to. Single IP addresses must be expressed using slash notation. – Single IP address: xxx.xxx.xxx.xxx/32 (IPv4) – Subnet: xxx.xxx.xxx.0/24 – All IP addresses: 0.0.0.0/0 ● **Security group**: A security group that access will be allowed from. ECSs in this security group will be granted access to DDS instance in the current security group.	● 192.168.10.0/24 ● default

Step 5 Click **OK**.

----End

4.3.5 Connecting to a Replica Set Instance Over Public Networks

Scenarios

This section describes how to connect to a replica set instance using the MongoDB client and Robo 3T over public networks.

You can directly perform operations on the primary and secondary nodes. Primary nodes are used for processing read and write requests. Secondary nodes replicate data from the primary and are used for processing read requests only.

The MongoDB client and Robo 3T can connect to a DB instance with a common connection or an encrypted connection (SSL). To improve data transmission security, you are advised to connect to DB instances using the SSL connection.

Different OS scenarios: The following uses Linux ECS and Window client as an example.

Prerequisites

1. **Bind an EIP** to the cluster instance and **set security group rules** to ensure that the EIP can be accessed through the ECS or Robo 3T.
2. Install the MongoDB client or Robo 3T.

   MongoDB client

   a. For details on how to create and log in to an ECS, see "Creating and Logging In to a Windows ECS" or "Creating and Logging In to a Linux ECS" in the *Elastic Cloud Server User Guide*.
   b. Install the MongoDB client on the ECS.

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

Robo 3T

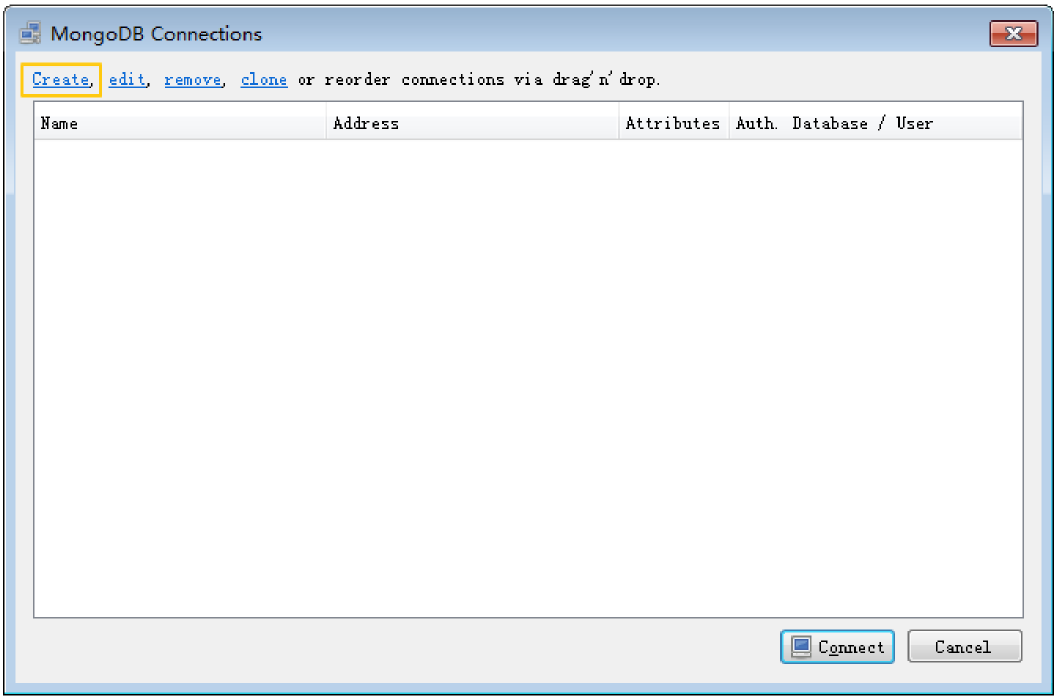
For details on how to install Robo 3T, see **How Do I Install Robo 3T?**

3. If you select SSL mode, download the SSL certificate on the DDS console.
   a. On the **Instance Management** page, click the target DB instance.
   b. In the navigation pane on the left, choose **Connections**.
   c. In the **Basic Information** area, click ⭳ next to the **SSL** field.

Connecting to a DB Instance Using Robo 3T (SSL)

Step 1 Run the installed Robo 3T. On the displayed dialog box, click **Create**.

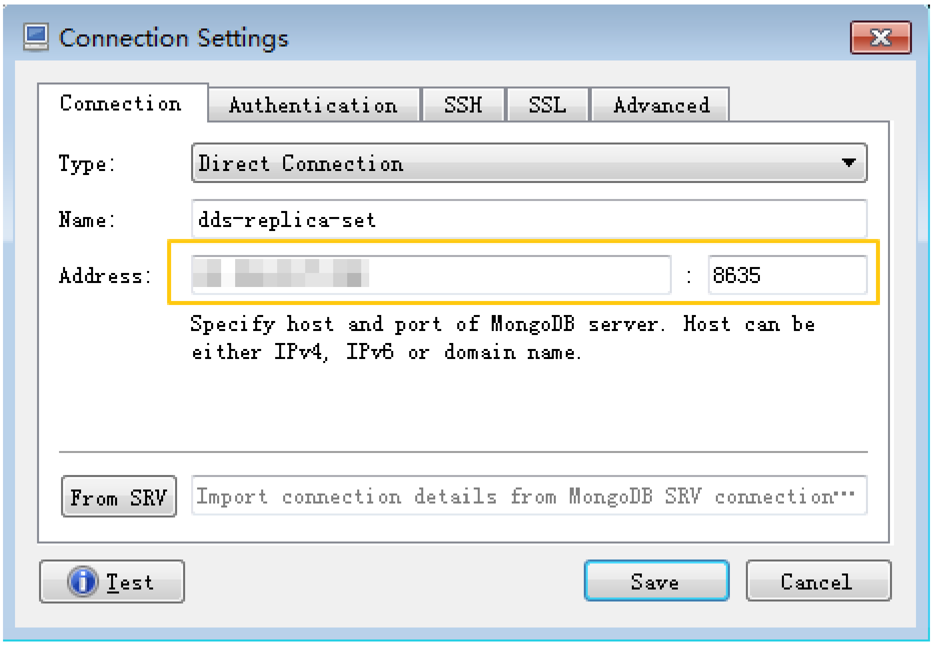
Figure 4-3 Connections



Step 2 In the **Connection Settings** dialog box, set the parameters of the new connection.

1. On the **Connection** tab, enter the name of the new connection in the **Name** text box and enter the EIP and database port that are bound to the replica set instance in the **Address** text box.

Figure 4-4 Connection



Connection Settings

Type: Direct Connection

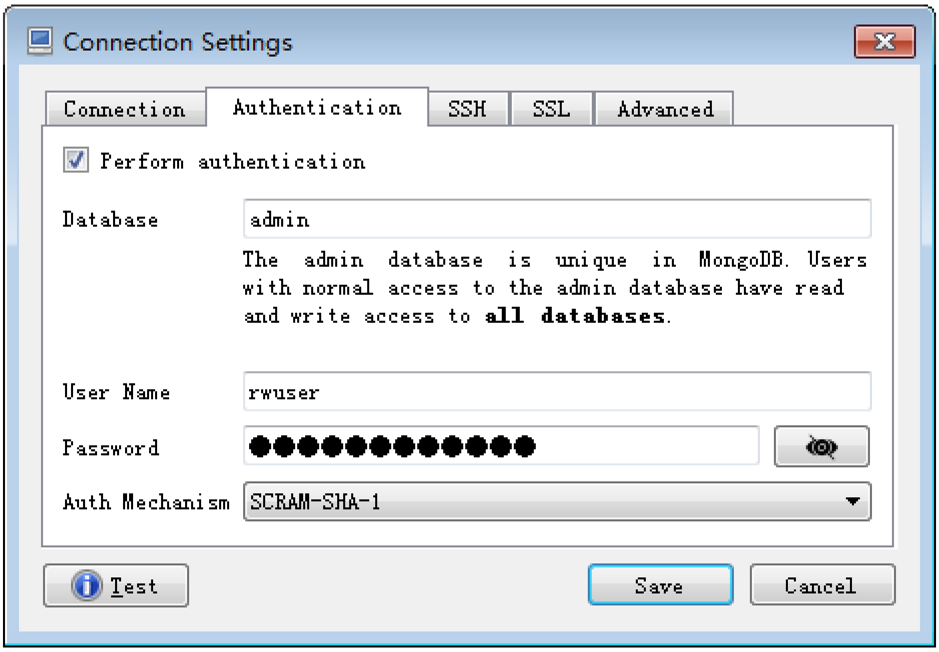
Name: dds-replica-set

Address: [redacted] : 8635

Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

2. On the **Authentication** tab, set **Database** to **admin**, **User Name** to **rwuser**, and **Password** to the administrator password you set during the creation of the replica set instance.

