ALIBABA CLOUD

Alibaba Cloud

ApsaraDB for MongoDB
User Guide

Document Version: 20210205

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.Preface

This document describes how to use the ApsaraDB for MongoDB console to help you manage ApsaraDB for MongoDB instances and learn about the features of ApsaraDB for MongoDB.

ApsaraDB for MongoDB is a stable, reliable, and scalable database service. Many features can be extended on ApsaraDB for MongoDB, such as secondary index, range query, sorting, aggregation, and geospatial index. ApsaraDB for MongoDB is fully complies with the MongoDB protocols and provides a full range of database solutions, such as disaster recovery, data backup, data recovery, monitoring, and alerts.

Why ApsaraDB for MongoDB?

For information about benefits of ApsaraDB for MongoDB, see Comparison between ApsaraDB for MongoDB and user-created databases and Scenarios.

Overview

To contact technical support personnel, you can submit a ticket.

For more information about the features and pricing of ApsaraDB for MongoDB, see the buy page of ApsaraDB for MongoDB.

Disclaimer

Some product features or services described in this document may be unavailable in certain regions. See the actual commercial contracts for specific Terms and Conditions. This document serves as a reference guide for your use of ApsaraDB for MongoDB. No content in this document shall be deemed as explicit or implicit guarantees. The information in this document is subject to change without notice. You must first verify the document with your software version.

2.Quick start

If you use ApsaraDB for MongoDB for the first time, you can read Alibaba Cloud ApsaraDB for MongoDB quick start guides, which can help you understand ApsaraDB for MongoDB and quickly migrate data from a user-created database to an ApsaraDB for MongoDB instance.

Get started

3.Billing management

3.1. Change the billing method of an ApsaraDB for MongoDB instance from pay-as-you-go to subscription

This topic describes how to change the billing method of an ApsaraDB for MongoDB instance from pay-as-you-go to subscription. Changes to the billing method do not impact on the running of the instance.

Prerequisites

- The instance is in the running state.
- The billing method of the instance is pay-as-you-go.
- The instance has no unpaid subscription orders.
- The instance type is available for purchase. For more information about unavailable instance types, see Historical instance types. If you want to change the billing method of an instance whose instance type is unavailable now to subscription, change the instance type first. For more information, see Configuration change overview.

Precautions

- The billing method of a subscription instance cannot be changed to pay-as-you-go. Exercise caution when changing the billing method of your instance.
- You cannot release a subscription instance.
- If the instance has an unpaid order, you cannot upgrade the specifications of a subscription instance. You need to cancel this order on the Billing Management page and change the billing method of the instance to subscription.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the **Basic Information** section, click **Switch to Subscription**.
6. On the **Confirm Order** page, specify **Purchase Cycle** of the instance.
7. Read and select **ApsaraDB for MongoDB Agreement of Service** and click **Activate**.

   > **Note** You must complete the generated subscription order. You cannot purchase a new instance or change the billing method to subscription until you pay for this order or cancel it. You can pay for or cancel this order on the Billing Management page.

8. Select a payment method and click **confirm to pay**.

3.2. Manually renew an ApsaraDB for MongoDB subscription instance

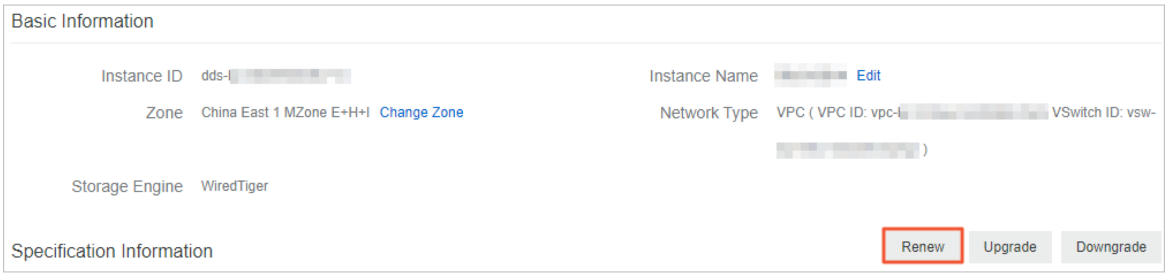
This topic describes how to manually renew an ApsaraDB for MongoDB subscription instance. We recommend that you manually renew your subscription instance before it expires, to prevent service interruptions or data loss.

Context

When a subscription instance expires, you need to renew it within seven days. After the seven-day grace period, the instance is released and its data is permanently deleted. For more information about renewal rules and billing instructions, see Billing items and pricing.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the **Basic Information** section, click **Renew**.



