ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

ApsaraDB for MongoDB

User Guide

Product Version: V3.16.2

Document Version: 20220727

Alibaba Cloud

Legal disclaimer

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

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

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

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

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

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

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

Document conventions

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

Table of Contents

1.Precautions

Before you begin to use ApsaraDB for MongoDB, you must familiarize yourself with the precautions and limits of the service.

To ensure the stability and security of ApsaraDB for MongoDB instances, take note of the limits described in .

ApsaraDB for MongoDB limits

Operation	Limits
Scale out nodes	• When a replica set instance is created, three nodes are added. • ApsaraDB for MongoDB provides a primary node, a secondary node, and a hidden node for each replica set instance. The primary and secondary nodes provide services, and the hidden node is invisible to you. • You cannot scale out secondary nodes.
Restart instances	You must restart an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console or by calling the API operation.

2.Log on to the ApsaraDB for MongoDB console

This topic describes how to log on to the ApsaraDB for MongoDB console.

Prerequisites

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

Procedure

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

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

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

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
   - It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
     a. On the Bind Virtual MFA Device page, bind an MFA device.
     b. Enter the account and password again as in Step 2 and click **Log On**.
     c. Enter a six-digit MFA verification code and click **Authenticate**.
   - You have enabled MFA and bound an MFA device.

     Enter a six-digit MFA authentication code and click **Authenticate**.

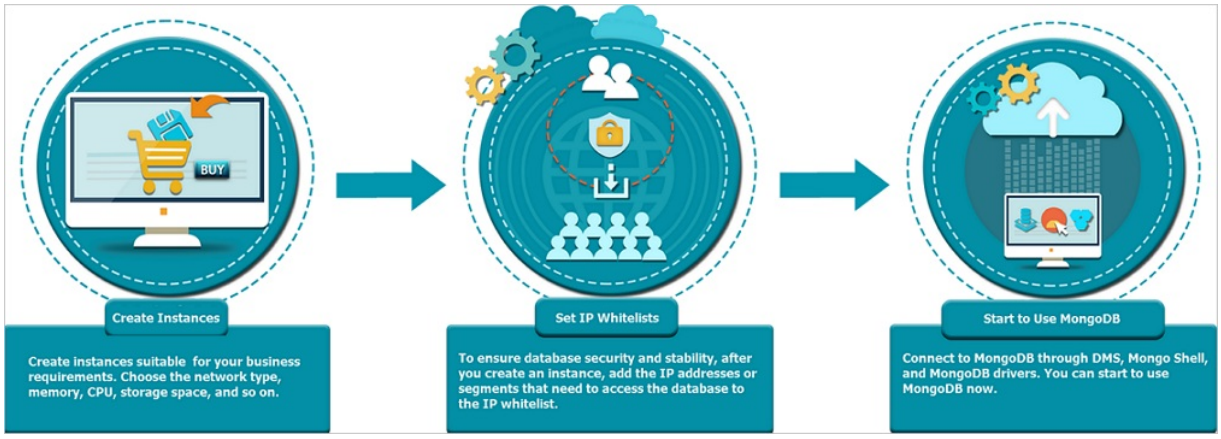
     > **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

5. In the top navigation bar, choose **Products** > **Database Services** > **ApsaraDB for MongoDB**.

3.Quick start

3.1. Use ApsaraDB for MongoDB

This topic is a quick start guide to basic usage operations for ApsaraDB for MongoDB, such as creating an instance, configuring a whitelist, and connecting to an instance. Flowcharts are used to describe the basic procedures in ApsaraDB for MongoDB, and guide you to create an ApsaraDB for MongoDB instance.



- Create an ApsaraDB for MongoDB instance

  An instance is a virtual database server on which you can create and manage multiple databases.

- Configure a whitelist for an ApsaraDB for MongoDB instance

  After you create an ApsaraDB for MongoDB instance, you need to configure a whitelist for the instance to allow external devices to access the instance.

  A whitelist can enhance access security for ApsaraDB for MongoDB instances. We recommend that you update the whitelist on a regular basis. The normal services of the instance are not affected if you configure a whitelist.

- Connect to a replica set instance by using the mongo shell

  After you create an instance and configure a whitelist, you can use the mongo shell to connect to the instance.

3.2. Create an instance

3.2.1. Create a replica set instance

This topic describes how to create a replica set instance in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click **Create Instance** in the upper-left corner.

4. On the **Create ApsaraDB for MongoDB Instance** page, configure the parameters described in the following table.

Section	Parameter	Description
Basic Settings	**Organization**	The organization where you want to deploy the instance.
	Resource Set	The resource set where you want to deploy the instance.
Region	**Region**	The region where you want to deploy the instance.
	Zone	The zone where you want to deploy the instance. **Note** If you select dual zones, the instance supports zone-disaster recovery across two data centers.
	Chip Architecture	The chip architecture of the host where you want to deploy the instance. **Note** If you do not have permissions to select an option, contact the operations administrator to grant such permissions to your account.
	Database Engine	The database engine of the instance. The value is fixed at **MongoDB**.
	Engine Version	The database engine version of the instance. Valid values: ○ 4.2 ○ 4.0 ○ 3.4 ○ 3.0
	Nodes	The number of nodes in the instance. The value is fixed at **3**.

Section	Parameter	Description
Specifications	Specifications	The instance family of the instance. Valid values: ◦ **Three-node Replica Set**: An instance of this type exclusively occupies the allocated memory and I/O resources. The instance shares CPU and storage resources with other instances of the same type deployed on the same server. ◦ **Dedicated Instance**: An instance of this type exclusively occupies CPU, memory, storage, and I/O resources to ensure stable long-term performance. In this case, the instance is not affected by other instances on the same server. ◦ **Exclusive Physics Machine**: An instance of this type exclusively occupies all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	The node specifications of the instance. Select an option based on your actual needs. For more information, see the descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the instance. This space contains data, system files, log files, and transaction files. For more information, see *Replica set instance types* in *ApsaraDB for MongoDB Product Introduction*.

Section	Parameter	Description
Network	**Network Type**	The network type of the instance. Valid values: ◦ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by using security groups or whitelists. ◦ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can specify custom route tables, CIDR blocks, and gateways within a VPC. For security purposes, we recommend that you select the VPC network type. **Note** If you select the **VPC** network type, you must specify the **VPC** and **vSwitch** parameters.
	VPC	The VPC where you want to deploy the instance. If no VPCs are available, click **Create VPC** to the right of the VPC drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	The vSwitch in the VPC. If no vSwitches are available, click **Create vSwitch** to the right of the vSwitch drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.

Section	Parameter	Description
Password Settings	**Instance Name**	The name of the instance. The name must meet the following requirements: ○ The name must start with a letter. ○ The name can contain digits, letters, underscores (_), and hyphens (-). ○ The name must be 2 to 256 characters in length.
	Password Setting	The time when you want to set the password. Valid values: ○ **Set Now**: immediately sets the logon password. ○ **Set after Purchase**: sets the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	The password used to log on to the instance. The password must meet the following requirements: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include !#$%^&*()_+= ○ The password must be 8 to 32 characters in length. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.

Section	Parameter	Description
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Logon Password. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.

5. Click **Submit**.

3.2.2. Create a sharded cluster instance

This topic describes how to create a sharded cluster instance in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click **Create Instance**.
4. On the **Create ApsaraDB for MongoDB Sharded Cluster Instance** page, configure the parameters described in the following table.

Section	Parameter	Description
Basic Settings	**Organization**	The organization where you want to deploy the instance.
	Resource Set	The resource set where you want to deploy the instance.
Region	**Region**	The region where you want to deploy the instance.
	Zone	The zone where you want to deploy the instance. **Note** If you select dual zones, the instance supports zone-disaster recovery across two data centers.

Section	Parameter	Description
Specifications	**Chip Architecture**	The chip architecture of the host where you want to deploy the instance. **Note** If you do not have permissions to select an option, contact the operations administrator to grant such permissions to your account.
	Database Engine	The database engine of the instance. The value is fixed at **MongoDB**.
	Engine Version	The database engine version of the instance. Valid values: ○ 4.2 ○ 4.0 ○ 3.4
Network	**Network Type**	The network type of the instance. Valid values: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by using security groups or whitelists. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can specify custom route tables, CIDR blocks, and gateways within a VPC. For security purposes, we recommend that you select the VPC network type. **Note** If you select the **VPC** network type, you must specify the **VPC** and **vSwitch** parameters.

Section	Parameter	Description
	VPC	The VPC where you want to deploy the instance. If no VPCs are available, click **Create VPC** to the right of the VPC drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	The vSwitch in the VPC. If no vSwitches are available, click **Create vSwitch** to the right of the vSwitch drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.
Mongos Specifications	Mongos Specifications	The specifications of the mongos node. For more information, see the descriptions in the ApsaraDB for MongoDB console.
	Quantity	The number of the mongos nodes. You can select 2 to 32 mongos nodes.
Shard Specifications	Shard Specifications	The specifications of the shard node. For more information, see the descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the shard node. The space contains the space for data, system files, log files, and transaction files. For more information, see *Sharded cluster instance types* in *ApsaraDB for MongoDB Product Introduction*.
	Quantity	The number of the shard nodes. You can select 2 to 32 shard nodes.

Section	Parameter	Description
Config Server Specifications	Config Server Type	The specifications of the Configserver node. The value is fixed at **1Cores, 2G**.
	Disk (GB)	The storage capacity of the Configserver node. The value is fixed at 20. Unit: GB.
Password Settings	Instance Name	The name of the instance. The name must meet the following requirements: ○ The name must start with a letter. ○ The name can contain digits, letters, underscores (_), and hyphens (-). ○ The name must be 2 to 256 characters in length.
	Password Setting	The time when you want to set the password. Valid values: ○ **Set Now**: immediately sets the logon password. ○ **Set after Purchase**: sets the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

Section	Parameter	Description
	Logon Password	The password used to log on to the instance. The password must meet the following requirements: ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include !#$%^&*()_+= ◦ The password must be 8 to 32 characters in length. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Logon Password. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.

5. Click **Submit**.

3.3. Reset the password

This topic describes how to reset your password in the ApsaraDB for MongoDB console.

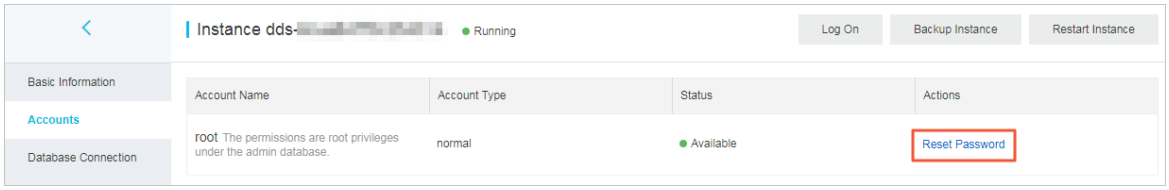
Context

Notice We recommend that you change your password on a regular basis to ensure data security.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.

3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Reset Password** in the Actions column and configure the parameters in the Reset Password panel.



describes the parameters.

Parameters for resetting a password

Parameter	Description
New Password	Specify the new password of the account based on the following rules: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include !#$%^&*()_+= ○ The password must be 8 to 32 characters in length.
Confirm New Password	Enter the password again. The password you enter here must be the same as that in New Password.

6. Click **OK**.

3.4. Configure a whitelist

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or CIDR blocks to an instance whitelist to allow access from those addresses. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

- The system creates a default whitelist for each instance. This whitelist can be modified or cleared but cannot be deleted.
- After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 127.0.0.1 to the **default** whitelist of this instance. The IP address 127.0.0.1 indicates that no IP addresses are allowed to access this instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the

console, see Log on to the ApsaraDB for MongoDB console.

2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
5. Use one of the following methods to add IP addresses to a whitelist:
   - Manually modify a whitelist
     a. Click ⋮ in the **Actions** column corresponding to a whitelist and select **Manually Modify**.
     b. In the Manually Modify panel, enter IP addresses or CIDR blocks in the **IP Whitelist** field.

     > **Note**
     > - Separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.
     >
     >   A whitelist can include IP addresses such as 0.0.0.0/0 and 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates that the prefix of the CIDR block is 24-bit long. You can replace 24 with a value within the range of 1 to 32.
     > - If the whitelist is empty or contains only `0.0.0.0/0`, all devices are granted access. This poses risks to your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

     c. Click **OK**.
   - Load IP addresses of ECS instances
     a. Click ⋮ in the **Actions** column corresponding to a whitelist and select **Import ECS Intranet IP**.
     b. In the Import ECS Intranet IP panel, select the IP addresses that you want to add to the whitelist in the **IP Whitelist** field and click > to add these IP addresses to the whitelist.
     c. Click **OK**.

3.5. Connect to an instance

3.5.1. Use DMS to log on to an ApsaraDB for MongoDB instance

You can use Data Management (DMS) to log on to an ApsaraDB for MongoDB instance.

Prerequisites

An IP address whitelist is configured. For more information about how to configure an IP address whitelist, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the upper-right corner of the page, click **Log On**.

   > **Note** For a sharded cluster instance, you must also select a mongos node.

   Log on to the **DMS** console.
5. In the **Login instance** dialog box, configure the parameters described in the following table.

Parameter	Description
Database Type	The database engine of the instance. By default, this parameter is set to the database engine of the instance that you want to access.
Instance Region	The region where the instance is deployed. By default, this parameter is set to the region where the current instance is deployed.
Connection string address	The connection string of the instance. By default, this parameter is set to the connection string of the current instance.
Database Name	The name of the database. By default, this parameter is set to admin.
Database Account	The account used to connect to the database. By default, this parameter is set to root.
Database Password	The password of the account used to connect to the database.

6. Click **Login**.

   > **Note** You can select **Remember password** to eliminate the need to manually enter the password again the next time you log on to the database.

3.5.2. Use the mongo shell to connect to an ApsaraDB for MongoDB instance

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB instance. The mongo shell is a database management tool provided by MongoDB. You can install MongoDB on your client and use the mongo shell provided by MongoDB to connect to an ApsaraDB for MongoDB instance.

Prerequisites

- The version of MongoDB that you install on your client is the same as that of your ApsaraDB for

MongoDB instance. This ensures successful authentication. For more information about the installation procedure, see Install MongoDB.

> **Note** You can select a MongoDB version corresponding to your client version in the upper-left corner of the page.

- The IP address of your client is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections** to obtain connection strings.

   > **Note**
   > - If the instance is a replica set instance, obtain the connection string or connection string URI of a node.
   > - If the instance is a sharded cluster instance, view the connection string or connection string URI of a mongos node.
   >
   > For more information about connection strings, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

5. Connect to a database of the instance from your client where the mongo shell is installed.
   - If the instance is a replica set instance, the following methods are available:

- High-availability (HA) connection (recommended)

  You can use a connection string URI to connect to both the primary and secondary nodes of the instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

  Command syntax:

  ```
  mongo "<ConnectionStringURI>"
  ```

  > Note

  - The connection string URI must be enclosed in a pair of double quotation marks ("").
  - <ConnectionStringURI>: the connection string URI of the instance.