Figure 4-5 Authentication



Connection Settings

☑ Perform authentication

Database: admin

The admin database is unique in MongoDB. Users with normal access to the admin database have read and write access to **all databases**.

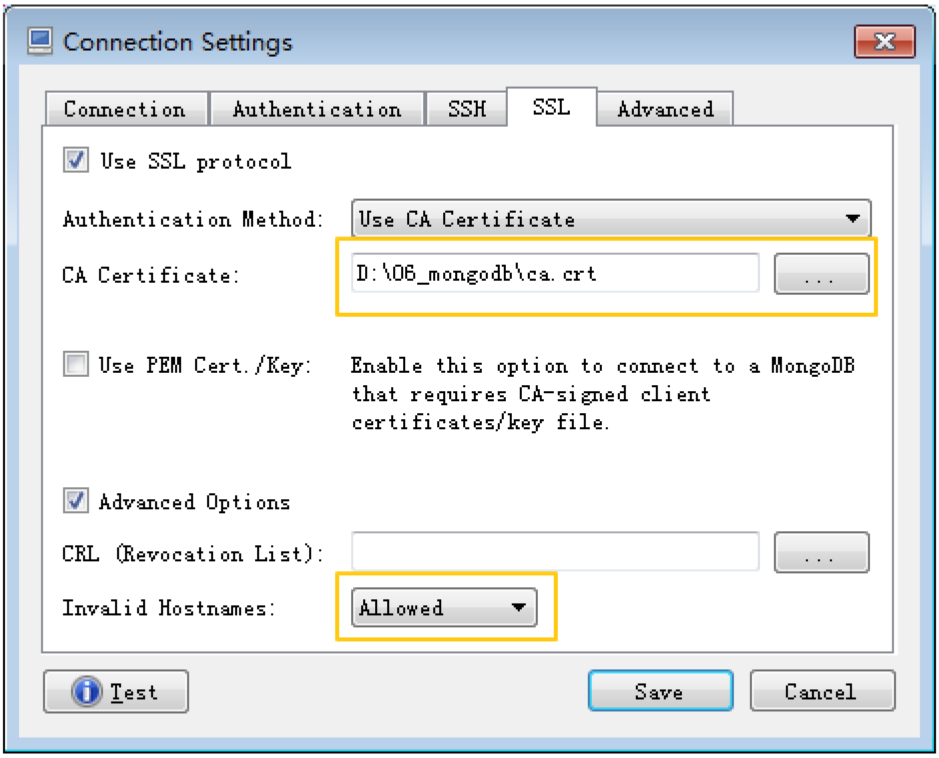
User Name: rwuser

Password: ●●●●●●●●●●●●●

Auth Mechanism: SCRAM-SHA-1

3. On the **SSL** tab, upload the SSL certificate and select **Allowed** for **Invalid Hostnames**.

Figure 4-6 SSL



Connection Settings

☑ Use SSL protocol

Authentication Method: Use CA Certificate

CA Certificate: D:\06_mongodb\ca.crt ...

□ Use PEM Cert./Key: Enable this option to connect to a MongoDB that requires CA-signed client certificates/key file.

☑ Advanced Options

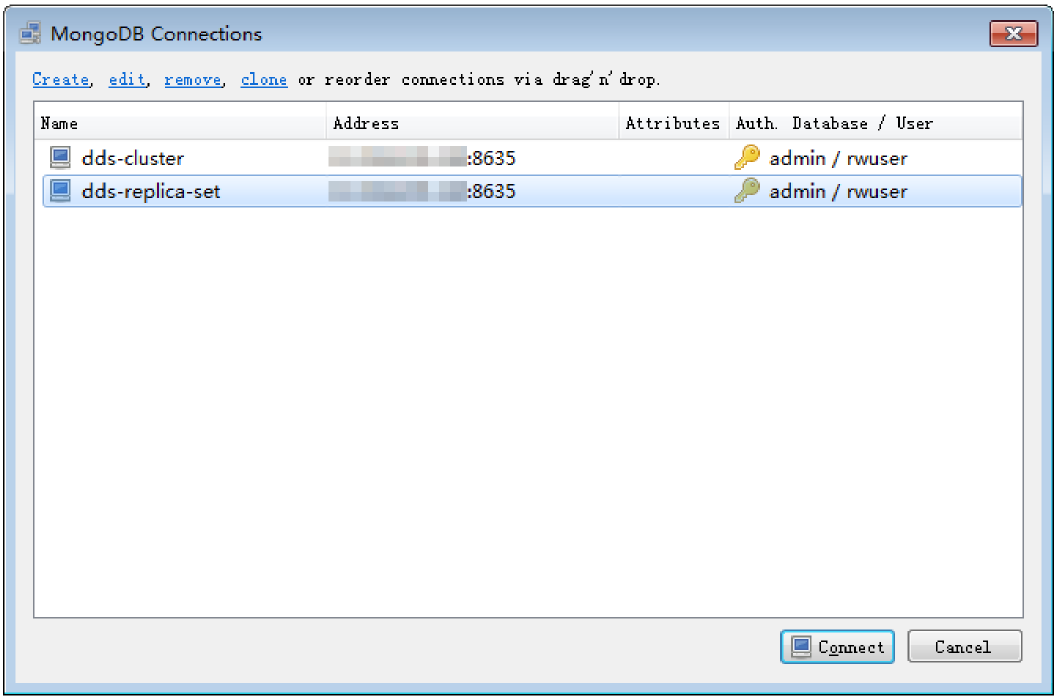
CRL (Revocation List): ...

Invalid Hostnames: Allowed

4. Click **Save**.

Step 3 On the **MongoDB Connections** page, click **Connect** to connect to the replica set instance.

Figure 4-7 Connections



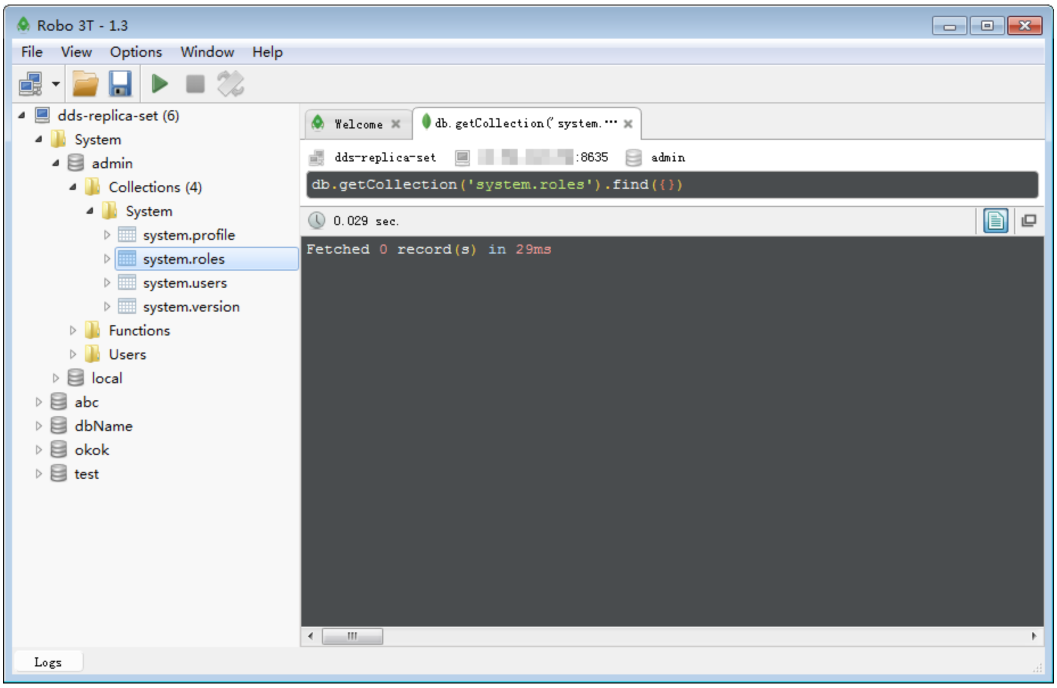
MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User
dds-cluster	[redacted]:8635		admin / rwuser
dds-replica-set	[redacted]:8635		admin / rwuser