Basic Information

Instance ID dds-[redacted]

Instance Name [redacted] Edit

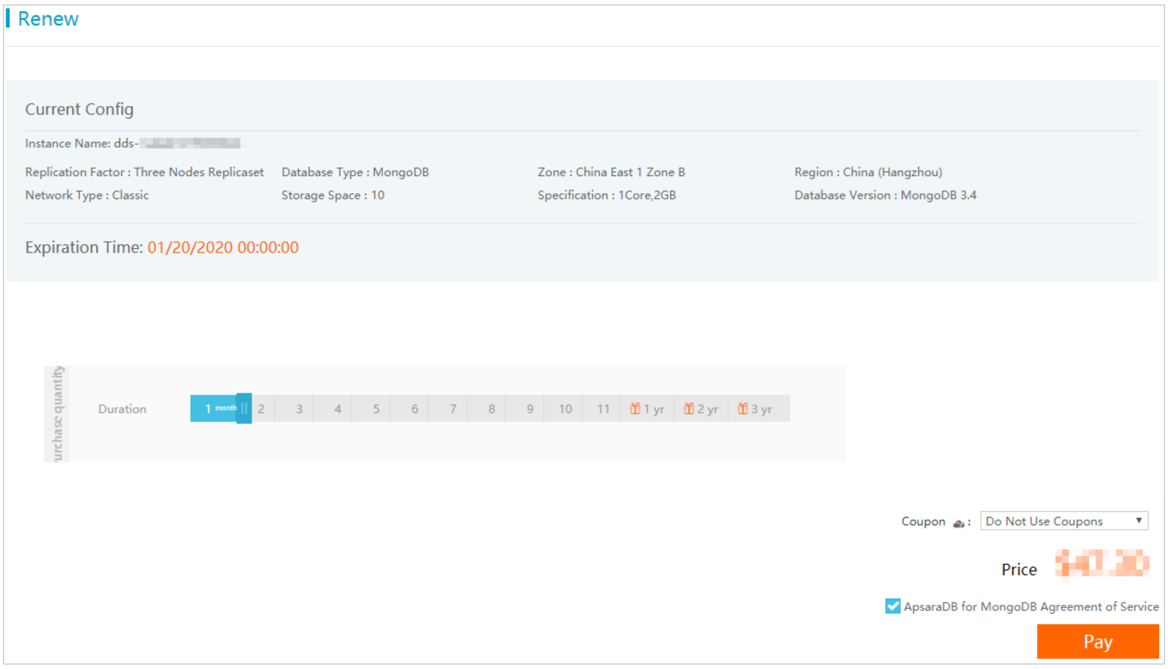
Zone China East 1 MZone E+H+I Change Zone

Network Type VPC (VPC ID: vpc-[redacted] VSwitch ID: vsw-[redacted])

Storage Engine WiredTiger

Specification Information

6. Specify **Duration**.



Renew

Current Config

Instance Name: dds-[redacted]

Replication Factor : Three Nodes Replicaset
Database Type : MongoDB
Zone : China East 1 Zone B
Region : China (Hangzhou)
Network Type : Classic
Storage Space : 10
Specification : 1Core,2GB
Database Version : MongoDB 3.4

Expiration Time: 01/20/2020 00:00:00

Duration: 1 month

Coupon: Do Not Use Coupons

Price [redacted]

☑ ApsaraDB for MongoDB Agreement of Service

> **Note** You can also enable auto-renewal for your ApsaraDB for MongoDB subscription instance. This prevents service interruptions due to overdue payments. For more information, see Enable and disable auto-renewal for an ApsaraDB for MongoDB subscription instance.

7. Select Agreement of Service and click **Pay**. Complete the payment as instructed.

3.3. Enable and disable auto-renewal for an ApsaraDB for MongoDB subscription instance

This topic describes how to enable and disable auto-renewal for an ApsaraDB for MongoDB subscription instance. Auto-renewal relieves you from the tedious work of regularly renewing your ApsaraDB for MongoDB subscription instance and helps ensure service continuity. You can also disable auto-renewal if needed.

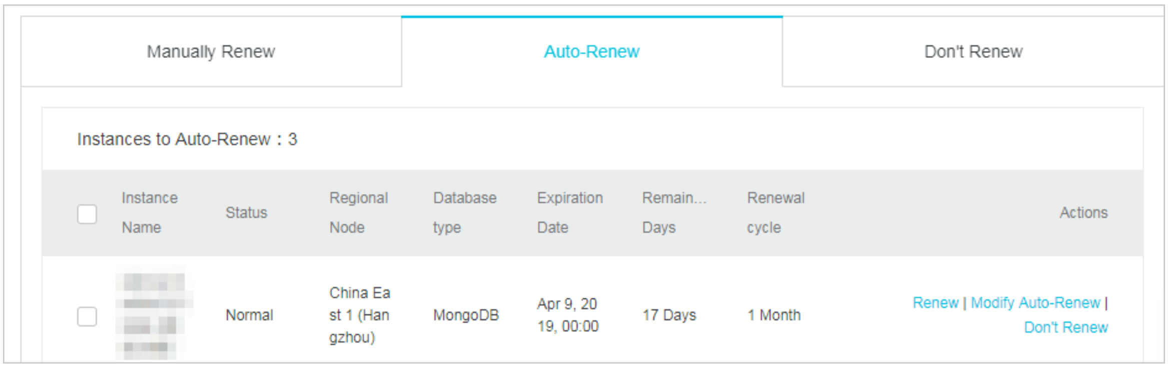
Context

You can enable auto-renewal when purchasing an ApsaraDB for MongoDB instance. You also have the option to enable auto-renewal in the ApsaraDB for MongoDB console after the instance is created. The system automatically renews your instance based on the selected renewal cycle. For example, if you select a three-month renewal cycle, you are charged for a three-month subscription each renewal cycle.