    You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717,dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717/admin?replicaSet=mgset-*****"
  ```

- Single-node connection

  In most cases, you can directly connect to the primary, secondary, or read-only nodes. If the primary node fails, the system automatically switches to the secondary node and the secondary node becomes the primary node. This affects the read and write operations of your application.

  Command syntax:

  ```
  mongo --host <host> -u <username> -p --authenticationDatabase <database>
  ```

  > **Note**
  >
  > - <host>: the connection string of the primary or secondary node.
  > - <username>: the username used to log on to a database of the instance. The initial username is **root**.
  > - <database>: the name of the database corresponding to the username if authentication is enabled. If the username is root, enter admin.

  Example:

  ```
  mongo --host dds-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authentica
  tionDatabase admin
  ```

  When `Enter password:` is displayed, enter the password of the username and press the Enter key. If you forget the password of the root username, you can reset the password. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

  > **Note** The password characters are not masked when you enter the password.

- If the instance is a sharded cluster instance, the following methods are available:

- HA connection (recommended)

  You can use a connection string URI to connect to a database of the instance. If one mongos node fails, another mongos node takes over business to ensure the HA of the connection.

  Command syntax:

  ```
  mongo "<ConnectionStringURI>"
  ```

  Note

  - The connection string URI must be enclosed in a pair of double quotation marks ("").
  - <ConnectionStringURI>: the connection string URI of the instance.

    You must replace `****` in the connection string URI with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@s-**********.mongodb.rds.intra.env17e.shuguang.com:3717,
  s-**********.mongodb.rds.intra.env17e.shuguang.com:3717/admin"
  ```

- Mongos node connection

  Command syntax:

  ```
  mongo --host <mongos_host> -u <username> -p --authenticationDatabase <database>
  ```

  Note

  - <mongos_host>: the connection string of a mongos node in the sharded cluster instance.
  - <username>: the username used to log on to a database of the instance. The initial username is **root**.
  - <database>: the name of the database corresponding to the username if authentication is enabled. If the username is root, enter admin.

  Example:

  ```
  mongo --host s-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticati
  onDatabase admin
  ```

3.5.3. Introduction to connection strings and URIs

3.5.3.1. Overview of replica set instance connections

ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to the primary or secondary node and use a connection string URI to connect to both of them. For high availability (HA), we recommend that you use connection string URIs to connect your application to both primary and secondary nodes. This topic provides an overview of replica set instance connections.

Prerequisites

A whitelist is configured for the instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Obtain connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections** to obtain connection strings.

Description of connection strings

Item	Description
Connection address type	• Classic network endpoint: Classic network connection is a type of internal connection, and classic network endpoints are used for communication over the classic network. In the classic network, Alibaba Cloud services are not isolated. To block unauthorized traffic, you must configure security groups or IP address whitelists. • VPC endpoint: Virtual Private Cloud (VPC) connection is a type of internal connection, and VPC endpoints are used for communication over VPCs. A VPC is an isolated network that provides higher security and higher performance than the classic network. By default, ApsaraDB for MongoDB provides VPC endpoints for ApsaraDB for MongoDB instances to ensure high security and high performance. • Public endpoint: Public endpoints are used for communication over the Internet. If you connect to an ApsaraDB for MongoDB instance over the Internet, the instance may be exposed to security risks. By default, ApsaraDB for MongoDB does not provide public endpoints for ApsaraDB for MongoDB instances. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Apsara Stack (such as an on-premises device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for a sharded cluster instance.
Role	• Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. • Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance. • Connection string URI: ApsaraDB for MongoDB allows you to use a connection string URI to connect to a replica set instance to achieve load balancing and HA.
Connection string	The connection string of a primary or secondary node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port number used to connect to the instance.

Item	Description
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix of a connection string URI. • username:password@: the username and password used to log on to a database of the instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port number used to connect to the instance. • /database: the name of the database corresponding to the username if authentication is enabled. • options: the additional options that are used to connect to the instance. **Note** We recommend that you use the URI to connect to the instance in a production environment. This way, when a node fails, the read and write operations of your application are not affected by the failover.

Related information

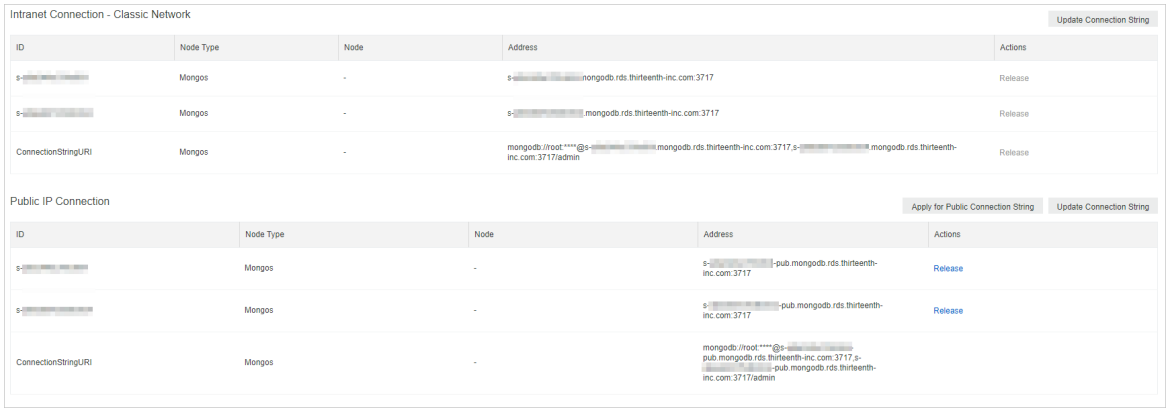
- Connect to a replica set instance by using the mongo shell

3.5.3.2. Overview of sharded cluster instance connections

ApsaraDB for MongoDB sharded cluster instances support both connection strings and connection string URIs. You can use a connection string to connect to a single mongos node and use a connection string URI to connect to multiple mongos nodes. For high availability (HA), we recommend that you use connection string URIs to connect your application to multiple mongos nodes. This topic provides an overview of sharded cluster instance connections.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click the ID of the instance whose connection strings you want to view.
4. In the left-side navigation pane, click **Database Connections** to view connection strings.



Description of connection strings

Item	Description
Connection address type	• Classic network endpoint: Classic network endpoints are used for communication over the classic network. In the classic network, Alibaba Cloud services are not isolated. To block unauthorized traffic, you must configure security groups or IP address whitelists. • VPC endpoint: Virtual Private Cloud (VPC) endpoints are used for communication over VPCs. A VPC is an isolated network that provides higher security and performance than the classic network. By default, ApsaraDB for MongoDB provides VPC endpoints for ApsaraDB for MongoDB instances to ensure high security and high performance. • Public endpoint: Public endpoints are used for communication over the Internet. If you connect to an ApsaraDB for MongoDB instance over the Internet, the instance may be exposed to security risks. By default, ApsaraDB for MongoDB does not provide public endpoints for ApsaraDB for MongoDB instances. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Apsara Stack (such as an on-premises device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.
Mongos node ID	The connection string of a mongos node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port number used to connect to the instance. **Note** During routine tests, you can use a connection string to directly connect to a mongos node.

Item	Description
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix of a connection string URI. • username:password@: the username and password used to log on to the instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port number used to connect to the instance. • /database: the name of the database corresponding to the username if authentication is enabled. • options: the additional options that are used to connect to the instance. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. Then, your client can automatically distribute your requests to multiple mongos nodes to balance loads. If a mongos node fails, your client automatically redirects requests to other mongos nodes in the normal state.

4.Instance management

4.1. Create an instance

4.1.1. Create a replica set instance

This topic describes how to create a replica set instance in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click **Create Instance** in the upper-left corner.
4. On the **Create ApsaraDB for MongoDB Instance** page, configure the parameters described in the following table.

Section	Parameter	Description
Basic Settings	Organization	The organization where you want to deploy the instance.
	Resource Set	The resource set where you want to deploy the instance.
Region	Region	The region where you want to deploy the instance.
	Zone	The zone where you want to deploy the instance. **Note** If you select dual zones, the instance supports zone-disaster recovery across two data centers.
	Chip Architecture	The chip architecture of the host where you want to deploy the instance. **Note** If you do not have permissions to select an option, contact the operations administrator to grant such permissions to your account.

Section	Parameter	Description
Specifications	**Database Engine**	The database engine of the instance. The value is fixed at **MongoDB**.
	Engine Version	The database engine version of the instance. Valid values: ○ 4.2 ○ 4.0 ○ 3.4 ○ 3.0
	Nodes	The number of nodes in the instance. The value is fixed at **3**.
	Specifications	The instance family of the instance. Valid values: ○ **Three-node Replica Set**: An instance of this type exclusively occupies the allocated memory and I/O resources. The instance shares CPU and storage resources with other instances of the same type deployed on the same server. ○ **Dedicated Instance**: An instance of this type exclusively occupies CPU, memory, storage, and I/O resources to ensure stable long-term performance. In this case, the instance is not affected by other instances on the same server. ○ **Exclusive Physics Machine**: An instance of this type exclusively occupies all resources of a server. This is the top configuration of exclusive specifications.
	Node Specifications	The node specifications of the instance. Select an option based on your actual needs. For more information, see the descriptions in the ApsaraDB for MongoDB console.

Section	Parameter	Description
	Storage Capacity (GB)	The storage space of the instance. This space contains the space for data, system files, log files, and transaction files. For more information, see *Replica set instance types* in *ApsaraDB for MongoDB Product Introduction*.
Network	**Network Type**	The network type of the instance. Valid values: ◦ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by using security groups or whitelists. ◦ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can specify custom route tables, CIDR blocks, and gateways within a VPC. For security purposes, we recommend that you select the VPC network type. **Note** If you select the **VPC** network type, you must specify the **VPC** and **vSwitch** parameters.
	VPC	The VPC where you want to deploy the instance. If no VPCs are available, click **Create VPC** to the right of the VPC drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.

Section	Parameter	Description
	vSwitch	The vSwitch in the VPC. If no vSwitches are available, click **Create vSwitch** to the right of the vSwitch drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.
Password Settings	**Instance Name**	The name of the instance. The name must meet the following requirements: ○ The name must start with a letter. ○ The name can contain digits, letters, underscores (_), and hyphens (-). ○ The name must be 2 to 256 characters in length.
	Password Setting	The time when you want to set the password. Valid values: ○ **Set Now**: immediately sets the logon password. ○ **Set after Purchase**: sets the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

Section	Parameter	Description
	Logon Password	The password used to log on to the instance. The password must meet the following requirements: ◦ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include !#$%^&*()_+= ◦ The password must be 8 to 32 characters in length. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Logon Password. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.

5. Click **Submit**.

4.1.2. Create a sharded cluster instance

This topic describes how to create a sharded cluster instance in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click **Create Instance**.

4. On the **Create ApsaraDB for MongoDB Sharded Cluster Instance** page, configure the parameters described in the following table.

Section	Parameter	Description
Basic Settings	**Organization**	The organization where you want to deploy the instance.
	Resource Set	The resource set where you want to deploy the instance.
Region	**Region**	The region where you want to deploy the instance.
	Zone	The zone where you want to deploy the instance. **Note** If you select dual zones, the instance supports zone-disaster recovery across two data centers.
Specifications	**Chip Architecture**	The chip architecture of the host where you want to deploy the instance. **Note** If you do not have permissions to select an option, contact the operations administrator to grant such permissions to your account.
	Database Engine	The database engine of the instance. The value is fixed at **MongoDB**.
	Engine Version	The database engine version of the instance. Valid values: ○ 4.2 ○ 4.0 ○ 3.4

Section	Parameter	Description
Network	Network Type	The network type of the instance. Valid values: ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by using security groups or whitelists. ○ **VPC**: A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can specify custom route tables, CIDR blocks, and gateways within a VPC. For security purposes, we recommend that you select the VPC network type. **Note** If you select the **VPC** network type, you must specify the **VPC** and **vSwitch** parameters.
	VPC	The VPC where you want to deploy the instance. If no VPCs are available, click **Create VPC** to the right of the VPC drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.
	vSwitch	The vSwitch in the VPC. If no vSwitches are available, click **Create vSwitch** to the right of the vSwitch drop-down list. **Note** If **Network Type** is set to **VPC**, you must specify this parameter.

Section	Parameter	Description
Mongos Specifications	**Mongos Specifications**	The specifications of the mongos node. For more information, see the descriptions in the ApsaraDB for MongoDB console.
	Quantity	The number of the mongos nodes. You can select 2 to 32 mongos nodes.
Shard Specifications	**Shard Specifications**	The specifications of the shard node. For more information, see the descriptions in the ApsaraDB for MongoDB console.
	Storage Capacity (GB)	The storage space of the shard node. The space contains the space for data, system files, log files, and transaction files. For more information, see *Sharded cluster instance types* in *ApsaraDB for MongoDB Product Introduction*.
	Quantity	The number of the shard nodes. You can select 2 to 32 shard nodes.
Config Server Specifications	**Config Server Type**	The specifications of the Configserver node. The value is fixed at **1Cores, 2G**.
	Disk (GB)	The storage capacity of the Configserver node. The value is fixed at 20. Unit: GB.
	Instance Name	The name of the instance. The name must meet the following requirements: ◦ The name must start with a letter. ◦ The name can contain digits, letters, underscores (_), and hyphens (-). ◦ The name must be 2 to 256 characters in length.

Section	Parameter	Description
Password Settings	Password Setting	The time when you want to set the password. Valid values: ○ **Set Now**: immediately sets the logon password. ○ **Set after Purchase**: sets the logon password after you create the instance. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
	Logon Password	The password used to log on to the instance. The password must meet the following requirements: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The supported special characters include !#$%^&*()_+= ○ The password must be 8 to 32 characters in length. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.
	Confirm Password	Enter the password again. The password you enter here must be the same as that in Logon Password. **Note** If **Password Setting** is set to **Set Now**, you must specify this parameter.

5. Click **Submit**.

4.2. View the details of an ApsaraDB for MongoDB instance

This topic describes how to view the details of an ApsaraDB for MongoDB instance, such as the basic information, internal network connection information, status, and configurations.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. Go to the basic information page by using one of the following methods:
   - Find the instance that you want to view and click its ID to go to the **Basic Information** page. Then, you can view the details of the instance.
   - Click in the **Actions** column corresponding to the instance that you want to view and select **Manage** to go to the **Basic Information** page. Then, you can view the details of the instance.

4.3. Restart an instance

This topic describes how to restart an instance when instance connections reach the upper limit or your instances have performance issues. This topic describes how to restart an ApsaraDB for MongoDB instance.