Step 4 If the replica set instance is successfully connected, the page shown in **Figure 4-8** is displayed.

Figure 4-8 Connection succeeded



```
db.getCollection('system.roles').find({})
```

```
Fetched 0 record(s) in 29ms
```

----End

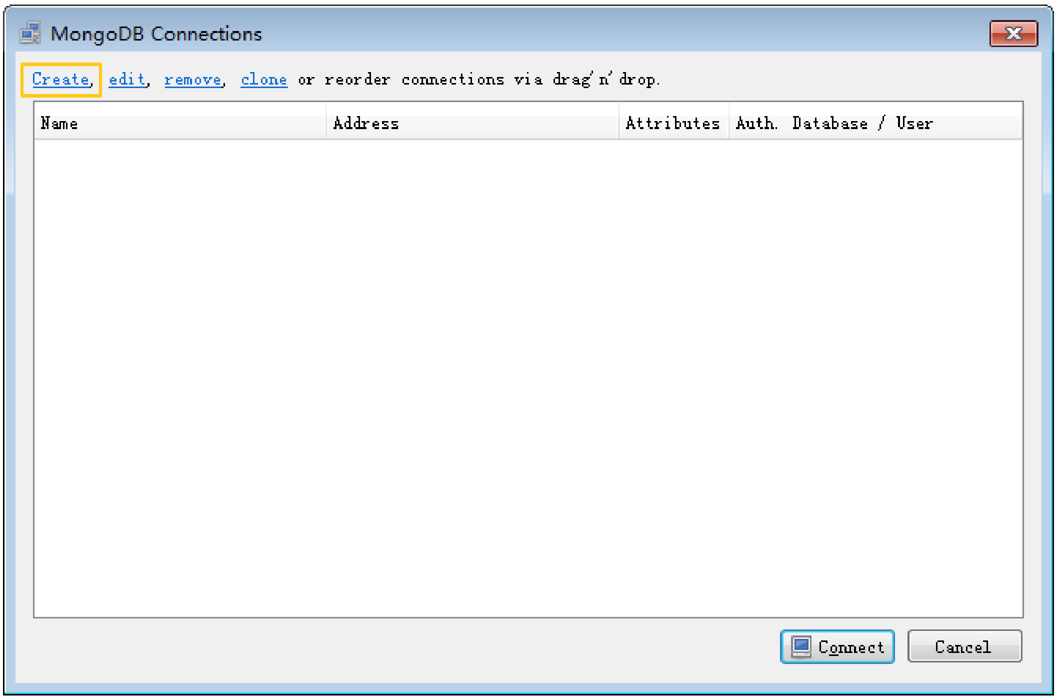
Connecting to a DB Instance Using Robo 3T (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Run the installed Robo 3T. On the displayed dialog box, click **Create**.

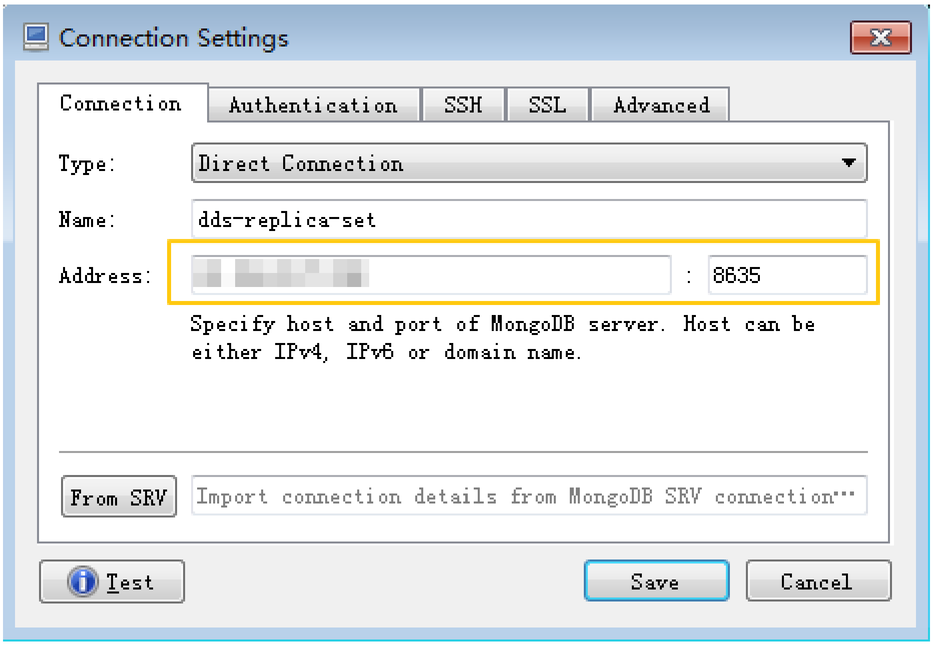
Figure 4-9 Connections



Step 2 In the **Connection Settings** dialog box, set the parameters of the new connection.

1. On the **Connection** tab, enter the name of the new connection in the **Name** text box and enter the EIP and database port that are bound to the replica set instance in the **Address** text box.

Figure 4-10 Connection



Type: Direct Connection

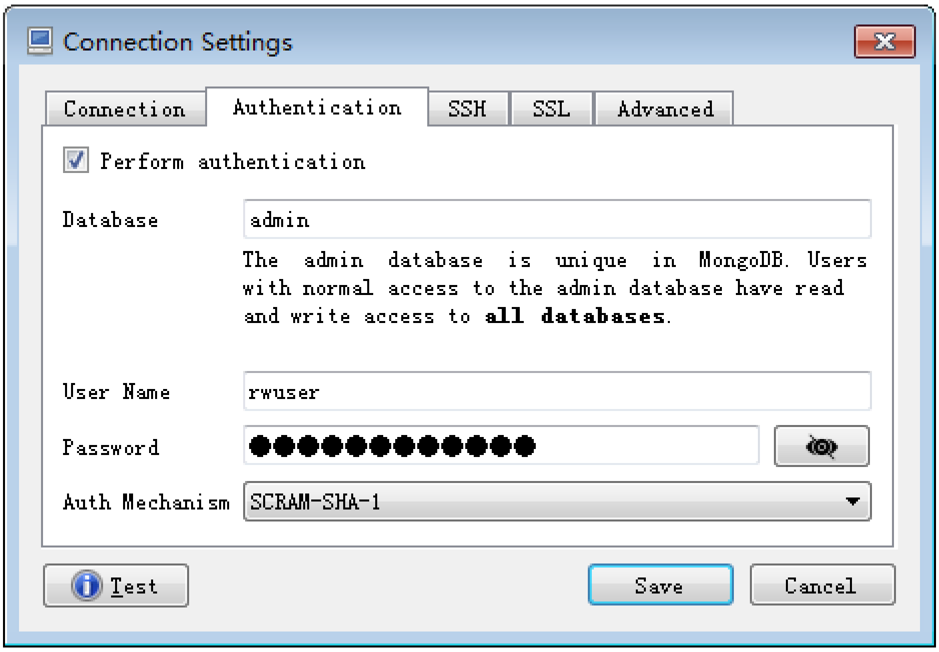
Name: dds-replica-set

Address: [redacted] : 8635

Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

2. On the **Authentication** tab, set **Database** to **admin**, **User Name** to **rwuser**, and **Password** to the administrator password you set during the creation of the replica set instance.