> **Note** When purchasing a subscription instance, you can select **Auto Renew** next to **Duration**.
>
> - Subscription on a monthly basis: The auto-renewal cycle is a month.
> - Subscription on a yearly basis: The auto-renewal cycle is a year.

Procedure

1.
2. In the top navigation bar, choose **Billing Management** > **Renew** to go to the **Renew** center.
3. In the left-side navigation pane, click **ApsaraDB for MongoDB** to go to the renewal page of **ApsaraDB for MongoDB**.
4. Click the **Auto-Renew** tab.

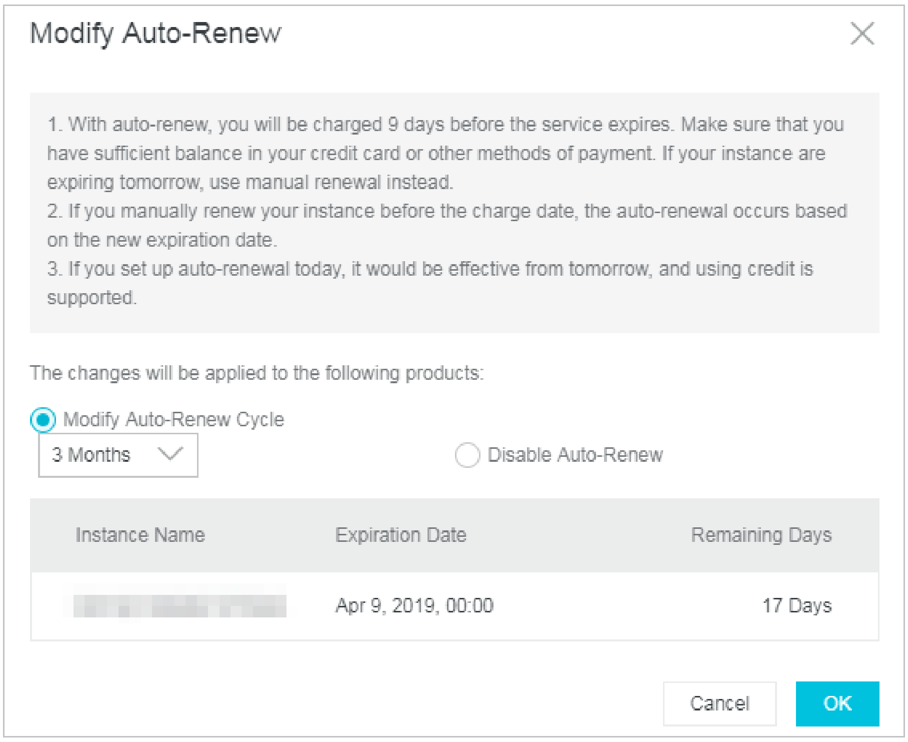


Instances to Auto-Renew : 3

□	Instance Name	Status	Regional Node	Database type	Expiration Date	Remain... Days	Renewal cycle	Actions
□	[redacted]	Normal	China East 1 (Hangzhou)	MongoDB	Apr 9, 2019, 00:00	17 Days	1 Month	Renew | Modify Auto-Renew | Don't Renew

> **Note**
> - You can click **Renew** in the Actions column corresponding to an instance. In the **Renew** dialog box that appears, renew the instance.
> - You can click **Don't Renew** in the Actions column corresponding to an instance. In the **Don't Renew** dialog box that appears, disable auto-renewal.

5. Find the target instance and click **Modify Auto-Renew** in the Actions column. The **Modify Auto-Renew** dialog box appears.



Modify Auto-Renew

1. With auto-renew, you will be charged 9 days before the service expires. Make sure that you have sufficient balance in your credit card or other methods of payment. If your instance are expiring tomorrow, use manual renewal instead.
2. If you manually renew your instance before the charge date, the auto-renewal occurs based on the new expiration date.
3. If you set up auto-renewal today, it would be effective from tomorrow, and using credit is supported.

The changes will be applied to the following products:

● Modify Auto-Renew Cycle
3 Months

○ Disable Auto-Renew

Instance Name	Expiration Date	Remaining Days
[redacted]	Apr 9, 2019, 00:00	17 Days

6. Select an auto-renewal cycle and click **OK**.

4.Instance connection

4.1. Connect to an ApsaraDB for MongoDB instance over the internal network across zones

Alibaba Cloud provides two internal network types: the classic network and VPCs. An ECS instance and an ApsaraDB for MongoDB instance in different zones of the same region can be interconnected over the internal network.