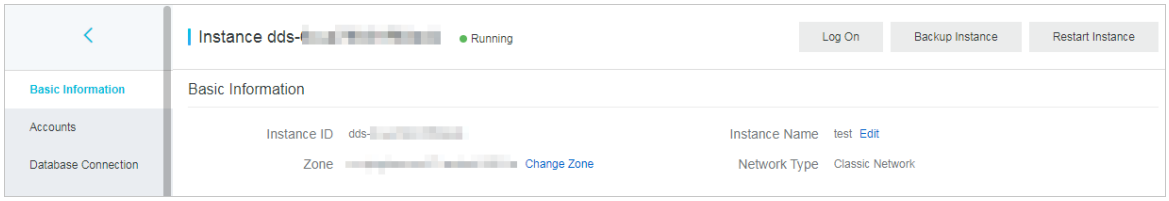
Prerequisites

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

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click in the **Actions** column corresponding to the instance and select **Manage**.
4. Click **Restart Instance** in the upper-right corner.

> **Note**
>
> - When an ApsaraDB for MongoDB instance is restarted, all its connections are closed. Plan your operations in advance before you restart an ApsaraDB for MongoDB instance.
> - You can also find the instance that you want to restart, click in the **Actions** column, and then select **Restart**.



5. In the **Restart Instance** message, click **OK**.

4.4. Change the configurations of an ApsaraDB for MongoDB instance

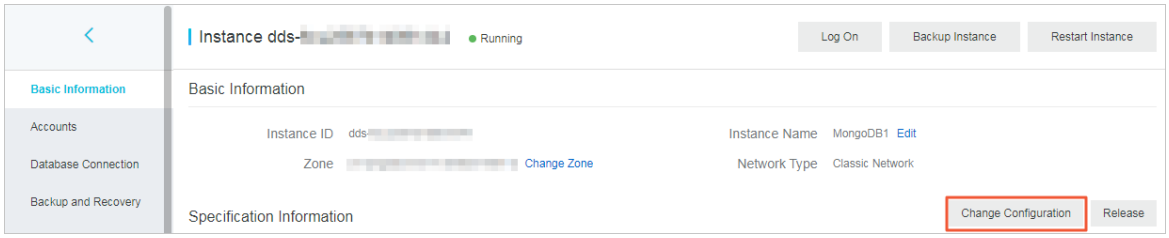
This topic describes how to change the configurations of an ApsaraDB for MongoDB instance. You can upgrade or downgrade an ApsaraDB for MongoDB instance to meet your business needs.

Prerequisites

The instance is an ApsaraDB for MongoDB replica set instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. Click **Change Configuration** in the upper-right corner of the Specification Information section to go to the **Change Specifications** page.



> **Note** To go to the **Change Specifications** page, you can also choose ⋮ > **Change Configuration** in the **Actions** column corresponding to the instance on the **Replica Set Instances** page.

5. On the **Change Specifications** page, change the instance configurations.

   You can change values of the following parameters:

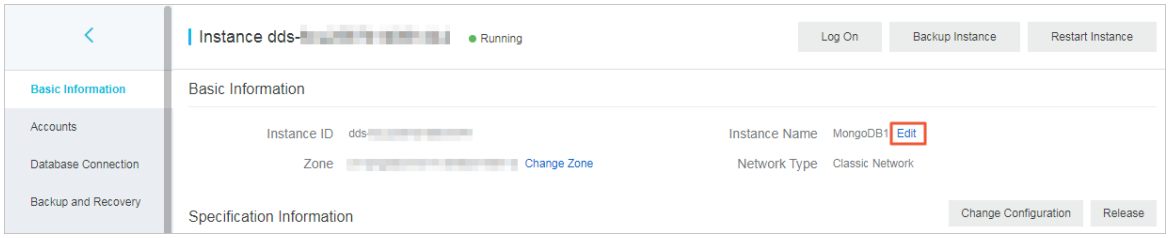
   - **Node Type**
   - **Node Specifications**
   - **Storage Capacity (GB)**
6. After you change the instance configurations, click **Submit**.

4.5. Change the name of an instance

This topic describes how to change the name of an ApsaraDB for MongoDB instance to facilitate management.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. Click **Edit** to the right of **Instance Name**.



> **Note**
> - The name must start with a letter. The name cannot start with http:// or https://.
> - The name can contain letters, underscores (_), hyphens (-), and digits.
> - The name must be 2 to 128 characters in length.

5. Click **OK**.

4.6. Reset the password for an ApsaraDB for MongoDB instance

This topic describes how to reset your password in the ApsaraDB for MongoDB console.

Context

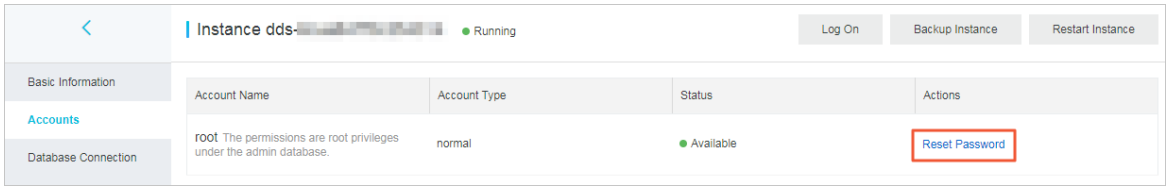
> **Notice** We recommend that you change your password on a regular basis to ensure data security.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or

click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.

4. In the left-side navigation pane, click **Accounts**.
5. Click **Reset Password** in the Actions column and configure the parameters in the Reset Password panel.



describes the parameters.

Parameters for resetting a password

Parameter	Description
New Password	Specify the new password of the account based on the following rules: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include !#$%^&*()_+= ○ The password must be 8 to 32 characters in length.
Confirm New Password	Enter the password again. The password you enter here must be the same as that in New Password.

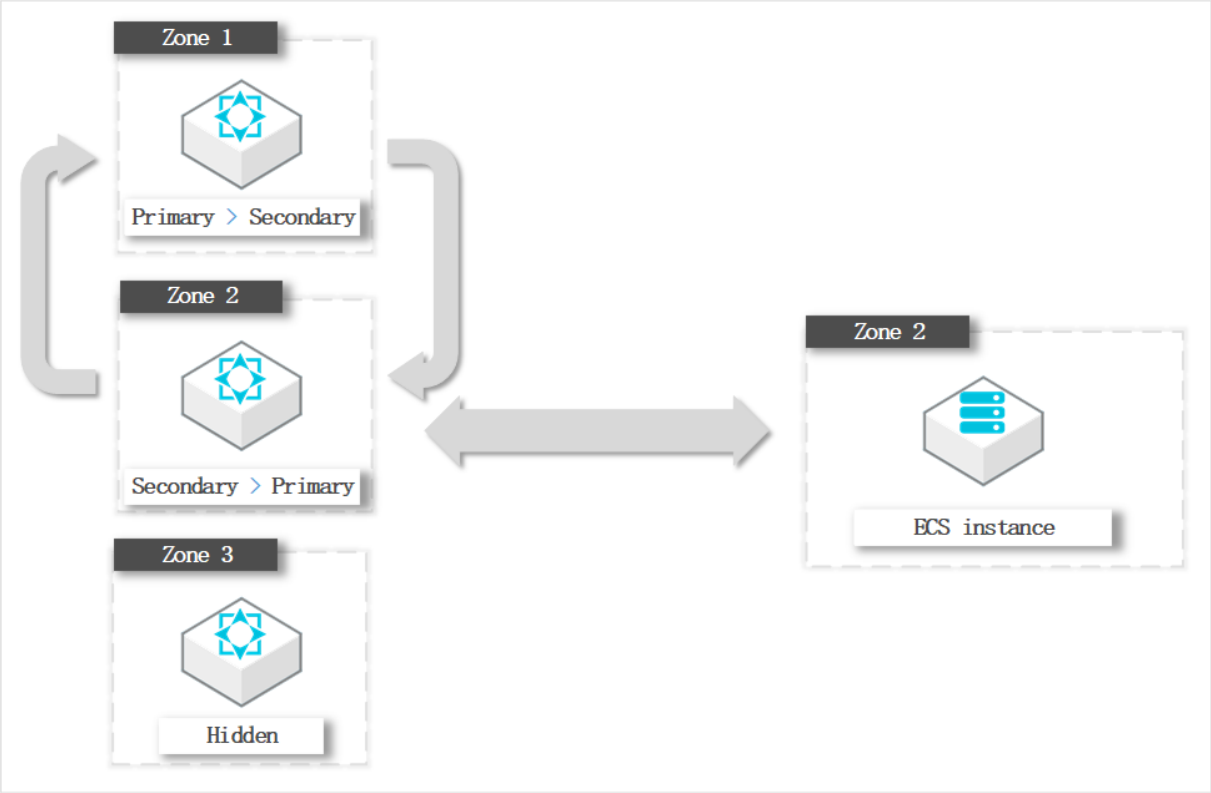
6. Click **OK**.

4.7. Switch node roles

You can switch the node roles of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console based on your business deployment.

Typical scenario

When an Elastic Compute Service (ECS) instance and an ApsaraDB for MongoDB instance are connected in the same zone over an internal network, the latency is minimal. If they are connected across different zones, the latency increases and the performance of the ApsaraDB for MongoDB instance and your business is affected.



In this example, the ECS instance to which the application belongs is in Zone 2. If the primary node of the ApsaraDB for MongoDB instance is in Zone 1, the ECS instance needs to connect to the primary node across zones.

To optimize the business deployment architecture, you can switch roles between the primary and secondary nodes. In this example, you can change the role of the node in Zone 2 to primary and the role of the node in Zone 1 to secondary. The ECS and ApsaraDB for MongoDB instances can be connected in the same zone.

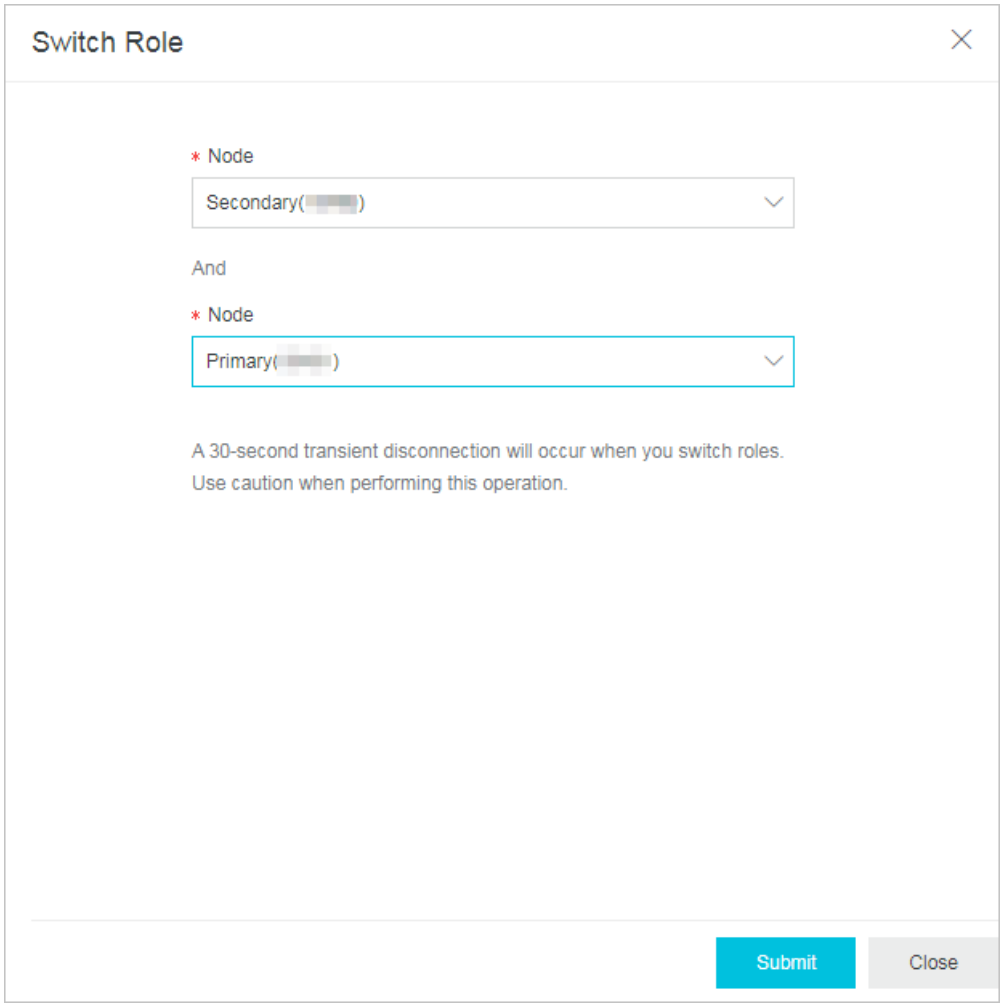
Precautions

- Each time you switch node roles, the instance may experience a transient connection of up to 30 seconds. Perform this operation during off-peak hours or make sure that your application can automatically re-establish a connection.
- Each time you switch node roles, only the node roles are changed and the zones and role IDs of nodes remain unchanged.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Service Availability**.
5. On the **Service Availability** page, perform the subsequent operations based on the corresponding instance architecture.
   - Replica set instances
     a. Click **Switch Role** in the upper-right corner of the page.

b. In the **Switch Role** panel, select the node for which you want to switch the role.