Figure 4-11 Authentication



Connection Settings

Connection | Authentication | SSH | SSL | Advanced

☑ Perform authentication

Database: admin

The admin database is unique in MongoDB. Users with normal access to the admin database have read and write access to **all databases**.

User Name: rwuser

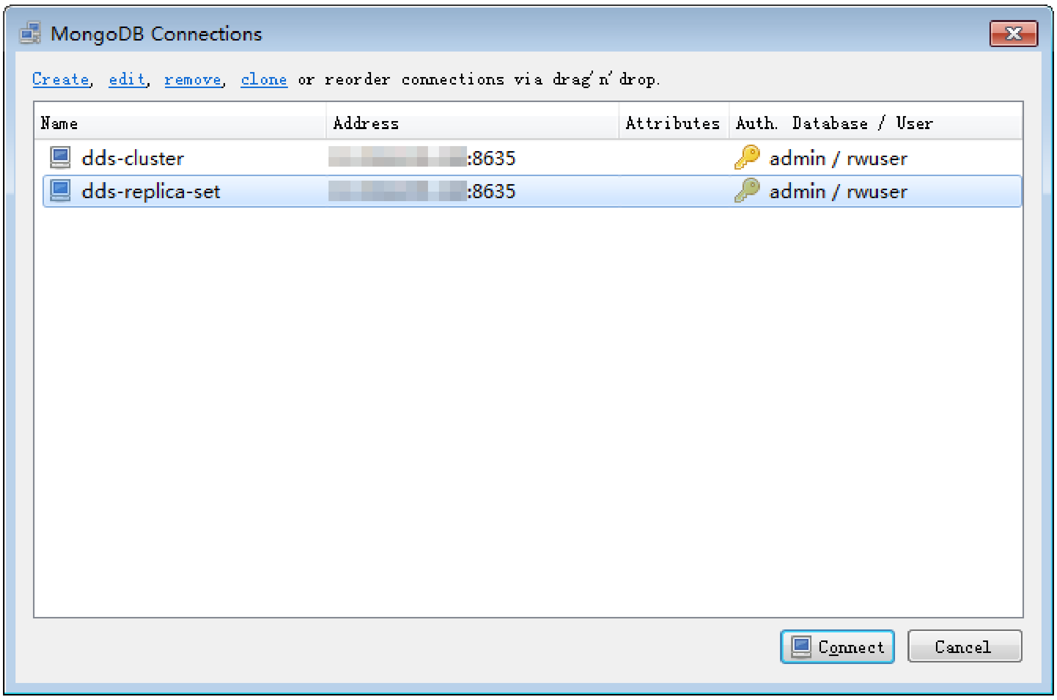
Password: ●●●●●●●●●●●●

Auth Mechanism: SCRAM-SHA-1

3. Click **Save**.

Step 3 On the **MongoDB Connections** page, click **Connect** to connect to the replica set instance.

Figure 4-12 Connections

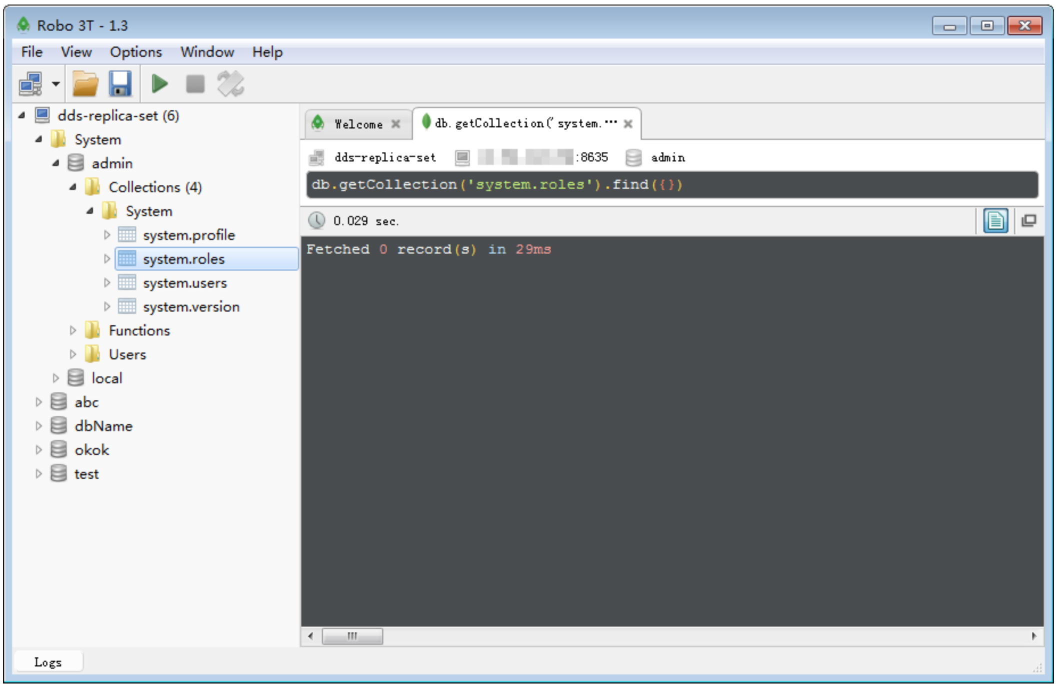


MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User
dds-cluster	[redacted]:8635		admin / rwuser
dds-replica-set	[redacted]:8635		admin / rwuser

Step 4 If the replica set instance is successfully connected, the page shown in **Figure 4-13** is displayed.



```
db.getCollection('system.roles').find({})
```

0.029 sec.

```
Fetched 0 record(s) in 29ms
```

Figure 4-13 Connection succeeded

----End

Connecting to a DB Instance Using the MongoDB Client (SSL)

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

Step 2 In the navigation pane on the left, choose **Connections**.

Step 3 In the **Basic Information** area, click 📥 next to the **SSL** field.

Step 4 Upload the root certificate to the ECS to be connected to the DB instance.

The following describes how to upload the certificate to a Linux and Window ECS:

- In Linux, run the following command:

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

  NOTE

  - **IDENTITY_FILE** indicates the directory where the root certificate locates. 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 root certificate is uploaded.

- In Windows, upload the root certificate using the remote connection tool.

Step 5 Connect to the DB instance in the directory where the MongoDB client is located.

- Method 1: Using Linux commands

./mongo --host <*DB_HOST*> **--port** <*DB_PORT*> **-u** <*DB_USER*> **-p --authenticationDatabase admin --ssl --sslCAFile** <*FILE_PATH*> **--sslAllowInvalidHostnames**

Enter the database account password when prompted:

```
Enter password:
```

- Method 2: Using the public connection address

  ./mongo "mongodb://rwuser:***@<*DB_HOST*>:<*DB_PORT*>**/test?authSource=admin&replicaSet=replica" --ssl --sslCAFile** <*FILE_PATH*> **--sslAllowInvalidHostnames**

  To obtain the public connection address, click the instance name and choose **Connections**. The address is displayed in **Public Network Connection Address** field on the **Public Connection** tab.

NOTE

- A replica set instance uses the management IP address to generate SSL certificate. **--sslAllowInvalidHostnames** is needed for the SSL connection through a public network.
- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **EIP** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- ********** indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).
- **FILE_PATH** indicates the path where the root certificate is stored.

- Connect to the instance using Linux commands. The following is an example command:

  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames

- Connect to the DB instance using the public connection address. The following is an example command:

  ./mongo "mongodb://rwuser:**@192.168.1.80:8635/test?authSource=admin&replicaSet=replica" --ssl --sslCAFile /tmp/ca.crt --sslAllowInvalidHostnames**

Step 6 Check the connection result. If the following information is displayed, the connection is successful.