This topic describes two scenarios.

Connect an ECS instance to a new ApsaraDB for MongoDB instance

- If the network type of the ECS instance is VPC and you purchase an ApsaraDB for MongoDB instance in a different zone of the same region, you must ensure that the two instances have the same VPC ID. In addition, you must create a VSwitch in the zone where the ApsaraDB for MongoDB instance is deployed. Then the two instances can be interconnected properly over the internal network.
- If the network type of the ECS instance is classic network and you purchase an ApsaraDB for MongoDB instance in a different zone of the same region, the two instances can be interconnected properly over the internal network because they are in the same region.

Connect an ECS instance to an existing ApsaraDB for MongoDB instance

The ECS instance and ApsaraDB for MongoDB instance must be in the same region.

- If both the ECS instance and ApsaraDB for MongoDB instance have the same network type (), they can be interconnected properly over the internal network.

  > **Note** The network type of both instances is classic network or VPC. If the network type is VPC, both instances must have the same VPC ID.

- If the two instances have different network types, you can change the network type of the ApsaraDB for MongoDB instance to be the same as that of the ECS instance before connecting them. For more information, see Switch the network type of an ApsaraDB for MongoDB instance.

  > **Note** You cannot switch the network type for standalone instances.

4.2. Manage ApsaraDB for MongoDB instances by using DMS

Data Management (DMS) can manage relational databases such as MySQL, SQL Server, PostgreSQL, Oracle, and OceanBase, as well as NoSQL databases such as MongoDB. DMS is an integrated data management service that offers data management, schema management, R&D, user management, permission management, and access control. You can use DMS to connect to a standalone ApsaraDB for MongoDB instance for easy management.

Context

DMS provides the following roles:

- DMS administrator: DMS administrators can use all system management features except for data protection.

  Note Only DMS administrators can manage users and the IP address whitelist.

  If you assume the role of DMS administrator, you can register your ApsaraDB for MongoDB instance to DMS for management. For more information, see the Register an ApsaraDB for MongoDB instance to DMS section.

- Security administrator: Security administrators can manage operations logs and use the data protection feature.

  Note Only security administrators can use the data protection feature.

- Database administrator (DBA): DBAs can manage instances, tasks, security rules, and configurations, and design the schema.
- Common user: Common users cannot use the system management features.

  If you assume the role of common user, you must first obtain the permission to manage your ApsaraDB for MongoDB instance. For more information, see the Apply for permissions section.

Preparations

Add the IP addresses of DMS servers to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for a standalone ApsaraDB for MongoDB instance.

Note Skip this step if you have added the IP addresses of DMS servers to the whitelist of the ApsaraDB for MongoDB instance.

IP addresses of the DMS server

Network type of ApsaraDB for MongoDB instance	IP address of DMS server
VPC	100.104.0.0/16
Classic network	120.55.177.0/24 121.43.18.0/24 101.37.74.0/24 10.153.176.0/24 10.137.42.0/24 11.193.54.0/24

Register an ApsaraDB for MongoDB instance to DMS

The following example shows how to register an ApsaraDB for MongoDB instance to DMS.

Note The ApsaraDB for MongoDB instance must be registered by a DMS administrator.

1. Log on to the Data Management console.
2. In the upper-left corner of the page, choose **Add instance/Batch entry > Add instance**. The **Add instance** dialog box appears.
3. On the **Cloud** tab of the Data source step, click **MongoDB**.
4. In the **Basic Information/Advanced information** step, configure the parameters. The following table describes the parameters.

Section	Parameter	Description
Basic Information	**Data source**	The source of the database instance to add. The default value is **Cloud**.
	Database type	The cagtegory of the instance to be registered. In this example, select **MongoDB** from the drop-down list.
	Instance Area	The region where the database instance is deployed.
	Entry mode	The method that you can use to log on to the database instance. You can set this parameter only to **Connection string mode**.
	Connection string address	The endpoint of the database instance. For more information about how to view the endpoint, see Connect to an ApsaraDB for MongoDB instance.
	Database Name	The name of the database. If the account is root, the database name is admin.
	Database account	The username that you can use to log on to the database.
	Database password	The password that you can use to log on to the database.
	Control Mode	The control mode that is used to manage the instance in DMS. For more information, see Control modes.

Section	Parameter	Description
Advanced information	**Environment type**	The environment of the database instance.
	Instance Name	The name of the database instance.
	DBA	The database administrator (DBA) of the database instance. The DBA can grant permissions to users.
	query timeout(s)	The timeout period for the execution of an SQL query statement. If the execution of an SQL query statement lasts longer than the specified timeout period, the execution of the statement is terminated to protect the database.
	export timeout(s)	The timeout interval of the statement that is used to export statistics. When the specified time interval is reached, the target statement executed in the SQL editor is stopped to protect the database security.