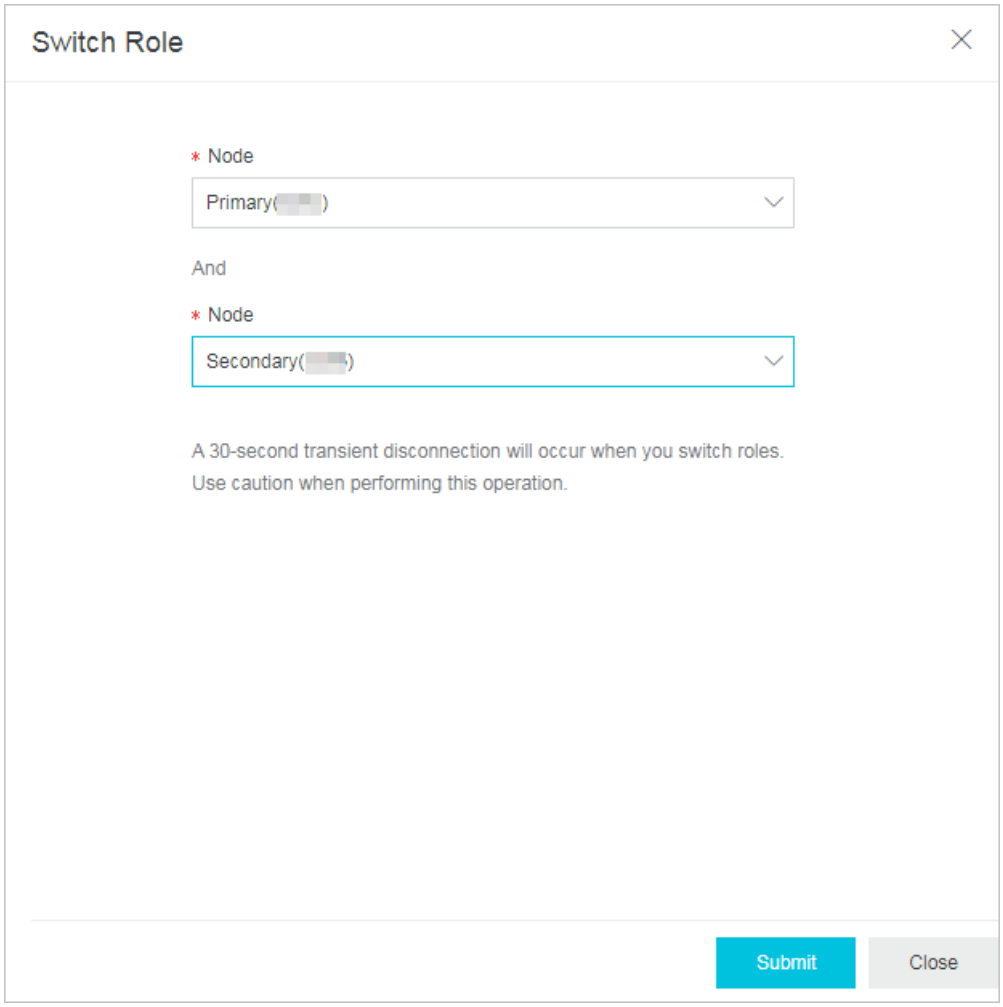
c. Click **OK**.

- Sharded cluster instances

> **Note** For sharded cluster instances, you can manage only the zone distribution of shard and Configserver nodes.

a. In the upper-right corner of the **Zone Distribution for Shards** or **Zone Distribution for Configservers** section, click **Switch Role**.

b. In the **Switch Role** panel, select the node for which you want to switch the role.



c. Click **OK**.

4.8. Migrate an instance across multiple zones

This topic describes how to migrate an ApsaraDB for MongoDB instance to a different zone within the same region. After an ApsaraDB for MongoDB instance is migrated to a different zone, the attributes, specifications, and connection addresses of the instance remain unchanged.

Prerequisites

- The ApsaraDB for MongoDB instance is a replica set instance or sharded cluster instance that runs MongoDB 4.2 or earlier.
- Transparent data encryption (TDE) is disabled for the ApsaraDB for MongoDB instance.
- The source zone and the destination zone belong to the same region.
- If the instance is deployed in a virtual private cloud (VPC), make sure that a vSwitch is created in the destination zone before you start migration. For more information, see *Create a VPC* and *Create a vSwitch* in *VPC User Guide*.
- The ApsaraDB for MongoDB instance does not have a public endpoint. If you have applied for a

public endpoint, you must release the public endpoint before you migrate the ApsaraDB for MongoDB instance to a different zone. For more information about how to release a public endpoint, see Release a public connection string.

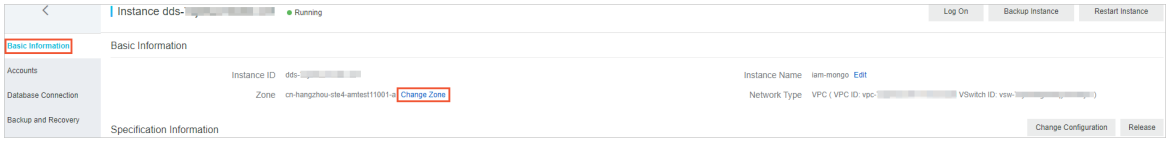
Precautions

- If the ApsaraDB for MongoDB instance resides in a VPC, the destination zone must also belong to the same VPC.
- The amount of time that is required for the migration varies based on factors such as the network conditions, task queue state, and data volume. We recommend that you migrate the instance during off-peak hours.
- During the migration, you may experience transient connections that last about 30 seconds. Make sure that your application is configured to automatically reconnect to the ApsaraDB for MongoDB instance.
- After the migration, the virtual IP addresses (VIPs) of the ApsaraDB for MongoDB instance change. A sample VIP is 172.16.88.60. If your application is connected to the ApsaraDB for MongoDB instance by using an original VIP after the migration, the application disconnects from the instance.

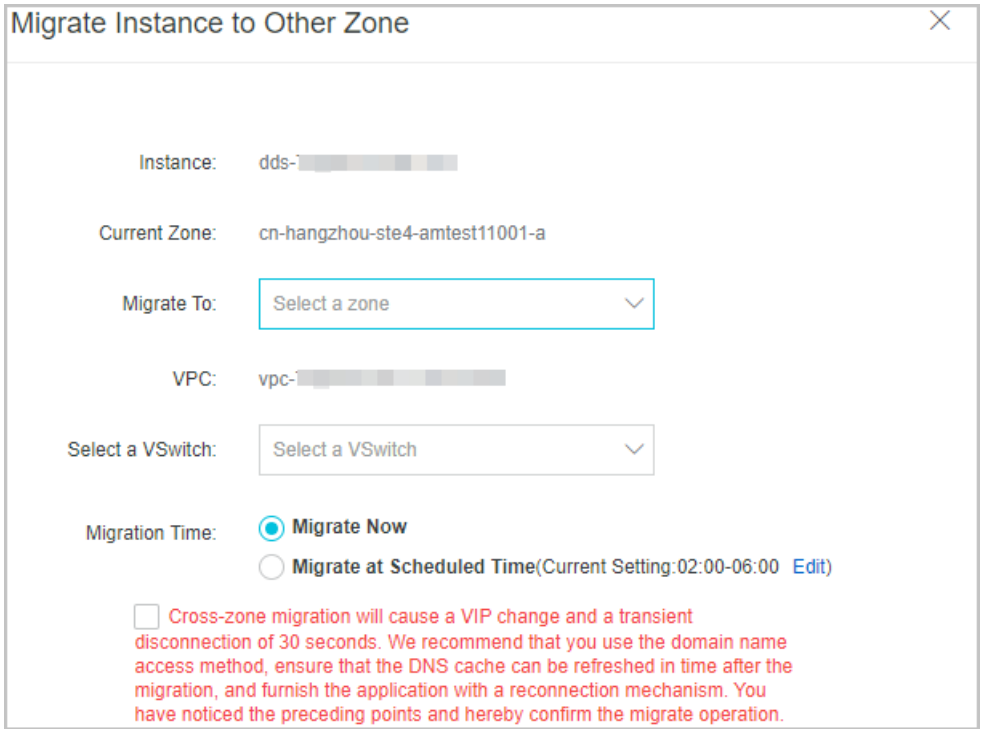
> **Note** We recommend that you connect your application to the ApsaraDB for MongoDB instance by using a connection string URI of the instance to ensure high availability. For more information, see Overview of replica set instance connections and Overview of sharded cluster instance connections.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the **Basic Information** section, click **Change Zone**.

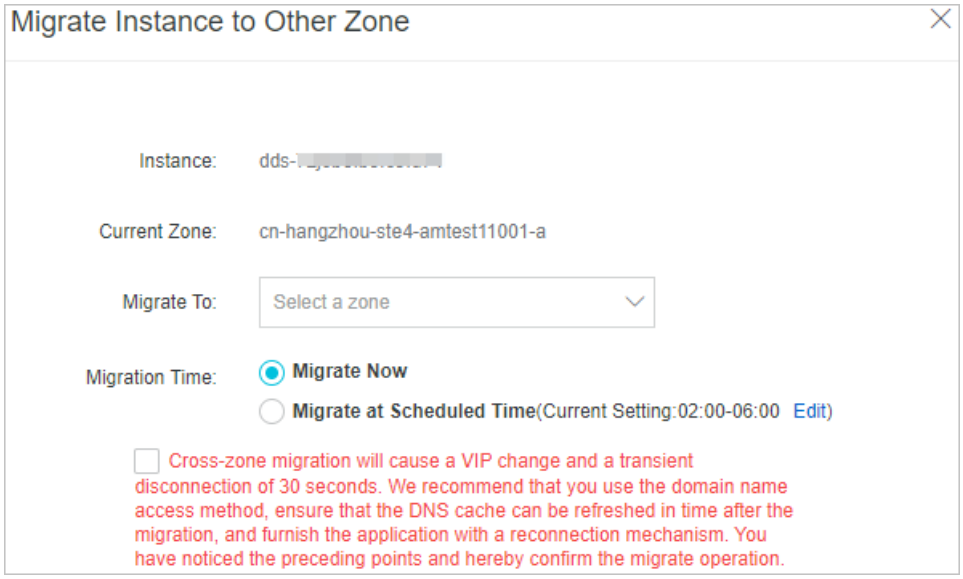


5. In the **Migrate Instance to Other Zone** panel, configure the parameters based on the network type of the instance.
   - Migrate an ApsaraDB for MongoDB instance in a VPC to a different zone



Parameter	Description
Migrate To	Select the destination zone.
Select a vSwitch	Select the destination vSwitch.
Migration Time	Select the time when you want to start the migration. Valid values: ■ **Switch Immediately after Migration**: The migration immediately starts after you configure the parameters. ■ **Switch within Maintenance Window**: The migration starts during the specified period. You can click **Edit** to the right of **Switch within Maintenance Window** to change the period.

- Migrate an ApsaraDB for MongoDB instance in the classic network to a different zone



Parameter	Description
Migrate To	Select the destination zone.
Migration Time	Select the time when you want to start the migration. Valid values: ■ **Switch Immediately after Migration**: The migration immediately starts after you configure the parameters. ■ **Switch within Maintenance Window**: The migration starts during the specified period. You can click **Edit** to the right of **Switch within Maintenance Window** to change the period.

6. Read the prompt and select the check box.
7. Click **Submit**.

4.9. Release an ApsaraDB for MongoDB instance

This topic describes how to manually release an ApsaraDB for MongoDB instance to meet your business needs.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. On the page that appears, click **Release** in the lower-right corner of the **Basic Information** section.

> **Note** You can also click ⋮ in the **Actions** column corresponding to the instance and select **Release**.

5. In the **Release Instance** message, click **OK**.

> **Warning** After you release an ApsaraDB for MongoDB instance, data in the instance can no longer be recovered. Proceed with caution.

4.10. Primary/secondary failover

4.10.1. Configure a primary/secondary switchover for a replica set instance

By default, an ApsaraDB for MongoDB replica set instance consists of three nodes. ApsaraDB for MongoDB provides connection strings for you to connect to the primary node and a secondary node. The other secondary node is hidden as a backup to ensure high availability (HA). If a node fails, the HA system of ApsaraDB for MongoDB automatically triggers a primary/secondary switchover to ensure the availability of the instance. You can also manually trigger a primary/secondary switchover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

Prerequisites

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

Context

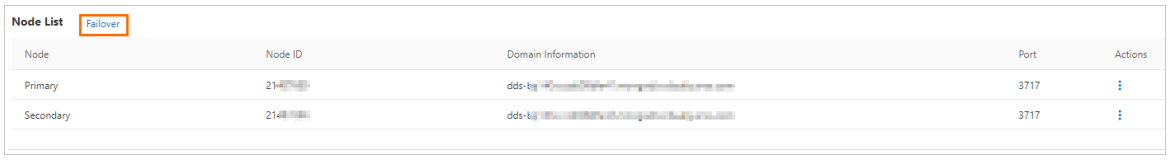
After a primary/secondary switchover is triggered for a replica set instance, the system switches roles between the primary and secondary nodes in the instance.

Precautions

Each time you trigger a primary/secondary switchover for an instance, the instance may experience a transient connection of up to 30 seconds. Make sure that your applications can automatically re-establish a connection.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
3. In the **Node List** section, click **Failover**.



4. In the **Failover** message, click **OK**.
5. The instance state changes to **Switching Role**. When the instance state changes to **Running**, the switchover is successful.

> **Note**
>
> - The switchover operation is complete in about 1 minute.
> - After the switchover operation, you must use the connection string of the new primary node to connect to the instance if your original connection uses the connection string of the original primary node. This is because the original primary node becomes a secondary node that has no write permissions after the switchover. For more information, see Overview of replica set instance connections.

4.10.2. Configure a primary/secondary failover for a sharded cluster instance

By default, each shard or Configserver node of a sharded cluster instance consists of three nodes. If a node fails, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to ensure the availability of the shard or Configserver node. You can also manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

Prerequisites

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

Context

After a primary/secondary failover is triggered, the system switches roles between the primary and secondary nodes in a shard node.

Precautions

Each time you trigger a primary/secondary failover for an instance, the instance may experience a transient connection of up to 30 seconds. We recommend that you perform this operation during off-peak hours and ensure that your applications can automatically re-establish a connection.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the **Shard List** or **ConfigServer List** section, click  in the **Actions** column corresponding to the node for which you want to perform failover and select **Failover**.

> **Note** You can separately trigger a primary/secondary failover for each shard node. The failover operation takes effect only for the current shard node and does not affect other shard nodes of the same sharded cluster instance.

5. In the **Failover** message, click **OK**.

> **Note** The failover operation is complete in about one minute.

4.11. View monitoring data

This topic describes the performance metrics provided by ApsaraDB for MongoDB to check the status of ApsaraDB for MongoDB instances. You can view instance monitoring data in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Monitoring Data**.
5. On the page that appears, specify a time range to view the monitoring data.

   Instance monitoring data is collected every 300 seconds.

Metric	Description
CPU utilization	cpu_usage: the CPU utilization of the instance.
Memory usage	mem_usage: the memory usage of the instance.
IOPS usage	The IOPS of the instance. The following items are included: ○ data_iops: the IOPS of the data disk. ○ log_iops: the IOPS of the disk that stores logs.
IOPS usage percentage	iops_usage: the ratio of the IOPS used by the instance to the maximum IOPS allowed.
Disk usage	The total disk space used by the instance. The following items are included: ○ ins_size: the total space used. ○ data_size: the disk space used by data files. ○ log_size: the disk space used by log files.
Disk usage percentage	disk_usage: the ratio of the total disk space used by the instance to the maximum disk space that can be used.

Metric	Description
QPS	The queries per second (QPS) of the instance. The following items are included: ◦ The number of insert operations. ◦ The number of query operations. ◦ The number of delete operations. ◦ The number of update operations. ◦ The number of getMore operations. ◦ The number of command operations.
Connections	current_conn: the number of current connections to the instance.
Cursors	The number of cursors used by the instance. The following items are included: ◦ total_open: the number of cursors that are opened. ◦ timed_out: the number of cursors that timed out.
Network traffic	The network traffic of the instance. The following items are included: ◦ bytes_in: the inbound network traffic. ◦ bytes_out: the outbound network traffic. ◦ num_requests: the number of requests that are processed.
Read/write queues	The length of the queues that are waiting for global locks for the instance. The following items are included: ◦ gl_cq_total: the length of the queue that is waiting for both global read and write locks. ◦ gl_cq_readers: the length of the queue that is waiting for global read locks. ◦ gl_cq_writers: the length of the queue that is waiting for global write locks.
WiredTiger	The cache metrics of the WiredTiger engine used by the instance. The following items are included: ◦ bytes_read_into_cache: the amount of data that is read into the cache. ◦ bytes_written_from_cache: the amount of data that is written from the cache to the disk. ◦ maximum_bytes_configured: the size of the maximum available disk space that is configured.
Primary/secondary replication latency	repl_lag: the latency in data synchronization between the primary and secondary nodes of the instance.