- Result from connecting the primary node in a replica set:

```
replica:PRIMARY>
```

- Result from connecting the secondary node in a replica set:

```
replica:SECONDARY>
```

----End

Connecting to a DB Instance Using the MongoDB Client (Non-SSL)

NOTICE

If you connect to a DB instance using this method, you need to disable the SSL connection. For details, see section **Disabling SSL**.

Step 1 Connect to the ECS.

Step 2 Connect to a DDS DB instance.

- Method 1: Using Linux commands

  ```
  ./mongo --host <DB_HOST> --port <DB_PORT> -u <DB_USER> -p --authenticationDatabase admin
  ```

  Enter the database account password when prompted:

  ```
  Enter password:
  ```

- Method 2: Using the public connection address

  ```
  ./mongo "mongodb://rwuser:****@<DB_HOST>:<DB_PORT>/test?authSource=admin&replicaSet=replica"
  ```

  To obtain the public connection address, click the instance name and choose **Connections**. The address is displayed in **Public Network Connection Address** field on the **Public Connection** tab.

NOTE

- **DB_HOST** indicates the IP address of the remotely connected DB instance. Obtain the value from the **EIP** column in the node list on the **Connections** page.
- **DB_PORT** indicates the port number. Obtain the value from **Database Port** in the **Basic Information** area on the **Connections** page.
- **DB_USER** indicates the database account name. The default value is **rwuser**.
- ********* indicates the password of the database account. If you use the connection address to connect to a DB instance:
  - If the password contains the at sign (@), change @ to %40.
  - If the password contains the exclamation mark (!), add an escape character (\) before the exclamation mark (!).

- Connect to the instance using Linux commands. The following is an example command:

  ```
  ./mongo --host 192.168.1.6 --port 8635 -u rwuser -p --authenticationDatabase admin
  ```

- Connect to the DB instance using the public connection address. The following is an example command:

  ```
  ./mongo "mongodb://rwuser:****@192.168.1.80:8635/test?authSource=admin&replicaSet=replica"
  ```

Step 3 Check the connection result. If the following information is displayed, the connection is successful.

- Result from connecting the primary node in a replica set:

  ```
  replica:PRIMARY>
  ```

- Result from connecting the secondary node in a replica set:

```
replica:SECONDARY>
```

----End

5 Connection Management

5.1 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 this document.
- 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 this document.

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 ○ to enable SSL in 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 to enable SSL in 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

5.2 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 In the **Address** area, click **Enable** to the right of **Cross-CIDR Access**, and then specify the client CIDR details

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.

NOTE

- To ensure the ECS and the DB instance can communicate with each other, configure the connection by referring to section "VPC Peering Connection Overview" in the *Virtual Private Cloud User Guide*.

----End

5.3 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 **Basic 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**.

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

----End

5.4 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:

- 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

5.5 Changing a Security Group

Scenarios

This section guides you on how to change a security group for cluster and replica set 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

6 Migrating Data

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.

Prerequisites

1. An ECS or a device that can access DDS is ready for use.
   - For details on how to connect to a DDS DB instance through a private network and how to create and log in to an ECS, see "Creating and Logging In to a Windows ECS" or "Creating and Logging In to a Linux ECS" in the *Elastic Cloud Server User Guide*.
   - 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.

```
./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

7 Account Management

7.1 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.

Prerequisites

A DDS DB instance has been connected.

- For details on how to connect to a cluster instance, see **Connecting to a Cluster Instance Over Private Networks**.
- For details on how to connect to a replica set instance, see **Connecting to a Replica Set Instance Over Private Networks**.

Account Description

To manage DDS DB instances, users **root**, **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>
```

----End

7.2 Resetting the Administrator Password

Scenarios

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

You cannot reset the administrator password under the following circumstances:

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

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: ~!@#%^*-_=+?

----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: ~!@#%^*-_=+?

----End

8 Instance Management

8.1 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

8.2 Adding Cluster Instance Nodes

Scenarios

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

NOTE

- 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**.

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

A cluster instance of Community Edition supports up to 12 mongos nodes.

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

- 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**.

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

- 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 12 shard nodes.

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

- 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

8.3 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 node 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

8.4 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 can scale up a DB instance a maximum of eight times.
- 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**.

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.

- 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 scaling-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**.

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 4 On the displayed page, confirm the storage space.

- 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 scaling-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

8.5 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

- 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.

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**.

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

- 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**.

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

- 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

8.6 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

- 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.

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**.

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

- 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

8.7 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**, 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 Replica Set Instance Over Private Networks**.

----End

8.8 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

8.9 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.

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

Step 4 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

8.10 Deleting a DB Instance

Scenarios

This section guides you on how to delete a DB instance no longer used to release resources. You can delete the following types of DB instances:

- Cluster instance
- Replica set instance

NOTICE

- 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.

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

----End

9 Backup and Restore

9.1 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 ○ .

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.
- 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.

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

9.2 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.

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

9.3 Restoring a Cluster Instance from a Backup

Scenarios

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

Restoration Precautions

- Currently, DDS Community Edition instances only support restoring backups to a new 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 in the backup list and click **Restore** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 5 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, 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 12. 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.

----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 backup to restore on the **Clusters** tab and click **Restore** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 4 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, 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 12. 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.

----End

9.4 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, DDS replica set instances only support restoring backups to a new 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 in the backup list and click **Restore** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 5 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.

----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 the **Replica Sets** tab, locate the backup to restore, and click **Restore** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 4 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.

----End

9.5 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, replica set instances only support restoring backups to a new DB instance when performing a point-in-time recovery.
- 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

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. Then, click **OK**.

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.

----End

9.6 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 OBS Browser and install it.

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 User 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 User 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 User 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

9.7 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

9.8 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

10 Parameter Group Settings

10.1 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 and replica set 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

10.2 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.

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 section **DB Instance 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

10.3 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

10.4 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

10.5 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:

- 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

10.6 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

10.7 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

10.8 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

11 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 cluster instance or replica set instance.
- Scaling up storage space

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

  Changing the class of a cluster instance or replica set instance.
- Adding nodes

  Adding nodes to a cluster instance.
- Restarting DB instances

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

  Restoring data to a new cluster instance or replica set instance.

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

12 Monitoring and Alarm Reporting

12.1 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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
mongo035_disk_usage	Disk Utilization	Disk usage of the monitored object	0–1	Monitored object: ECS Monitored object type: ● Primary node ● Secondary node
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

Metrics	Metrics Name	Description	Value Range	Remarks
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
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
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
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
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

Metrics	Metrics Name	Description	Value Range	Remarks
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

Dimensions

Key	Value
mongodb_cluster_id	DDS DB instance ID Supports cluster and replica set instances.
mongos_instance_id	mongos node ID
mongod_primary_instance_id	Primary node ID Includes the primary config and shard nodes of cluster instances and the primary nodes of replica set instances.
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.

12.2 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.

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

12.3 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 instances, cluster instance nodes, and replica set instance nodes.

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

13 Auditing

13.1 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 13-1 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	ddsBindIP
Unbinding an EIP	instance	ddsUnbindIP
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

13.2 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

14 Log Management

14.1 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.

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, click **Error Logs**.

Step 4 On the displayed page, click **Error Logs**. Then, view the log details.

- 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

----End

14.2 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.

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, click **Slow Query Logs**.

Step 4 On the **Slow Query Logs** page, set search criteria 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

----End

15 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 15-1**.
- Up to 10 tags can be added for a DB instance.

Table 15-1 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

16 FAQs

16.1 Product Consulting

16.1.1 What Precautions Should Be Taken When Using DDS?

1. Failover

   DDS uses multiple mongos, replica sets, and shards to ensure data reliability. When a mongos is faulty, the other mongos takes over services immediately to ensure service continuity. A replica set consists of a primary, a secondary, and a hidden node. When the primary node is faulty, DDS selects the secondary node as the new primary within 30 seconds.