5. In the Basic Information section, click **Test connection** in the lower-left corner. Wait until the connectivity test is successful.

> ? **Note** If the test fails, check the parameter values that you specified.

6. Click **Submit**.

Apply for permissions

For more information, see the **Apply for permissions** topic in Permission management.

4.3. Connect to an ApsaraDB for MongoDB instance

This topic describes how to connect to an ApsaraDB for MongoDB instance.

Connection methods

Category	Connection method

Category	Connection method
Standalone instance	• Connect to a standalone ApsaraDB for MongoDB instance by using DMS • Connect to a standalone ApsaraDB for MongoDB instance by using the mongo shell • Connection sample code for MongoDB drivers
Replica set instance	• Connect to a replica set ApsaraDB for MongoDB instance by using DMS • Connect to a replica set instance by using the mongo shell • Connection sample code for MongoDB drivers
Sharded cluster instance	• Connect to a sharded cluster ApsaraDB for MongoDB instance by using DMS • Connect to a sharded cluster instance by using the mongo shell • Connection sample code for MongoDB drivers

Common connection scenarios

- Connect to an ApsaraDB for MongoDB instance over the Internet
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions
- Connect an ECS instance with an ApsaraDB for MongoDB instance in another Alibaba Cloud account

FAQ

- How to troubleshoot logon issues for the mongo shell
- How to troubleshoot database connection failures after the number of connections reaches the upper limit
- Troubleshoot the high CPU usage of ApsaraDB for MongoDB
- How to query and limit the number of connections

4.4. How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different

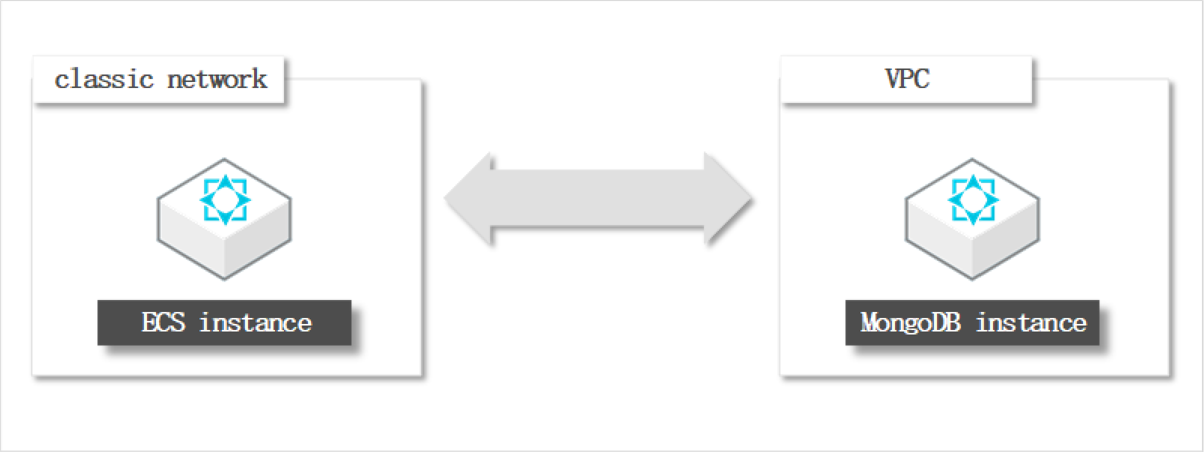
If the ECS instance is in a classic network and the ApsaraDB for MongoDB instance is in a VPC, or the MongoDB instance is in a classic network and the ECS instance is in a VPC, you can use the methods described in this topic to quickly connect the ECS instance to the ApsaraDB for MongoDB instance.

Prerequisites

- The ECS instance and ApsaraDB for MongoDB instance belong to the same Alibaba Cloud account and are in the same region.
- You also need to add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

> **Note** For more information about how to obtain the IP address of an ECS instance, see How to query the IP address of an ECS instance.

Connect an ECS instance in a classic network to an ApsaraDB for MongoDB instance in a VPC



You can connect an ECS instance in a classic network to an ApsaraDB for MongoDB instance in a VPC by using the following methods:

- Migrate the ECS instance to the VPC to which the ApsaraDB for MongoDB instance belongs. For more information, see Migrate an ECS instance to a VPC.
- Change the network type of the ApsaraDB for MongoDB instance to classic network. For more information, see Switch from a VPC to a classic network.
- Use ClassicLink.

  > **Note** The ClassicLink-based interconnection is a temporary solution in special conditions. To achieve high-speed connection in the production environment, we recommend that you create the ECS and ApsaraDB for MongoDB instances in the same VPC.

  Before you create a ClassicLink connection, make sure that you understand the limits of ClassicLink. For more information, see ClassicLink.