5.Backup and restoration

5.1. Data backup

5.1.1. Configure automatic backup for an ApsaraDB for MongoDB instance

This topic describes how to configure automatic backup for an ApsaraDB for MongoDB instance. ApsaraDB for MongoDB can automatically back up data based on the default backup policy or the backup policy that you specify.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of the instance for which you want to configure the backup policy.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. In the upper-left corner of the page, click **Backup Settings**.
6. In the **Backup Settings** panel, configure the parameters described in the following table.

Parameter	Description
Retention Days	The backup retention period that is fixed at seven days.
Backup Time	The period of time during which you want to back up data. We recommend that you specify a time period that is during off-peak hours.
Day of Week	The days of a week on which you want to back up data. You can select multiple days.

7. Click **OK**.

5.1.2. Manually back up an ApsaraDB for MongoDB instance

This topic describes how to manually back up an ApsaraDB for MongoDB instance. ApsaraDB for MongoDB supports both automatic backup and manual backup. You can configure a backup policy for the system to automatically back up your ApsaraDB for MongoDB instance based on the backup cycle you specify.

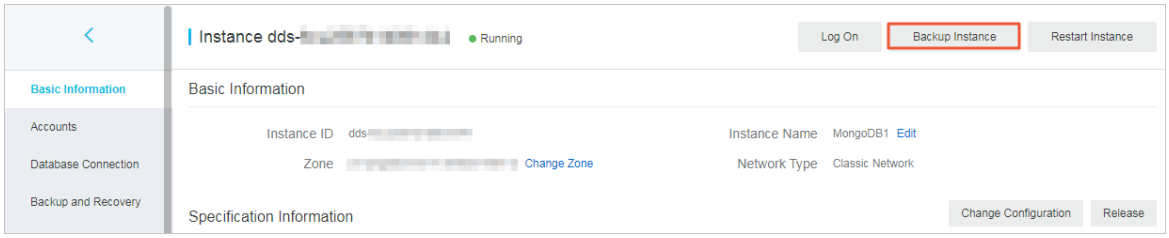
Backup methods

- Physical backup: This method backs up physical database files of an ApsaraDB for MongoDB instance. Compared with logical backup, physical backup provides faster data backup and restoration.

- Logical backup: The mongodump tool is used to store operation records of databases in a logical backup file. Logical backup restores data in the form of playback commands during restoration.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click ⋮ in the **Actions** column corresponding to the instance and select **Manage**.
4. In the upper-right corner of the page, click **Back up Instance**.



5. In the **Back up Instance** panel, select a backup method from the **Backup Method** drop-down list and then click **OK**.

5.2. Download a backup file

ApsaraDB for MongoDB allows you to download backup files that can be used to restore databases. This topic describes how to download a backup file in the ApsaraDB for MongoDB console.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of the instance for which you want to configure the backup policy.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. On the page that appears, click ⋮ in the **Actions** column corresponding to a backup file and select **Download**.
6. In the **Download Backup** dialog box, click **Copy** next to the download URL and then click **OK**.
7. Select an appropriate method to download the backup file based on your business environment. You can run the wget command, or paste the copied download URL into the address bar of your browser and press the Enter key to download the backup file.

5.3. Data restoration

5.3.1. Restore backup data to a new ApsaraDB for MongoDB instance by point in time

ApsaraDB for MongoDB allows you to create an instance at a point in time when the current instance is running and restore the backup data at that point in time to the new instance. This method is suitable for data restoration and verification.

Limits

Only points in time from the last seven days can be selected.

Precautions

- Individual databases can be restored only from physical backups. If your instance runs a MongoDB version earlier than 4.0 and the total number of collections and indexes in your instance exceeds 10,000, physical backups may fail.
- The amount of time required to restore data of individual databases varies based on factors such as the data volume, task queue state, and network conditions. When the state of the new instance changes to **Running**, the restoration is complete.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. Select the databases that you want to restore.
   i. On the **Backup and Recovery** page, click **Create Instance By Time Point**.

ii. In the **Create Instance By Time Point** panel, specify the following parameters.

Parameter	Description
Select recovery time point	Selects a point in time from which you want to restore data. Points in time from the last seven days can be selected. **Note** The point in time you select must be earlier than the current time and later than the time when the source instance was created.
Select databases to recover	■ **All Databases**: All databases in the source instance are restored. ■ **Select Databases**: Only selected databases are restored. You can directly select the databases that you want to restore or click **Enter Databases** to enter the names of the databases that you want to restore. **Note** ■ If you enter the names of the databases, separate the names with commas (,). ■ By default, all databases are restored for sharded cluster instances. You can skip this step if the instance is a sharded cluster instance. ■ If the instance is a sharded cluster instance, do not select the latest point in time (usually the latest hour). Otherwise, the restoration will fail.

iii. Click **OK**.

6. On the **Clone ApsaraDB for MongoDB instance** page, configure the parameters of the new instance. For more information, see Create a replica set instance or Create a sharded cluster instance.

   Note
   - If the instance is a replica set instance, the storage capacity of the new instance must be greater than or equal to that of the source instance.
   - If the instance is a sharded cluster instance, the following requirements must be met:
     - The number of shard nodes in the new instance must be equal to that in the source instance.
     - The storage capacity of each shard node in the new instance must be greater than or equal to that in the source instance.

7. Click **Submit**.

5.3.2. Restore backup data to a new ApsaraDB for MongoDB instance by backup point

ApsaraDB for MongoDB allows you to restore data to a new ApsaraDB for MongoDB instance by backup point. This method is suitable for data restoration and verification.

Prerequisites

The instance is a replica set instance.

Limits

Only backup points from the last seven days can be selected.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. Select the databases that you want to restore.
   i. Find the backup file that you want to use for data restoration. Choose  > **Create Instance from Backup Point** in the **Actions** column.
   ii. In the **Create Instance from Backup Point** panel, select the databases that you want to restore.
   - **All Databases**: All databases in the source instance are restored.
   - **Select Databases**: Only selected databases are restored.

     You can directly select the databases that you want to restore or click **Enter Databases** to enter the names of the databases that you want to restore.

     > **Note** If you enter the names of the databases, separate the names with commas (,).

   iii. Click **OK**.
6. On the **Clone ApsaraDB for MongoDB instance** page, configure the parameters of the new instance. For more information, see Create a replica set instance or Create a sharded cluster instance.

   > **Note** The storage capacity of the new instance must be greater than or equal to that of the source instance.

7. Click **Submit**.

5.3.3. Restore data to the current ApsaraDB for MongoDB instance

This topic describes how to restore data to the current ApsaraDB for MongoDB instance. This helps minimize the data loss caused by incorrect operations.

Prerequisites

The instance is a replica set instance with three nodes.

Background information

- The time required to restore data to your current instance varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the instance changes to **Running**, the restoration is complete.
- If you restore data to your current instance, all existing data is overwritten and cannot be restored.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
3. In the left-side navigation pane, click **Backup and Recovery**.
4. On the **Backup and Recovery** page that appears, find the backup set and choose  > **Data Recovery** in the Actions column.

   > **Note** If you have upgraded the database version, you cannot use the backup files of the earlier database version to restore data.

5. In the **Roll Back Instance** message, click **OK**.

   > **Note** The instance status becomes **Restoring from Backup** after you click **OK**. You can click **Refresh** in the upper-right corner of the **Backup and Recovery** page to update the instance status. The restoration is complete when the instance status changes to **Running**.

6.Database connections

6.1. Modify a public or internal endpoint of an ApsaraDB for MongoDB instance

This topic describes how to modify a public or internal endpoint of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console.

Limits

Architecture	Limits
Replica set instance	You can modify only the public and internal endpoints of primary and secondary nodes.
Sharded cluster instance	You can modify only the public and internal endpoints of mongos nodes.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of the instance whose endpoint you want to modify.
4. In the left-side navigation pane, click **Database Connection**.
5. In the **Internal Connections** or **Public Connections** section, click **Update Connection String**.
6. In the **Update Connection String** panel, specify a new endpoint.

   > **Note** Only the prefix of the endpoint can be modified. You must comply with the following rules when you modify the prefix:
   >
   > - The prefix must start with a lowercase letter.
   > - The prefix must start or end with a lowercase letter or digit.
   > - The prefix must be 8 to 64 characters in length and can contain lowercase letters, digits, and hyphens (-).

7. Click **Submit**.

What's next

After you modify the public or internal endpoint, you must connect a client or an application to your ApsaraDB for MongoDB instance by using the new endpoint.

6.2. Use DMS to log on to an ApsaraDB for MongoDB instance

You can use Data Management (DMS) to log on to an ApsaraDB for MongoDB instance.

Prerequisites

An IP address whitelist is configured. For more information about how to configure an IP address whitelist, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance, or click  in the **Actions** column corresponding to the instance and select **Manage**.
4. In the upper-right corner of the page, click **Log On**.

   > **Note** For a sharded cluster instance, you must also select a mongos node.

   Log on to the **DMS** console.
5. In the **Login instance** dialog box, configure the parameters described in the following table.

Parameter	Description
Database Type	The database engine of the instance. By default, this parameter is set to the database engine of the instance that you want to access.
Instance Region	The region where the instance is deployed. By default, this parameter is set to the region where the current instance is deployed.
Connection string address	The connection string of the instance. By default, this parameter is set to the connection string of the current instance.
Database Name	The name of the database. By default, this parameter is set to admin.
Database Account	The account used to connect to the database. By default, this parameter is set to root.
Database Password	The password of the account used to connect to the database.

6. Click **Login**.

   > **Note** You can select **Remember password** to eliminate the need to manually enter the password again the next time you log on to the database.

6.3. Use the mongo shell to connect to an ApsaraDB for MongoDB instance

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB instance. The mongo shell is a database management tool provided by ApsaraDB for MongoDB. You can install it on your client or in an Elastic Compute Service (ECS) instance.

Prerequisites

- The version of the mongo shell is the same as that of your ApsaraDB for MongoDB instance. This ensures successful authentication. For more information about the installation procedure, see Install MongoDB.

  > **Note** You can select a MongoDB version corresponding to your client version in the upper-left corner of the page.

- The IP address of your client is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections** to view connection strings.

   > **Note**
   > - Replica set instances: View the connection string or connection string URI of a node.
   > - Sharded cluster instances: View the connection string or connection string URI of a mongos node.
   >
   > For more information about connection strings, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

5. Connect to a database of the instance from your client or ECS instance where the mongo shell is installed.
   - Replica set instances

- High-availability connection (recommended)

  You can use a connection string URI to connect to both the primary and secondary nodes of the instance. This ensures that your application is always connected to the primary node and the read and write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

  Command syntax:

  ```
  mongo "<ConnectionStringURI>"
  ```

  Note

  - The connection string URI must be enclosed in a pair of double quotation marks ("").
  - <ConnectionStringURI>: the **connection string URI** of the instance.

    You must replace `****` in the **connection string URI** with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717,dds-**********.mongodb.rds.intra.env17e.shuguang.com:3717/admin?replicaSet=mgset-*****"
  ```

- Single-node connection

  In most cases, you can directly connect to the primary, secondary, or read-only node. When the primary node fails, the system automatically switches to the secondary node and the secondary node becomes the primary node. This affects the read and write operations of your application.

  Command syntax:

  ```
  mongo --host <host> -u <username> -p --authenticationDatabase <database>
  ```

  > **Note**
  >
  > - <host>: the endpoint used to log on to the primary or secondary node.
  > - <username>: the username used to log on to a database of the instance. The initial username is **root**.
  > - <database>: the name of the database corresponding to the username if authentication is enabled. If the username is root, enter admin as the database name.

  Example:

  ```
  mongo --host dds-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authentica
  tionDatabase admin
  ```

  When `Enter password:` is displayed, enter the password of the username and press the Enter key. If you forget the password of the root username, you can reset the password. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

  > **Note** The password characters are not displayed when you enter the password.

- Sharded cluster instances

- High-availability connection (recommended)

  You can use a connection string URI to connect to a database of the instance. If one mongos node fails, another mongos node takes over business to ensure the high availability of the connection.

  Command syntax:

  ```
  mongo "<ConnectionStringURI>"
  ```

  Note

  - The connection string URI must be enclosed in a pair of double quotation marks ("").
  - <ConnectionStringURI>: the **connection string URI** of the instance.

    You must replace `****` in the **connection string URI** with the database password. For more information about how to set a database password, see Reset the password for an ApsaraDB for MongoDB instance.

  Example:

  ```
  mongo "mongodb://root:****@s-**********.mongodb.rds.intra.env17e.shuguang.com:3717,
  s-**********.mongodb.rds.intra.env17e.shuguang.com:3717/admin"
  ```

- Mongos node connection

  Command syntax:

  ```
  mongo --host <mongos_host> -u <username> -p --authenticationDatabase <database>
  ```

  Note

  - <mongos_host>: the endpoint used to log on to a mongos node.
  - <username>: the username used to log on to a database of the instance. The initial username is **root**.
  - <database>: the name of the database corresponding to the username if authentication is enabled. If the username is root, enter admin as the database name.

  Example:

  ```
  mongo --host s-bp**********.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticati
  onDatabase admin
  ```

6.4. Apply for a public endpoint for a sharded cluster instance

This topic describes how to apply for a public endpoint for an ApsaraDB for MongoDB instance when you want to connect to this instance over the Internet.

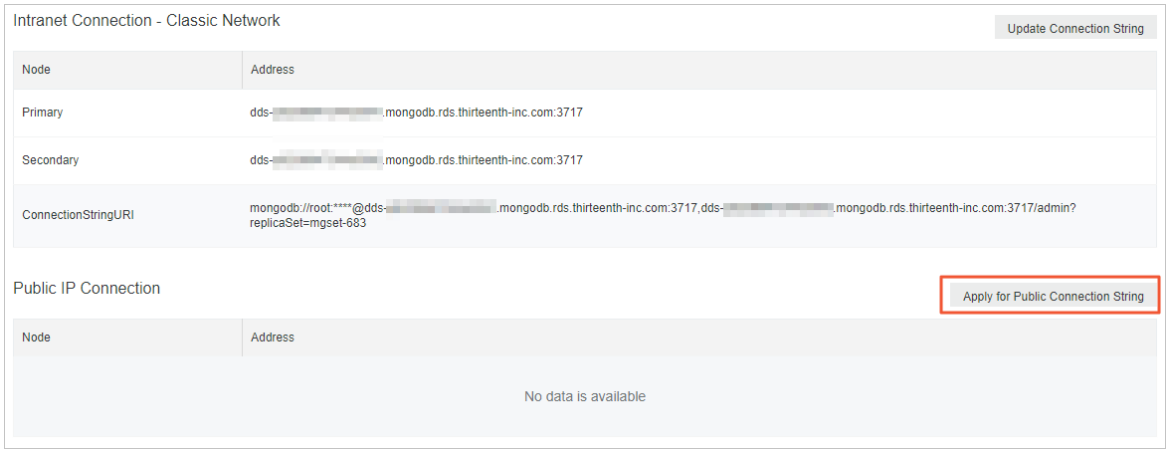
Context

The following table describes the Virtual Private Cloud (VPC) and classic network endpoints supported by ApsaraDB for MongoDB.

Connection address type	Description
VPC endpoint	• A VPC is an isolated network that provides higher security and performance than the classic network. • By default, ApsaraDB for MongoDB provides VPC endpoints for an ApsaraDB for MongoDB instance to ensure high security and high performance.
Classic network endpoint	Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by using security groups or whitelists.
Public endpoint	• Your ApsaraDB for MongoDB instance is at risk when you connect to it over the Internet. For this reason, ApsaraDB for MongoDB does not provide public endpoints by default. • If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Apsara Stack (such as an on-premises device), you must apply for a public endpoint.

Apply for a public endpoint for a replica set instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance.
3. In the left-side navigation pane, click **Database Connections**.
4. In the **Public Connections** section, click **Apply for Public Connection String**.



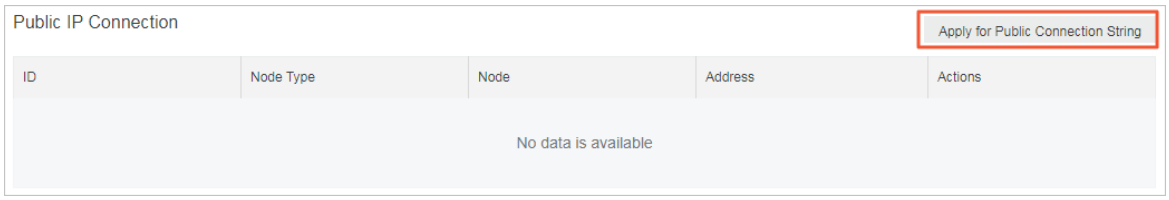
5. In the **Apply for Public Connection String** message, click **OK**.

> **Note** If you want to connect to a replica set instance by using a public endpoint, you must add the public IP address of your client to a whitelist of this instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

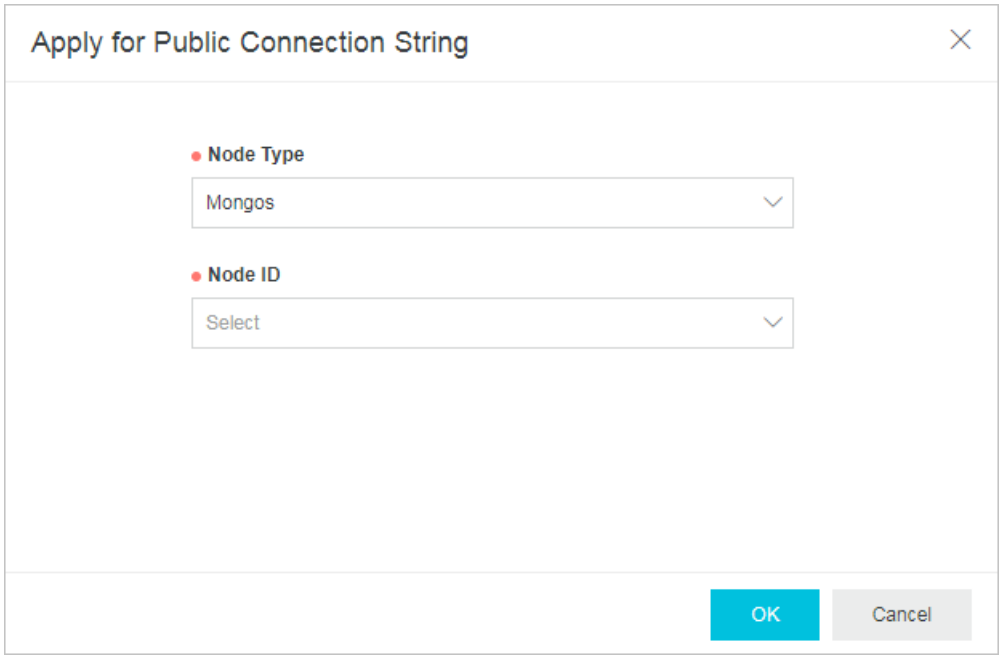
After the application is complete, the replica set instance generates new endpoints for both the primary and secondary nodes and the corresponding connection string URI. For more information, see Overview of replica set instance connections.

Apply for a public endpoint for a sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections**.
5. In the **Public Connections** section, click **Apply for Public Connection String**.



6. In the **Apply for Public Connection String** panel, select a node ID from the **Node ID** drop-down list and click **OK**.



7. (Optional)If you want to apply for public endpoints for multiple nodes in a sharded cluster instance, repeat the preceding steps.

> **Note** To apply for a public endpoint for another node in the instance, you must wait until the state of the instance becomes **Running**.

References

- To ensure data security, we recommend that you release a public endpoint if you no longer need it. For more information, see Release a public connection string.

- Before you connect to a database over the Internet, we recommend that you enable SSL encryption. For more information, see Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode.

6.5. Release a public endpoint

To ensure data security, you can release public endpoints that are no longer needed in the ApsaraDB for MongoDB console.

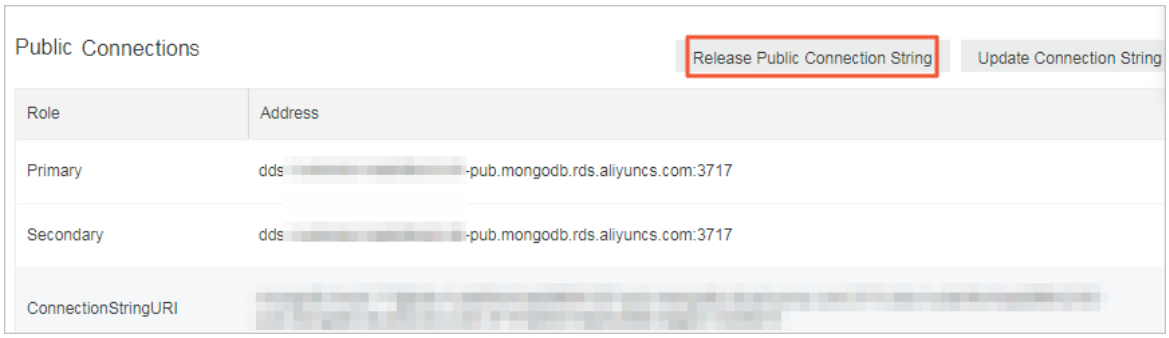
Precautions

- You can release one or more public endpoints of the mongos nodes for a sharded cluster instance.
- After a public endpoint is released for an instance or node, you cannot connect to the instance or node by using the public endpoint.
- After the public endpoint is released, we recommend that you delete the corresponding public IP address from the whitelist to ensure data security. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Release a public endpoint for a replica set instance

> **Note** After the public endpoint of a replica set instance is released, the public endpoints of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. On the **Replica Set Instances** page, click the ID of an instance.
3. In the left-side navigation pane, click **Database Connections**.
4. In the **Public Connections** section, click **Release Public Connection String**.



5. In the **Release Public Connection String** message, click **OK**.

Release a public endpoint for a sharded cluster instance

> **Note**
> - You can release one or more public endpoints of the mongos, shard, and Configserver nodes for a sharded cluster instance.
> - After the public endpoint of a shard or Configserver node is released, the public endpoints of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections**.
5. In the **Public Connections** section, find the mongos node for which you want to release the public endpoint.
6. Click **Release** in the **Actions** column.

   > **Note** You can repeat this step to release the public endpoints of other nodes as needed. To release the public endpoint of another node in the instance, you must wait until the public endpoint of the current node is released or the state of the current instance becomes **Running**.

7. In the **Release Public Connection String** message, click **OK**.
8. (Optional)You can repeat this step to release the public endpoints of multiple nodes in a sharded cluster instance.

   > **Note** To release the public endpoint of another node in the instance, you must wait until the state of the instance becomes **Running**.

6.6. Overview of replica set instance connections

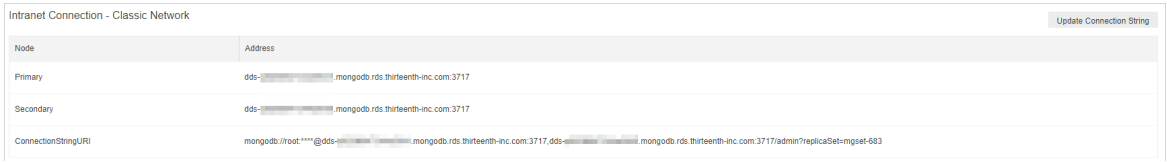
ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to the primary or secondary node, and use a connection string URI to connect to both of them. For high availability, we recommend that you use connection string URIs to connect your application to both primary and secondary nodes. This topic provides an overview of replica set instance connections.

Prerequisites

A whitelist is configured for the replica set instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click the ID of an instance.
4. In the left-side navigation pane, click **Database Connections** to view connection strings.



Description of connection strings

Item	Description
Connection address type	• Classic network endpoint: Classic network endpoints are used for communication over the classic network. In the classic network, Apsara Stack services are not isolated. To block unauthorized traffic, you must configure security groups or IP address whitelists. • VPC endpoint: Virtual private cloud (VPC) endpoints are used for communication over VPCs. A VPC is an isolated network that provides higher security and higher performance than the classic network. By default, ApsaraDB for MongoDB provides VPC endpoints for ApsaraDB for MongoDB instances to ensure high security and high performance.
Role	• Primary: the primary node in the replica set instance. If you connect to this node, you can perform read and write operations on the databases of the replica set instance. • Secondary: the secondary node in the replica set instance. If you connect to this node, you can perform only read operations on the databases of the replica set instance. • Connection String URI: ApsaraDB for MongoDB allows you to use a connection string URI to connect to a replica set instance to achieve load balancing and high availability.
Connection string	The connection string of a primary or secondary node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the replica set instance. • <port>: the port number used to connect to the replica set instance.
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix of the connection string URI. • username:password@: the username and password used to log on to the replica set instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port number used to connect to the replica set instance. • /database: the name of the database corresponding to the username if authentication is enabled. • ?options: the additional options that are used to connect to the replica set instance. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, when a node fails, the read and write operations of your application are not affected as a result of the failover.

Related information

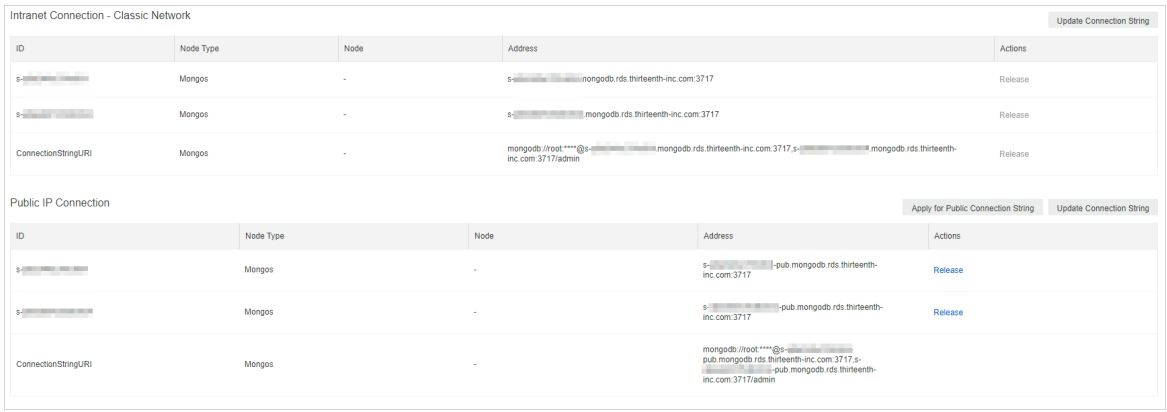
- Connect to a replica set instance by using the mongo shell

6.7. Overview of sharded cluster instance connections

ApsaraDB for MongoDB sharded cluster instances support both connection strings and connection string URIs. You can use a connection string to connect to a single mongos node and use a connection string URI to connect to multiple mongos nodes. For high availability (HA), we recommend that you use connection string URIs to connect your application to multiple mongos nodes. This topic provides an overview of sharded cluster instance connections.

View connection strings

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click the ID of the instance whose connection strings you want to view.
4. In the left-side navigation pane, click **Database Connections** to view connection strings.



Description of connection strings

Item	Description
Connection address type	• Classic network endpoint: Classic network endpoints are used for communication over the classic network. In the classic network, Alibaba Cloud services are not isolated. To block unauthorized traffic, you must configure security groups or IP address whitelists. • VPC endpoint: Virtual Private Cloud (VPC) endpoints are used for communication over VPCs. A VPC is an isolated network that provides higher security and performance than the classic network. By default, ApsaraDB for MongoDB provides VPC endpoints for ApsaraDB for MongoDB instances to ensure high security and high performance. • Public endpoint: Public endpoints are used for communication over the Internet. If you connect to an ApsaraDB for MongoDB instance over the Internet, the instance may be exposed to security risks. By default, ApsaraDB for MongoDB does not provide public endpoints for ApsaraDB for MongoDB instances. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Apsara Stack (such as an on-premises device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.

Item	Description
Mongos node ID	The connection string of a mongos node is in the following format: `<host>:<port>` • <host>: the endpoint used to connect to the instance. • <port>: the port number used to connect to the instance. **Note** During routine tests, you can use a connection string to directly connect to a mongos node.
Connection string URI	A connection string URI is in the following format: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]` • mongodb://: the prefix of a connection string URI. • username:password@: the username and password used to log on to the instance. You must separate them with a colon (:). • hostX:portX: the endpoint and port number used to connect to the instance. • /database: the name of the database corresponding to the username if authentication is enabled. • options: the additional options that are used to connect to the instance. **Note** If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. Then, your client can automatically distribute your requests to multiple mongos nodes to balance loads. If a mongos node fails, your client automatically redirects requests to other mongos nodes in the normal state.

7.Data security

7.1. Configure a whitelist for an ApsaraDB for MongoDB instance

This topic describes how to configure a whitelist for an ApsaraDB for MongoDB instance. Before you use an ApsaraDB for MongoDB instance, you must add the IP addresses or CIDR blocks that you use for database access to a whitelist of this instance. This improves database security and stability. Proper configuration of whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you maintain the whitelists on a regular basis.

Context

- The system creates a default whitelist for each instance. This whitelist can be modified or cleared but cannot be deleted.
- After an ApsaraDB for MongoDB instance is created, the system automatically adds the IP address 127.0.0.1 to the **default** whitelist of this instance. The IP address 127.0.0.1 indicates that no IP addresses are allowed to access this instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
5. On the page that appears, use one of the following methods to add IP addresses to a whitelist:
   - Manually modify a whitelist
     a. Click  in the **Actions** column corresponding to a whitelist and select **Manually Modify**.
     b. In the Manually Modify panel, enter IP addresses or CIDR blocks in the **IP White List** field.

     > **Note**
     > - Separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.
     >
     >   Supported formats are 0.0.0.0/0, IP addresses such as 10.23.12.24, or CIDR blocks such as 10.23.12.24/24. /24 indicates the length of the IP address prefix in the CIDR block. An IP address prefix can contain 1 to 32 bits.
     > - If the IP address whitelist is empty or contains only `0.0.0.0/0`, all devices are granted access. This poses risks to your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

     c. Click **OK**.

- Load internal IP addresses of ECS instances
  a. Click ⋮ in the **Actions** column corresponding to a whitelist and select **Import ECS Intranet IP**.
  b. In the **Import ECS Intranet IP** panel, select the IP addresses that you want to add to the IP address whitelist and click > to add these IP addresses to the whitelist.
  c. Click **OK**.

7.2. Create or delete a whitelist

This topic describes how to create or delete a whitelist. Whitelists consist of the IP addresses allowed to access specific databases.

Context

If your business involves multiple applications and you need to create a whitelist for each of them, you can sort the IP addresses of the applications into different whitelists.

Create a whitelist

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
5. Click **Create Whitelist**.
6. In the **Create Whitelist** panel, specify **Group Name** and **IP Whitelist** and click **OK**.

Parameter	Description
Group Name	The name of the whitelist. The name must comply with the following rules: ○ The name must start with a lowercase letter. ○ The name must end with a lowercase letter or digit. ○ The name can contain lowercase letters, digits, and underscores (_). ○ The name must be 2 to 32 characters in length.

Parameter	Description
IP Whitelist	The IP addresses or CIDR blocks that you want to add to the whitelist. **Note** ◦ Separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added. A whitelist can include IP addresses such as 0.0.0.0/0 and 10.23.12.24 or CIDR blocks such as 10.23.12.24/24. /24 indicates that the prefix of the CIDR block is 24-bit long. You can replace 24 with a value within the range of 1 to 32. ◦ If the whitelist is empty or contains `0.0.0.0/0`, all devices are granted access. This poses risks to your ApsaraDB for MongoDB instance. We recommend that you add only the IP addresses or CIDR blocks of your own web servers to the whitelist.

Delete a whitelist

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
5. Click ⋮ in the **Actions** column corresponding to the whitelist that you want to delete and select **Delete Whitelist Group**.

   Note You cannot delete the **default** whitelist.
6. In the **Delete Whitelist Group** message, click **OK**.

7.3. Audit logs

This topic describes audit logs provided in the ApsaraDB for MongoDB console. You can query the statement execution logs, operation logs, and error logs of an ApsaraDB for MongoDB instance to identify and analyze faults.

Context

The audit log feature records all operations that a client performs on a connected database. This feature provides references for you to perform fault analysis, behavior analysis, and security auditing because you can obtain the operation execution details from the audit logs. Audit logs are essential in the regulatory operations of Finance Cloud and other core business scenarios.

Note Audit logs are stored for seven days, after which they are deleted.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > Audit Logs**.
5. In the upper-left corner of the page, click **Enable Audit Log**.
6. In the **Enable Audit** message, click **OK**.

Result

After the audit log feature is enabled, specify the time range, database name, database user, and keyword to query audit logs. You can also use the following options:

- **Export File**: exports an audit log file.
- **File List**: displays a list of audit log files.
- **Disable Audit Log**: stops the collection of information on database operations and deletes the saved audit logs.

7.4. Configure SSL encryption for an ApsaraDB for MongoDB instance

To enhance link security, you can enable SSL encryption and install SSL certification authority (CA) certificates on your application services. SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity. This topic describes operations related to SSL encryption.

Prerequisites

- The instance is a replica set instance.
- The MongoDB version of the instance is 3.4, 4.0, or 4.2.

Notes

When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your applications can automatically re-establish a connection.

> **Note** When an instance is restarted, all its nodes are restarted in turn and each node goes through a transient connection of about 30 seconds. If the instance contains more than 10,000 collections, the transient connections last longer.

Precautions

- You can download SSL CA certificate files only from the ApsaraDB for MongoDB console.
- After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when encryption needs arise. For example, you can enable SSL encryption when you connect to an ApsaraDB for MongoDB instance

over the Internet.

> **Note** Internal network connections are more secure than Internet connections and do not need SSL encryption.

- After you enable SSL encryption for an instance, both SSL and non-SSL connections are supported.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > SSL**.
5. Perform the corresponding operations based on your needs.

> **Note** When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your applications can automatically re-establish a connection.

Operation	Prerequisite	Procedure
Enable SSL encryption	The SSL encryption state is **Disabled**.	i. Turn on **SSL Status**. ii. In the **Restart Instance** message, click **OK**.
Update an SSL CA certificate	The SSL encryption state is **Enabled**.	i. Click **Update Certificate**. ii. In the **Restart Instance** message, click **OK**.
Download an SSL CA certificate file	The SSL encryption state is **Enabled**.	Click **Download Certificate** to download an SSL CA certificate file to your computer.
Disable SSL encryption	The SSL encryption state is **Enabled**.	i. Turn off **SSL Status**. ii. In the **Restart Instance** message, click **OK**.

7.5. Configure TDE for an ApsaraDB for MongoDB instance

This topic describes how to configure Transparent Data Encryption (TDE) for an ApsaraDB for MongoDB instance. Before data files are written to disks, TDE encrypts the data files. When data files are loaded from disks to the memory, TDE decrypts the data files. TDE does not increase the sizes of data files. When you use TDE, you do not need to modify your application that uses the ApsaraDB for MongoDB instance. To enhance data security, you can enable the TDE feature for an instance in the ApsaraDB for MongoDB console.

Prerequisites

The MongoDB version of the instance is 4.0 or 4.2.

> **Note** Before you enable TDE, you can create a MongoDB 4.0 or 4.2 instance to test the compatibility between your application and the database version. You can release the instance after the test is complete.

Notes

- When you enable TDE, your instance is restarted, and your application is disconnected from the instance. We recommend that you enable TDE during off-peak hours and make sure that your application can reconnect to the instance after it is disconnected.
- TDE increases the CPU utilization of your instance.

Precautions

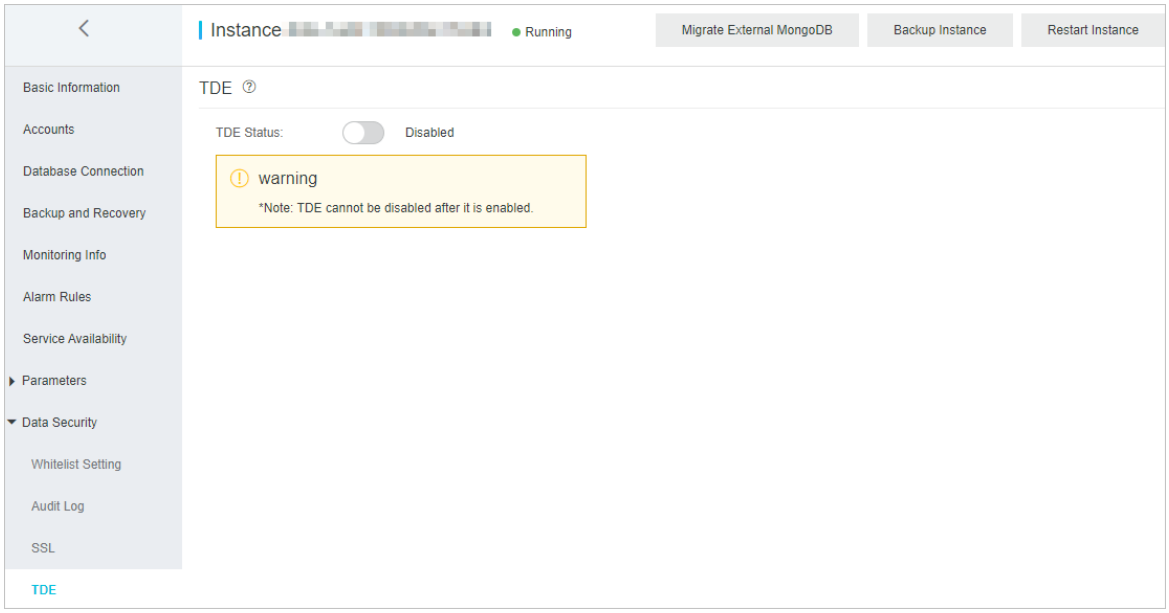
- You cannot disable TDE after it is enabled.
- You can enable TDE for an instance and disable encryption for a collection.

  > **Note** In special business scenarios, you can choose not to encrypt a collection when you create it. For more information, see Disable encryption for a specified collection.

- After you enable TDE, only new collections are encrypted. Existing collections are not encrypted.
- Key Management Service (KMS) generates and manages the keys used by TDE. ApsaraDB for MongoDB does not provide keys or certificates required for encryption.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **Data Security > TDE**.
5. On the **TDE** page, turn on **TDE Status**.



6. In the **Restart Instance** message, click **OK**.
   The instance state changes to **Modifying TDE**. After the state changes to **Running**, TDE is enabled.

Disable encryption for a specified collection

After you enable TDE, all new collections are encrypted. When you create a collection, you can perform the following steps to disable encryption for the collection:

1. Connect to a replica set instance by using the mongo shell. For more information, see Connect to a replica set instance by using the mongo shell.
2. Run the following command to create a collection with encryption disabled:

```
db.createCollection("<collection_name>",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)" } } })
```

> **Note** <collection_name>: the name of the collection.

Example:

```
db.createCollection("customer",{ storageEngine: { wiredTiger: { configString: "encryption=(name=none)" } } })
```

7.6. Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode

This topic describes how to use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode. SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity.

Prerequisites

- The instance is a replica set instance, and the database version of the instance is 3.4, 4.0, or 4.2.
- The mongo shell of the required version is installed on the on-premises server or Elastic Compute Service (ECS) instance from which you want to connect to the database. For more information about the installation procedure, see Install MongoDB.
- SSL encryption is enabled for the instance. For more information, see Configure SSL encryption.
- The IP address of the on-premises server or ECS instance is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Precautions

After you enable SSL encryption for an instance, the CPU utilization of the instance is significantly increased. We recommend that you enable SSL encryption only when encryption needs arise.

Procedure

An on-premises server with a Linux operating system is used in the following example.

1. Download an SSL CA certificate package. For more information, see Configure SSL encryption.
2. Decompress the package and upload the certificate files to the on-premises server or ECS instance where the mongo shell is installed.

   > **Note** In this example, the *.pem* file is uploaded to the */root/sslcafile/* directory of the on-premises server.

3. On the on-premises server or ECS instance that has the mongo shell installed, run the following command to connect to a database of the ApsaraDB for MongoDB instance:

   ```
   mongo --host <host> -u <username> -p --authenticationDatabase <database> --ssl --sslCAF
   ile <sslCAFile_path> --sslAllowInvalidHostnames
   ```

   > **Note**
   > - If you want to connect to a database of the ApsaraDB for MongoDB instance over an internal network, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instance. If the network type is VPC, make sure that the two instances reside within the same virtual private cloud (VPC).
   > - <host>: the endpoint of the primary or secondary node for a replica set instance or of the mongos node for a sharded cluster instance. For more information, see Overview of replica set instance connections or Overview of sharded cluster instance connections.
   > - <username>: the username you use to log on to a database of the ApsaraDB for MongoDB instance. The initial username is root.
   > - <database>: the name of the database corresponding to the username if authentication is enabled. If the username is root, enter admin as the database name.
   > - <sslCAFile_path>: the path of the SSL CA certificate files.

   Example:

```
mongo --host dds-bpxxxxxxxx-pub.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticati
onDatabase admin --ssl --sslCAFile /root/sslcafile/ApsaraDB-CA-Chain.pem  --sslAllowInv
alidHostnames
```

4. When `Enter password:` is displayed, enter the password of the database user and press the Enter key.

   Note

   - The password characters are not displayed when you enter the password.
   - If you forget the password of the root user, you can reset the password. For more information, see Reset the password for an ApsaraDB for MongoDB instance.

8.Zone-disaster recovery

8.1. Create a dual-zone replica set instance

This topic describes how to create a dual-zone replica set instance. ApsaraDB for MongoDB provides a zone-disaster recovery solution to ensure the reliability and availability of your replica set instance. This solution deploys the three nodes of a replica set instance across two different zones within one region. The components in these zones exchange data over an internal network. When one of the two zones becomes unavailable due to unexpected events such as a power or network failure, the high-availability (HA) system switches services over to another zone.

Deployment policies

The primary, secondary, and hidden nodes of a replica set instance are deployed in two different zones within one region.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances**.
3. On the **Replica Set Instances** page, click **Create Instance**.
4. On the **Create ApsaraDB for MongoDB Instance** page, configure parameters.

   > Note
   > - For the **Zone** parameter, you must select dual zones, such as the amtest17001-a and amtest17001-b zones of the cn-qingdao-env11e-MAZ1 region.
   > - For more information, see Create a replica set instance.

5. Click **Submit**.

8.2. Create a dual-zone sharded cluster instance

This topic describes how to create a dual-zone sharded cluster instance. ApsaraDB for MongoDB provides a zone-disaster recovery solution to ensure the high availability (HA) of your sharded cluster instances. This solution deploys the components of a sharded cluster instance across two different zones within the same region. The components in these zones exchange data over an internal network. If one zone becomes unavailable due to uncontrollable circumstances, the HA system automatically switches business over to the other zone to ensure service continuity of the sharded cluster instance.

Deployment policies

The components of a sharded cluster instance are deployed across two different zones within one region.

- Mongos nodes are evenly deployed across all data centers. At least two mongos nodes are deployed at a time, with each in one zone. Each new mongos node added later is deployed to one of the zones in turn.
- The primary, secondary, and hidden nodes in each shard node are not deployed to the two zones in sequence. The deployment of these nodes may change when manual switchover or HA failover between primary and secondary nodes is triggered.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Sharded Cluster Instances**.
3. On the **Sharded Cluster Instances** page, click **Create Instance**.
4. On the **Create ApsaraDB for MongoDB Sharded Cluster Instance** page, configure the required parameters.

   > **Note**
   > - For the **Zone** parameter, you must select two zones, such as the amtest17001-a and amtest17001-b zones of the cn-qingdao-env11e-MAZ1 region.
   > - For more information, see Create a sharded cluster instance.

5. Click **OK**.

9.CloudDBA

9.1. Performance trends

This topic describes how to view performance trends in specific ranges, compare performance trends, and customize charts to view performance trends on an ApsaraDB for MongoDB instance.

Go to the Performance page

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **CloudDBA > Performance Trends**.

> **Note** For more information about performance trends, see *Performance trends* in *Database Autonomy Service User Guide*.

9.2. Real-time monitoring

This topic describes how to view real-time performance data of an ApsaraDB for MongoDB instance, such as read/write latency, queries per second (QPS), operations, connections, and network traffic.

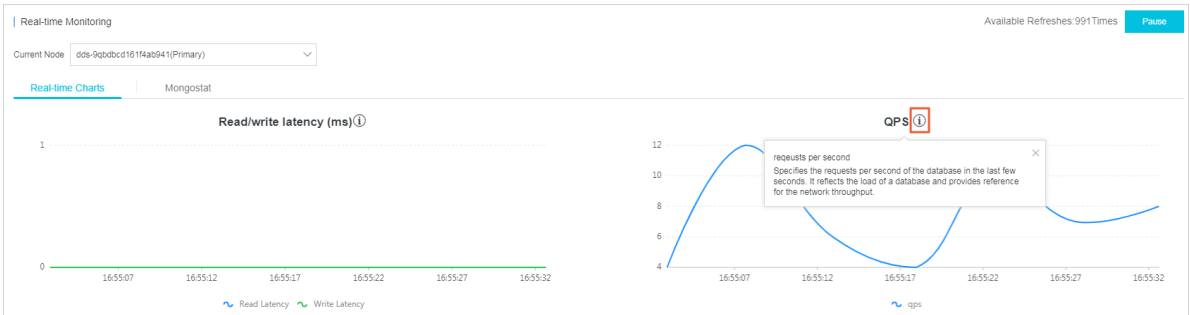
Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **CloudDBA > Real-time Performance**.

Page overview

On the Real-time Performance page, you can click the Real-time Charts or Mongostat tab to view performance data. When you refresh the **Real-time Performance** page, the data on the Real-time Charts and Mongostat tabs is refreshed and **Available Refreshes** is reset in the upper-right corner.

- Real-time charts



By default, content on the **Real-time Charts** tab is displayed when you go to the Real-time Performance page. Line charts on the tab are refreshed every 5 seconds to provide up-to-date performance trends.

> **Note** You can move the pointer over ⓘ to view the detailed information of performance parameters.

- Mongostat

Click the **Mongostat** tab. On the tab, you can view Mongostat command outputs. A new line of real-time performance data is added every 5 seconds. The tab can contain up to 999 lines of data.

> **Note** For more information about Mongostat command outputs, see mongostat.

9.3. Session management

The instance sessions feature allows you to view the statistics for sessions between an ApsaraDB for Redis instance and clients in real time. These statistics include clients, commands that were run, and connection duration. You can also close abnormal sessions based on your business requirements.

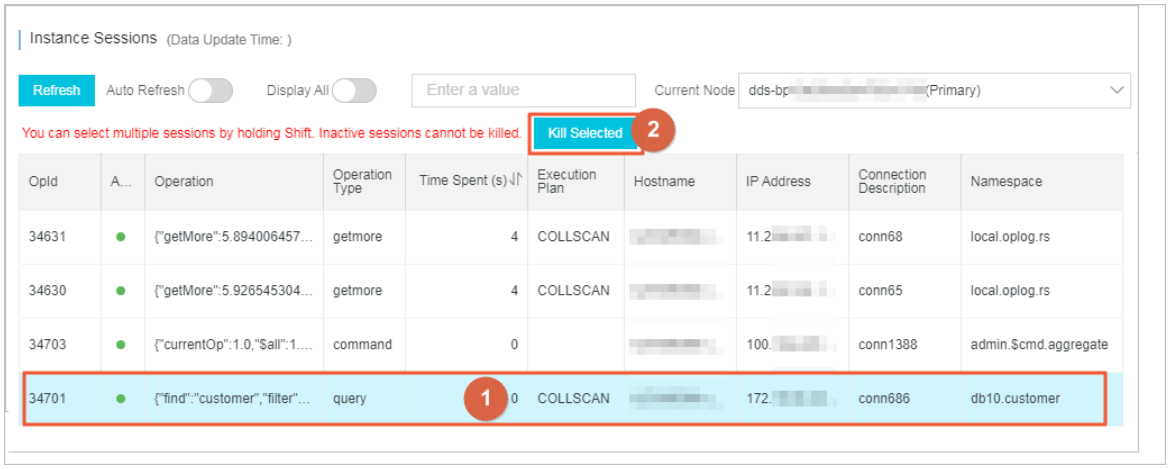
View instance sessions

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **CloudDBA > Sessions**.
   - If you turn on **Auto Refresh**, the system updates session data on the page every 5 seconds.
   - By default, the system displays only active sessions. You can turn on **Display All** to view all sessions.
   - In the **Session Statistics** section, you can view information about sessions in the **Overview**, **Statistics by Client**, and **Statistics by Namespace** charts.

Terminate instance sessions

Warning To prevent unexpected results, we recommend that you do not terminate system-level sessions.

1. In the **Instance Sessions** section, select the session that you want to terminate.



2. Click **Kill Selected**.
3. In the message that appears, click **OK**.

9.4. Storage analysis

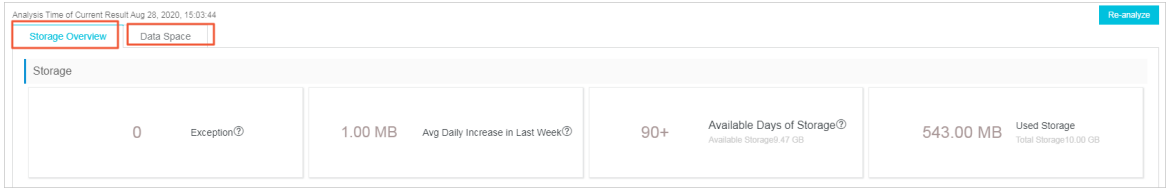
This topic describes how to view storage analysis information, such as the information in the **Storage Overview**, **Exceptions**, **Storage Trend**, **Tablespaces**, and **Data Space** sections. The information helps you identify and resolve exceptions in the database storage to ensure database stability.

Prerequisites

The database version of the ApsaraDB for MongoDB instance is 4.0 or 4.2.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **CloudDBA > Storage Analysis**.
5. In the upper-right corner, click **Re-analyze**. Then, wait until the analysis is complete.
6. Click the **Storage Overview** or **Data Space** tab to view analysis results.



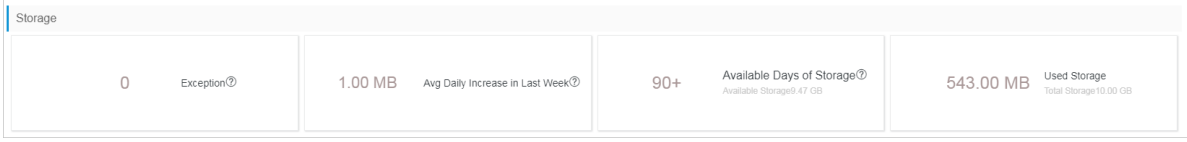
- For more information about the Storage Overview tab, see Storage Overview tab.

- For more information about the Data Space tab, see Data Space tab.

Storage Overview tab

This tab contains four sections: **Storage**, **Exceptions**, **Storage Trend**, and **Tablespaces**.

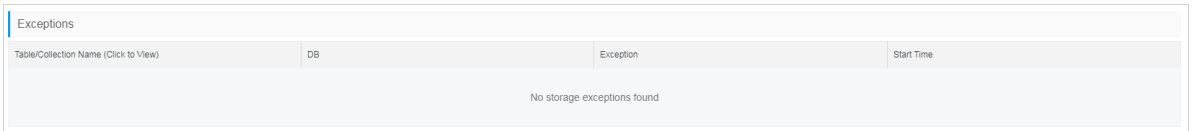
- **Storage** section



Item	Description
Exception	The number of storage exceptions that are identified in the instance. ApsaraDB for MongoDB can identify the following types of exceptions: ○ More than 90% of the storage capacity is used. ○ The available physical storage space may be exhausted within seven days. ○ A single collection contains more than 10 indexes.
Avg Daily Increase in Last Week	The average daily increase of storage usage in the instance over the last seven days. Formula: (Size of available storage space at the time of collection - Size of available storage space seven days ago)/7. **Note** ○ This increase indicates the average daily increase over the last seven days at the time of collection. ○ This parameter is suitable for scenarios in which the traffic volume remains stable. The value of this parameter is inaccurate in the event of abrupt storage changes that are caused by operations such as batch import, deletion of historical data, instance migration, or instance recreation.
Available Days of Storage	The number of days during which storage space is available in the instance. Formula: Size of available storage space/Average daily increase over the last seven days. **Note** ○ 90+ indicates that the storage space is sufficient for more than 90 days of usage. ○ This parameter is suitable for scenarios in which the traffic volume remains stable. The value of this parameter is inaccurate in the event of abrupt storage changes that are caused by operations such as batch import, deletion of historical data, instance migration, or instance recreation.
Used Storage	The amount of storage space that is used in the instance to the total storage space.

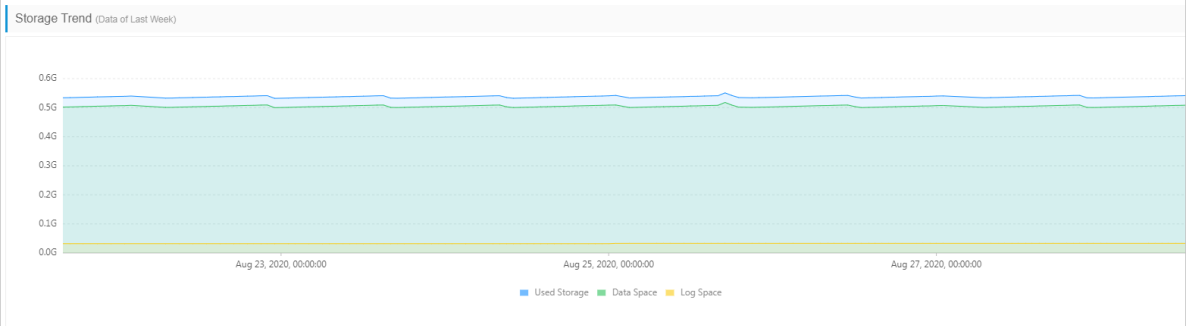
- **Exceptions** section

  Information about detected storage exceptions. You can resolve the exceptions based on the information in this section.



- **Storage Trend** section

  Changes in storage usage over the last week, such as changes in **used storage**, **data space**, and **log space**.



- **Tablespaces** section

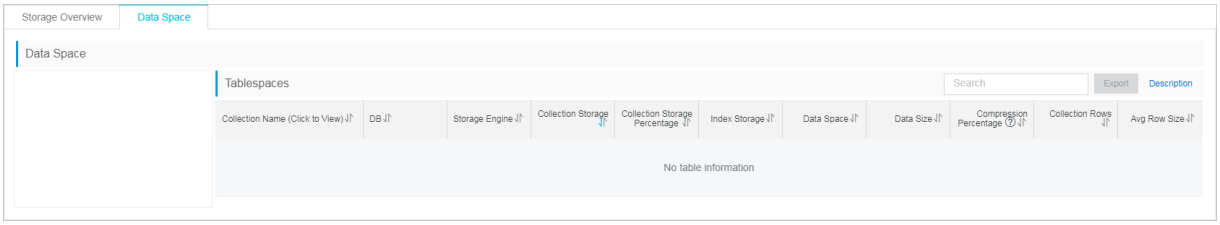
  Information about all tables, such as the database name, storage engine, and collection storage.

> **Note** You can click the name of a collection to view its indexes.

Data Space tab

The Data Space tab shows the total storage space and tablespace information of each database.

> **Note**
>
> - You can click the name of a data space to view its tablespace information.
> - You can click the name of a collection to view its indexes.



9.5. Slow query logs

This topic describes how to view slow query logs of an ApsaraDB for MongoDB instance. You can identify, analyze, diagnose, and track slow query logs to create indexes, which improves the utilization of resources in the instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console. For more information about how to log on to the console, see Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**.
3. On the **Replica Set Instances** or **Sharded Cluster Instances** page, click the ID of an instance.
4. In the left-side navigation pane, choose **CloudDBA > Slow Query Logs**.

   > **Note** By default, slow query logs generated in the past 15 minutes are displayed in the trend chart. You can specify a time range and click **Search** to view slow query logs. The maximum time range is one day.

5. View details of slow query logs by using one of the following methods:

   Method 1

   i. Click the **Slow Log Details** tab in the lower part of the page.

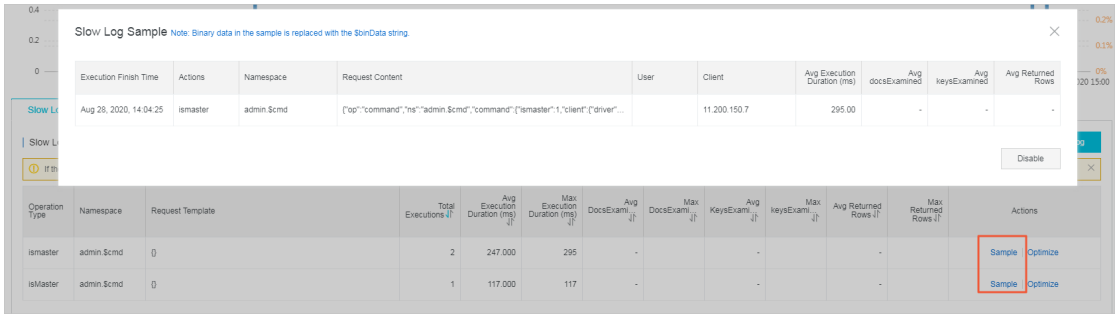
   ii. On the **Slow Log Details** tab, select the database that you want to query.

   > **Note** If the request content of the database is hidden, you can move the pointer over the corresponding content in the **Request Content** column and view the complete content.

   Method 2

   i. In the Slow Log Trend chart, click a point in time. Then, you can view the statistics of the slow query logs generated at the point in time on the **Slow Log Statistics** tab.

ii. On the **Slow Log Statistics** tab, click **Sample** in the **Actions** column.
In the **Slow Log Sample** dialog box, you can view details of the slow query log.



> **Note** If the request content of the database is hidden, you can move the pointer over the corresponding content in the **Request Content** column and view the complete content.

Export slow query logs

You can click **Export Slow Log** on the **Slow Log Statistics** tab to save the slow query log information to your computer.