2. ECSs used for DB instances are invisible to you. Your applications can access only the IP addresses and ports corresponding to the database.
3. Backup files stored on OBS are invisible to you. They are only visible in the DDS backend management system.
4. Precautions after applying for DDS:

   After purchasing DDS DB instances, you do not need to perform basic database O&M operations, such as HA and security patches. You need to pay attention to the following:

   a. Whether the vCPUs, IOPS, and storage space for DDS DB instances are sufficient. If any of them is insufficient, optimize or upgrade the related configuration.
   b. Whether the performance of DDS DB instances is satisfying, whether a large number of slow query statements exist, whether query statements need to be optimized, and whether any index is redundant or missing.

16.1.2 What Is the Availability of DDS DB Instances?

Formula for a DDS DB instance availability:

DDS DB instance availability = (1 – Failure duration/Total service duration) × 100%.

16.1.3 Will My DDS DB Instances Be Affected by Other Users' DDS DB Instances?

No. Your DDS DB instances and resources are isolated from others.

16.2 Database Performance

16.2.1 Does DDS Support Read/Write Splitting?

Yes. DDS can perform write operations only on the primary node in a replica set. You can configure DDS to perform read operations on the secondary node to split read and write operations.

16.2.2 What Should I Do If My Query Is Slow?

- You can view the slow query logs to check whether any slowly executed SQL queries exist and view the performance characteristics of queries (if any) to locate the cause of slow queries.

  For details on DDS log queries, see section **Slow Query Log**.
- You can also view CPU usage metrics of DDS DB instances to facilitate problem locating. For details, see section **Viewing DDS Metrics**.

16.2.3 What Is the Time Delay for Primary/Secondary Synchronization in a Replica Set?

The delay for primary/secondary synchronization cannot be calculated using a formula. The delay is affected by the following factors:

1. Network communication status
2. Transaction pressure on the primary node, that is, transactions per second (TPS) of the primary node
3. Transaction size executed by the primary node, that is, the duration of a transaction execution
4. Load of the secondary node

If the primary node bears heavy pressure within a period and executes a large number of transactions per second, the synchronization to the secondary node is delayed.

You can view **Replication Lag** of the secondary node on the Cloud Eye console to know the synchronization delay.

16.3 Creation and Deletion

16.3.1 Can I Use a Template to Create DDS DB Instances?

You do not need a template to create DDS DB instances. When you create a DB instance, DDS provides different DB instance specifications which are similar to templates.

16.3.2 Why Is Data Missing from My Database?

DDS does not delete or perform any operations on any user data. If this problem occurs, check whether a misoperation has been performed. Restore the data using backup files, if necessary.

Solution:

- Use the DDS restoration function to restore data. For details, see chapter "Backup and Restore" in the *Document Database Service User Guide*.
- Import backups to DDS. For details, see section **Migrating Data**.

16.3.3 Will My Backups Be Deleted If I Delete My Cloud Account?

If your cloud account is deleted, both your automated and manual backups are deleted.

16.4 Database Connection

16.4.1 Can an External Server Access the DDS DB Instance?

You can access the DDS DB instance using the following methods:

- If a DDS DB instance and an ECS are created in the same VPC, you can access the DDS DB instance through the ECS.
- If the DDS DB instance is publicly accessible, you can access the DB instance over public networks.

16.4.2 What Is the Number of DDS Database Connections?

The number of connections indicates the number of applications that can be simultaneously connected to the database. The number of connections is irrelevant to the maximum number of users allowed by your applications or websites.

- For a cluster instance, the number of connections indicates the number of connections between the client and the mongos.
- For a replica set instance, the number of connections indicates the number of connections between the client and the primary and secondary nodes.

16.4.3 What Should I Do If an ECS Cannot Connect to a DDS DB Instance?

Perform the following steps to identify the problem: The following uses the cluster instance as an example.

Step 1 Check whether the ECS and DDS DB instance are located in the same VPC.

- If yes, go to **Step 2**.
- If no, create an ECS in the VPC where the DDS DB instance is located.

Step 2 Check whether a security group has been added to the ECS.

- If yes, check whether the security group rules are suitable. For details, see the security group description in section **Setting a Security Group**. Then, go to **Step 3**.
- If no, go to the VPC console from the ECS details page and click **Security Groups** to add a security group.

Step 3 On the ECS, check whether the DDS DB instance address port can be connected.

```
telnet <DB instance address> {8635}
```

- If yes, the network is normal. Check the database account and password. For details, see section **Connecting to a Cluster Instance Over Private Networks**.
- If no, contact post-sales technical support for troubleshooting.

----End

16.4.4 What Should I Do If a Database Client Problem Causes a Connection Failure?

Identify a DDS DB instance connection failure caused by a client problem from the following aspects.

1. ECS security policy

   In Windows, check whether the DDS port is enabled in the Windows security policy.

   In Linux, run the **iptables** command to check whether the DDS port is enabled in firewall settings.

2. Application Configuration

   Check whether the IP address, port parameter, and Java database connectivity (JDBC) are configured correctly.

NOTE

If the problem persists, contact post-sales technical support.

16.4.5 What Should I Do If a DDS Server Problem Causes a Connection Failure?

Check whether the following problems occur on the DDS database. Check the following one at a time.

1. The maximum number of connections is reached.

   Solution: Use the Cloud Eye resource monitoring function to check whether the number of current connections and the CPU usage are normal. If the number of connections or CPU usage reaches the maximum, restart the DDS database, disconnect DB instances, or increase the node quantity.

2. DB instance status is normal, such as a restarting or system failure.

   Solution: Restart the DB instance to see if the problem is resolved. If the problem persists, contact post-sales technical support.

16.4.6 How Can My Applications Access a DDS DB Instance in a VPC?

Ensure that the ECS in which your applications are located is in the same VPC and subnet as the DDS DB instance. If the ECS and the DDS DB instance are in different subnets or VPCs, modify the VPC route table and network access control list (ACL) to ensure the ECS can access the DDS DB instance.

16.4.7 Do Applications Need to Support Automatic Reconnecting to the DDS Database?

It is recommended that your applications support automatic reconnections to the database. After a database reboot, your applications will automatically reconnect to the database to increase service availability and continuity.

In addition, you are advised to set your applications to connect to the database using a long connection to reduce resource consumption and improve performance.

16.4.8 How Do I Create and Log In to an ECS?

- The ECS to be created must be in the same VPC with the DDS DB instance to which it connects.
- When you create an ECS, select an OS, such as Red Hat 6.6, and bind an EIP to it.
- Configure the security group to enable the ECS to access the DB instance through the private IP address, that is, the node address in the **Private IP Address** column on the **Basic Information** page.

16.4.9 How Can I Install a MongoDB Client?

MongoDB official website provides client installation packages for different OSs. Download the official binary installation package at **https://www.mongodb.com/download-center#community**.

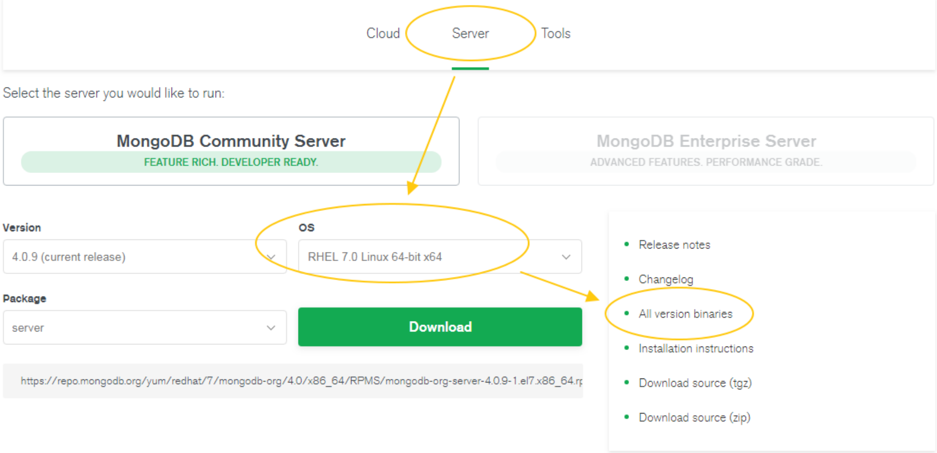
The following uses Red Hat Linux 7 and MongoDB 3.4.0 as examples to describe how to obtain the required installation package and install the MongoDB client.