  To enable ClassicLink, perform the following steps:

  i. Log on to the VPC console.

  ii. Select the region of the VPC and click the ID of the VPC.

  iii. On the **VPC Details** page, click **Enable ClassicLink**. In the dialog box that appears, click **OK**.

  iv. Log on to the ECS console.

  v. In the left-side navigation pane, click **Instances**.

vi. In the upper-left corner of the page, select the region where the instance resides.

vii. In the **Operation** column corresponding to the ECS instance in a classic network, choose **More > Network and Security Group > Set classic link**.

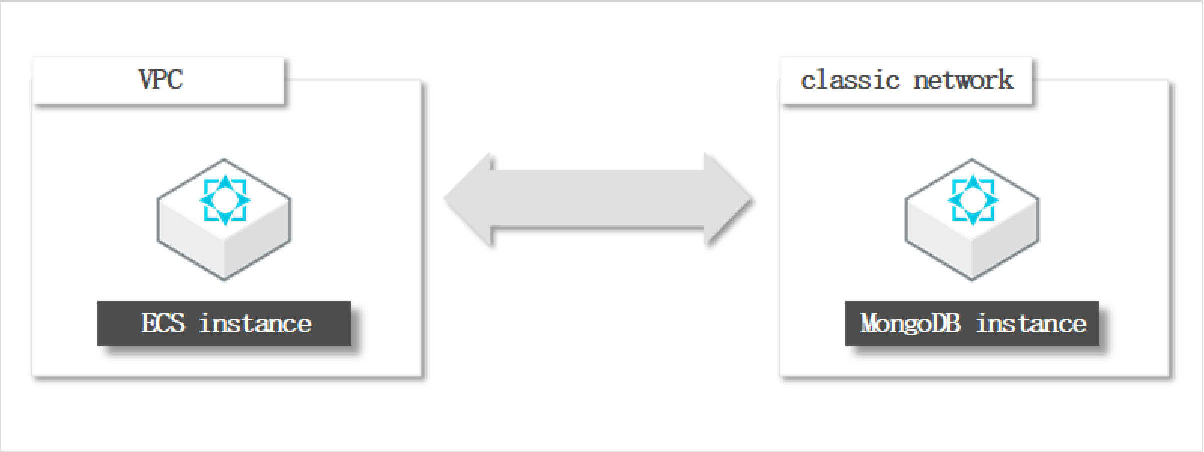
viii. In the dialog box that appears, select the VPC to which the ApsaraDB for MongoDB instance belongs and click **OK**.

ix. In the **Connect to VPC** dialog box that appears, click **Go to the instance security group list and add ClassicLink rules**.

x. Click **Add ClassicLink Rule**. Configure the following parameters and then click **OK**.

Parameter	Description
Classic Security Group	The name of the classic network security group.
Select VPC Security Group	Select a VPC security group.
Mode	Select an authorization mode. ■ Classic <=> VPC: allows ECS instances in a VPC and cloud resources in a classic network to access each other. We recommend that you select this mode. ■ Classic => VPC: allows ECS instances in a classic network to access cloud resources in a VPC. ■ VPC => Classic: allows cloud resources in a VPC to access ECS instances in a classic network.
Protocol	Select a communication protocol.
Port Range	Specify the port range in the format of xx/xx. The port used here is port 3717 for MongoDB instances. Enter **3717/3717**.
Priority	The priority of the rule. The smaller the value, the higher the priority.
Description	The description of the security group. It must be 2 to 256 characters in length and cannot start with http:// or https://.

Connect an ECS instance in a VPC to an ApsaraDB for MongoDB instance in a classic network



Switch the network type of the ApsaraDB for MongoDB instance to the VPC to which the ECS instance belongs. For more information, see Switch from a classic network to a VPC.

Note

- You cannot change the network type of standalone instances.
- Switching network types will cause a transient disconnection of the ApsaraDB for MongoDB instance. Perform this operation during off-peak hours or ensure that your application has a reconnection mechanism to prevent negative impacts on your business.

4.5. How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions

If an ECS instance and an ApsaraDB for MongoDB instance are in different regions, you can use the methods described in this topic to quickly connect the ECS instance to the ApsaraDB for MongoDB instance.

Method 1: Migrate the ApsaraDB for MongoDB instance to the region where the ECS instance is located

This method uses the data migration feature of Data Transmission Service (DTS) to migrate the ApsaraDB for MongoDB instance to the region where the ECS instance is located. For example, you can migrate a MongoDB instance from China (Qingdao) to China (Hangzhou).

1. Create an ApsaraDB for MongoDB instance in the region where the ECS instance is located. For more information, see Create an instance. Skip this step if you have already created an ApsaraDB for MongoDB instance.
2. Migrate the MongoDB database from the instance in the source region to the instance in the destination region. For more information, see Migrate the data of an ApsaraDB for MongoDB instance across regions.
3. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

> ⓘ **Note** For more information about how to obtain the IP address of an ECS instance, see How to query the IP address of an ECS instance.