Procedure

Step 1 Obtain the installation package.

1. Log in at **https://www.mongodb.com/download-center/community**.
2. Choose **Server**, select **RHEL 7.0 Linux 64-bit x64** for **OS**, and click **All version binaries**. **Figure 16-1** shows an example.

Figure 16-1 MongoDB official webpage



Cloud
Server
Tools
Select the server you would like to run:
MongoDB Community Server
FEATURE RICH. DEVELOPER READY.
MongoDB Enterprise Server
ADVANCED FEATURES. PERFORMANCE GRADE.
Download
Release notes
Changelog
All version binaries
Installation instructions
Download source (tgz)
Download source (zip)

Version: 4.0.9 (current release)

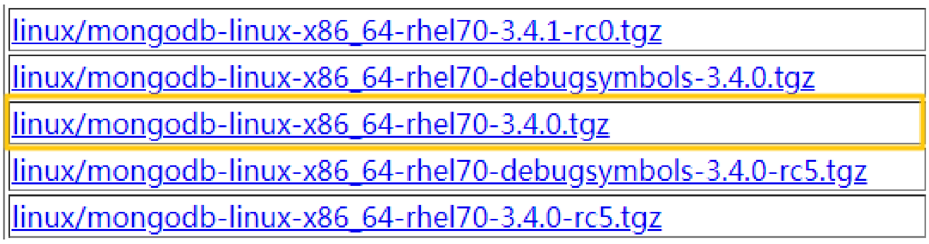
OS: RHEL 7.0 Linux 64-bit x64

Package: server

https://repo.mongodb.org/yum/redhat/7/mongodb-org/4.0/x86_64/RPMS/mongodb-org-server-4.0.9-1.el7.x86_64.rp

3. Open the downloading page, click **linux/mongodb-linux-x86_64-rhel70-3.4.0.tgz** to download the binary installation package of MongoDB 3.4.0. **Figure 16-2** shows an example.

Figure 16-2 Downloading page



linux/mongodb-linux-x86_64-rhel70-3.4.1-rc0.tgz

linux/mongodb-linux-x86_64-rhel70-debugsymbols-3.4.0.tgz

linux/mongodb-linux-x86_64-rhel70-3.4.0.tgz

linux/mongodb-linux-x86_64-rhel70-debugsymbols-3.4.0-rc5.tgz

linux/mongodb-linux-x86_64-rhel70-3.4.0-rc5.tgz

Step 2 Upload the installation package to the ECS. For details about how to log in to an ECS, see **How Do I Create and Log In to an ECS?**.

Step 3 Decompress the installation package on the ECS.

tar zxvf mongodb-linux-x86_64-rhel70-3.4.0.tgz

Step 4 Obtain the client tool from the **bin** directory of the installation package.

cd mongodb-linux-x86_64-rhel70-3.4.0/bin

The common tools are as follows:

- MongoDB client mongo
- Data export tool mongoexport
- Data import tool mongoimport

Step 5 Before using a client tool, assign the execute permission to it.

- Run the **chmod +x mongo** command to grant a client permission to connect to a DB instance.
- Run the **chmod +x mongoexport** command to grant a client permission to export data.

- Run the **chmod +x mongoimport** command to grant a client permission to import data.

Step 6 Connect to a DB instance from the client. For details, see section "Connecting to a DB Instance" in *Document Database Service Getting Started*

----End

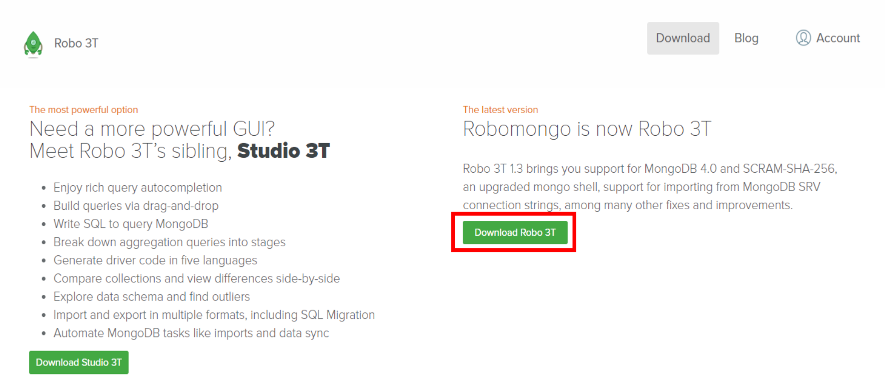
16.4.10 How Do I Install Robo 3T?

This section describes how to obtain the Robo 3T installation package and install Robo 3T.

Procedure

Step 1 Download Robo 3T from **https://robomongo.org/download**.

Figure 16-3 Downloading page



The most powerful option

Need a more powerful GUI?
Meet Robo 3T's sibling, **Studio 3T**

- Enjoy rich query autocompletion
- Build queries via drag-and-drop
- Write SQL to query MongoDB
- Break down aggregation queries into stages
- Generate driver code in five languages
- Compare collections and view differences side-by-side
- Explore data schema and find outliers
- Import and export in multiple formats, including SQL Migration
- Automate MongoDB tasks like imports and data sync

Download Studio 3T

The latest version

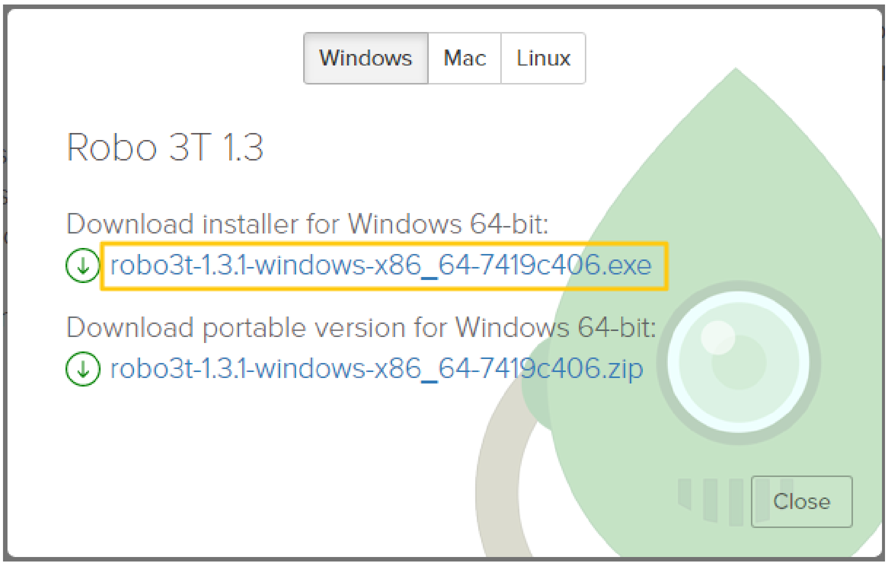
Robomongo is now Robo 3T

Robo 3T 1.3 brings you support for MongoDB 4.0 and SCRAM-SHA-256, an upgraded mongo shell, support for importing from MongoDB SRV connection strings, among many other fixes and improvements.

Download Robo 3T

Step 2 In the displayed dialog box, download **robo3t-1.3.1-windows-x86_64-7419c406.exe**.