Method 2: Migrate the ECS instance to the region where the ApsaraDB for MongoDB instance is located

You can use the custom image feature or the migration tool to migrate the ECS instance data from the original region to the region where the ApsaraDB for MongoDB instance is located. For example, you can migrate the ECS instance from the China (Qingdao) region to the China (Hangzhou) region.

- Create a custom image from the ECS instance and then create an ECS instance in the region where the ApsaraDB for MongoDB instance is located from the custom image (this method is recommended).
  i. Create a custom image from the ECS instance.
  ii. Copy the created custom image to the region where the ApsaraDB for MongoDB instance is located. For more information, see Copy an image.
  iii. Create an ECS instance from the custom image.

  > ⓘ **Note** When creating the ECS instance, select the same VPC as the ApsaraDB for MongoDB instance.

  iv. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

  > ⓘ **Note** For more information about how to obtain the IP address of an ECS instance, see How to query the IP address of an ECS instance.

- Use the migration tool to migrate the ECS instance to the region where the ApsaraDB for MongoDB instance is located.
  i. Migrate the ECS instance to the region where the ApsaraDB for MongoDB instance is located. For more information, see Migrate ECS instances.
  ii. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

4.6. Connect an ECS instance with an ApsaraDB for MongoDB instance in another Alibaba Cloud account

If an ECS instance and an ApsaraDB for MongoDB instance do not belong to the same Alibaba Cloud account, you can use the methods in this topic to connect the ECS instance to the ApsaraDB for MongoDB instance over an internal network.

Method 1: Migrate the ApsaraDB for MongoDB instance to the Alibaba Cloud account to which the ECS instance belongs

This method uses the data migration feature of Data Transmission Service (DTS) to migrate the ApsaraDB for MongoDB database to the Alibaba Cloud account to which the ECS instance belongs.

Procedure

1. Create an ApsaraDB for MongoDB instance in the Alibaba Cloud account to which the ECS instance belongs. For more information, see Create a replica set instance. Skip this step if you have created an ApsaraDB for MongoDB instance.

   > **Note** When you create the ApsaraDB for MongoDB instance, select the same **region**, **zone**, and **VPC** as the ECS instance.

2. Migrate the MongoDB database from the instance that belongs to the source Alibaba Cloud account to the instance that belongs to the destination Alibaba Cloud account. For more information, see Migrate data between ApsaraDB for MongoDB instances created by different Alibaba Cloud accounts.
3. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

   > **Note** For information about how to obtain the IP address of an ECS instance, see How do I query IP addresses of ECS instances?

Method 2: Migrate the ECS instance to the Alibaba Cloud account to which the ApsaraDB for MongoDB instance belongs

This method migrates the ECS instance to the Alibaba Cloud account to which the ApsaraDB for MongoDB instance belongs by sharing the ECS instance as a custom image.

Prerequisites

The ECS instance and ApsaraDB for MongoDB instance are in the same region. Images cannot be shared across regions.

Procedure

1. Create a custom image from the ECS instance.
2. Share the custom image to the Alibaba Cloud account to which the ApsaraDB for MongoDB instance belongs. For more information, see Share or unshare custom images.
3. Create an ECS instance from the custom image.

   > **Note** When you create the ECS instance, select the same VPC as the ApsaraDB for MongoDB instance.

4. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

> **Note** For information about how to obtain the IP address of an ECS instance, see How do I query IP addresses of ECS instances?

Method 3: Establish a connection between the ECS instance and ApsaraDB for MongoDB instance by using Cloud Enterprise Network

This method uses Cloud Enterprise Network (CEN) to establish a connection between the VPCs that belong to different Alibaba Cloud accounts to connect the ECS instance to the ApsaraDB for MongoDB instance.

> **Note** Make sure that the CIDR blocks of the VPCs or vSwitches involved do not conflict with each other.

Procedure

1. Switch the network type of the ApsaraDB for MongoDB instance to VPC. For more information, see Switch the network type of an ApsaraDB for MongoDB instance. If the network type is VPC, skip this step.
2. Switch the network type of the ECS instance to VPC. For more information, see Migrate ECS instances. If the network type is VPC, skip this step.
3. Based on the running environment, select one of the following CEN-based connections over an internal network. For more information, see
   - Connect network instances created in the same region but by different accounts.
   - Connect network instances created by different accounts and in different regions.
4. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

   > **Note** For information about how to obtain the IP address of an ECS instance, see How do I query IP addresses of ECS instances?

4.7. Connect to an ApsaraDB for MongoDB instance over the Internet

This topic describes how to connect a local client to an ApsaraDB for MongoDB instance over the Internet.