Figure 16-4 Downloading Robo 3T



Robo 3T 1.3

Download installer for Windows 64-bit:

robo3t-1.3.1-windows-x86_64-7419c406.exe

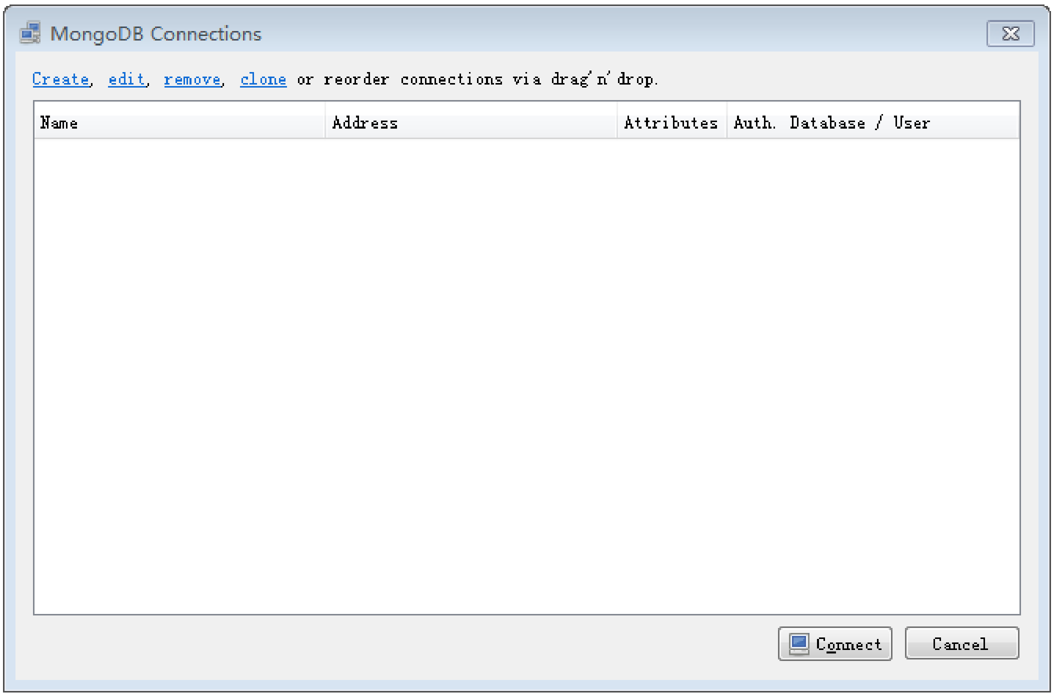
Download portable version for Windows 64-bit:

robo3t-1.3.1-windows-x86_64-7419c406.zip

Step 3 Double-click the **robo3t-1.3.1-windows-x86_64-7419c406.exe** file to start the installation.

Step 4 After the installation is complete, start the tool.

Figure 16-5 Main window



MongoDB Connections

Create, edit, remove, clone or reorder connections via drag'n'drop.

Name	Address	Attributes	Auth. Database / User

Step 5 Connect to a DB instance.

- For details on how to connect to a cluster instance, see **Connecting to a Cluster Instance Over Public Networks**.
- For details on how to connect to a replica set instance, see **Connecting to a Replica Set Instance Over Public Networks**.

----End

16.5 Database Usage

16.5.1 Are My DDS DB Instances Available When Scaling?

Yes. Adding shards does not affect the existing shards. Services are still available.

16.6 Database Storage

16.6.1 What Is the DDS DB Instance Storage Configuration?

The EVS is used for data storage of DDS DB instances. For details on the EVS, see *Elastic Volume Service User Guide*.

The DDS DB instance backup data is stored in the OBS and does not occupy the storage space you have. For details on the DDS DB instance storage configuration, see the *Object Storage Service User Guide*.

16.6.2 What Should I Do If My Data Exceeds the Database Storage Space of a DDS DB Instance?

If the storage space required by your applications exceeds the allowed maximum space allocated to you, you can do either of the following:

- Scale up storage space.
- Add shards for the DDS cluster instance of Community Edition.

16.6.3 Which Items Occupy the Storage Space of DDS DB Instances?

The following types of data will occupy the storage space:

- User data except backups
- Data required for ensuring DB instance proper running occupy, such as system database data, rollback logs, and indexes
- Log output files that are generated by DDS ensure the stable operating of DDS DB instances. For example, Oplogs occupy 10% of storage space and cannot be resized.

16.6.4 What Overhead Does the Storage Space Have After I Applied for a DDS DB Instance?

The storage space you applied for will contain the system overhead required for inode, reserved block, and database operation.

16.7 Database Parameter Modification

16.7.1 What DB Instance Monitoring Metrics Do I Need to Pay Attention To?

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.

16.8 Backup and Restoration

16.8.1 How Do I Back Up DDS Databases to an ECS?

You can store DDS backup data on the ECS using mongoexport. However, you are advised not to store database backups on ECSs. To ensure high data reliability and service assurance, you can use the DDS backup function to store backups to a professional storage object, such as OBS.

16.8.2 How Long Does DDS Store Backup Data For?

The automated backup retention period is 7 days by default. You can set a backup retention period from 1 to 732 days. There is no limit on the manual backup retention period. You can delete manual backup files as needed.

16.9 Network Security

16.9.1 What Security Protection Policies Does DDS Have?

DDS allows you to set the VPC which DDS DB instances belong to, ensuring that the DDS DB instances are isolated from other services. In addition, the IAM service is provided, achieving access control over DDS resources.

16.9.2 Do I Need to Use DDS in a VPC?

A VPC allows you to create virtual network environment in a private and isolated network to control the private IP address range, subnets, route tables, and network gateways. The VPC also allows you to define the virtual network topology and network configuration to make the network similar to the traditional IP network you are operating in the data center.

You may need to use DDS in the VPC in the following cases:

You want to run Internet-oriented web applications and retain the backend server that the public cannot access. To do so, you can create an ECS and a DDS DB instance in the same VPC, allocate a public IP address for the ECS, and deploy a web server on the ECS.

16.9.3 How Do I Ensure the Security of DDS in a VPC?

The VPC security group helps ensure the security of DDS in a VPC. In addition, ACL can be used to allow or reject I/O network traffic for each subnet. Use the internal security infrastructure (including the network firewall, intrusion detection, and protection system) to monitor all IPsec VPN connection-based input and output network traffic for VPC.

16.10 Log Management

16.10.1 Which Types of Logs and Files Occupy DDS DB Instance Storage Space?

Logs and files listed in the following table occupy storage space.

Database Type	File Type
DDS	Log files: DDS log files
	Data files: database content and index file
	Other files: some DDS temporary files

16.11 Which Commands are Supported or Restricted by DDS?

The following tables list the commands supported and restricted by DDS.

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

Table 16-1 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**.
Geospatial Commands	geoNear	√	-
	geoSearch	√	-
Query and Write Operation Commands	find	√	-
	insert	√	-
	update	√	-
	delete	√	-
	findAndModify	√	-
	getMore	√	-
	getLastError	√	-
	resetError	√	-

Type	Command	Supported	Description
	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	√	-
	revokeRolesFromUser	√	-
	usersInfo	√	-
Role Management Commands	invalidateUserCache	√	-
	createRole	√	-
	updateRole	√	-

Type	Command	Supported	Description
	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
	replSetRequestVotes	x	System command
	replSetDeclareElectionWinner	x	System command
	resync	x	System command
	applyOps	x	System command
	isMaster	√	-

Type	Command	Supported	Description
	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	√	-
	shardingState	x	System command
	unsetSharding	x	System command
	split	√	-
	splitChunk	√	-
	splitVector	√	-
	moveChunk	√	-

Type	Command	Supported	Description
	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	√	-
	compact	x	High-risk commands
	collMod	√	-
	reIndex	√	-
	setParameter	x	System configuration command
	getParameter	√	-
	repairDatabase	x	High-risk commands

Type	Command	Supported	Description
	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	√	-
	listDatabases	√	-
	netstat	x	System command
	ping	√	-
	profile	√	-
	serverStatus	√	-
	shardConnPoolStats	x	System command

Type	Command	Supported	Description
	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
System Events Auditing Commands	logApplicationMessage	x	System command

A Change History

Released On	Description
2020-06-30	This issue is the first official release.