Prerequisites

A public endpoint for the ApsaraDB for MongoDB instance is obtained. For more information, see the following topics:

- Apply for a public endpoint for a standalone ApsaraDB for MongoDB instance
- Apply for a public endpoint for an ApsaraDB for MongoDB instance
- Apply for a public endpoint for a sharded cluster instance

Precautions

Read this topic only when you want to connect a local client to an ApsaraDB for MongoDB instance. If you want to connect to an ApsaraDB for MongoDB instance by using an ECS instance, you can obtain both the public and private IP addresses from the ECS instance details page in the ECS console.

If you connect to an ApsaraDB for MongoDB instance over the Internet, security risks may arise. We recommend that you connect to an ApsaraDB for MongoDB instance by using an ECS instance.

Method 1: Query an IP address library for the public IP address of your local client and connect to an ApsaraDB for MongoDB instance

You can query an IP address library for the public IP address of your local client and connect to an ApsaraDB for MongoDB instance.

1. Query the public IP address of your local client.
2. Add the public IP address to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.
3. Log on to the ApsaraDB for MongoDB instance by using the mongo shell from your local client. For more information, see Connect to an ApsaraDB for MongoDB instance.

   > **Note** You can also log on to the ApsaraDB for MongoDB instance by using other client tools.

You may have added the public IP address of your local client to the whitelist but still fail to connect to the ApsaraDB for MongoDB instance. However, after you add 0.0.0.0/0 to the whitelist, you can connect to the instance. In this case, we recommend that you query the connection information for the public IP address. For more information, see Method 2: Query the connection information for the public IP address of your local client and connect to an ApsaraDB for MongoDB instance.

Method 2: Query the connection information for the public IP address of your local client and connect to an ApsaraDB for MongoDB instance

You can query the connection information for the public IP address of your local client and connect to an ApsaraDB for MongoDB instance.

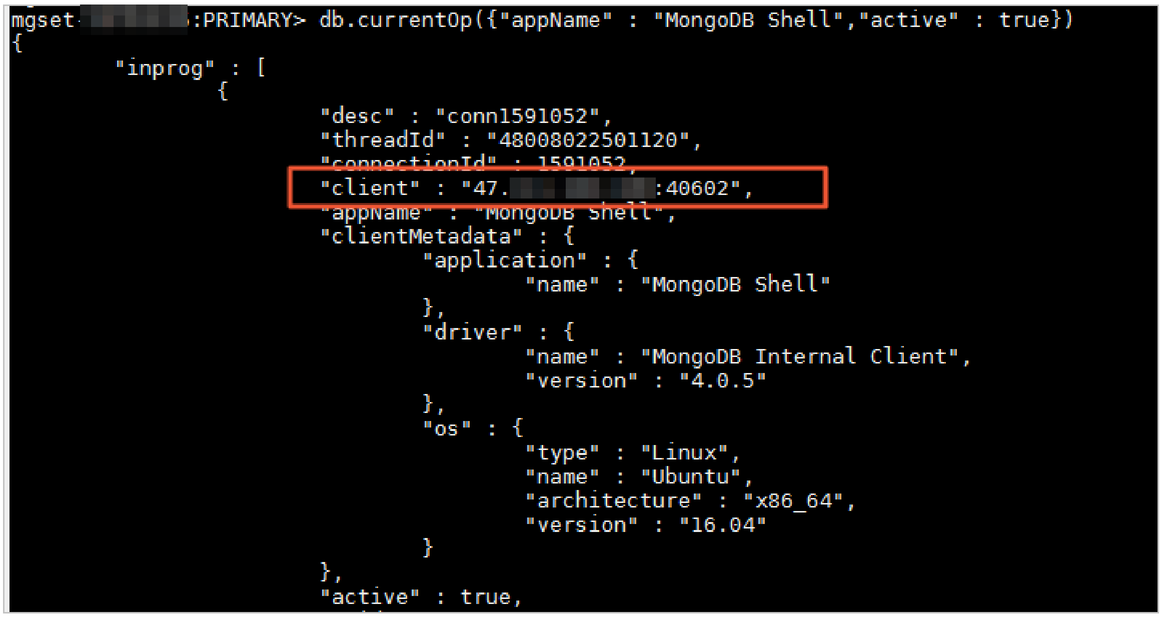
1. Add the 0.0.0.0/0 entry to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

   > **Note** If you add the 0.0.0.0/0 entry to the whitelist, all servers are granted access to the ApsaraDB for MongoDB instance. This may raise security risks. Exercise caution when you add the 0.0.0.0/0 entry to a whitelist. Remove the 0.0.0.0/0 entry if you no longer need it.

2. Log on to the ApsaraDB for MongoDB instance by using the mongo shell from your local client. For more information, see Connect to an ApsaraDB for MongoDB instance.
3. Run the following command to query information about the client where you log on:

   ```
   db.currentOp({"appName" : "MongoDB Shell","active" : true})
   ```

   The following figure shows an example.



```
mgset[redacted]:PRIMARY> db.currentOp({"appName" : "MongoDB Shell","active" : true})
{
        "inprog" : [
                {
                        "desc" : "conn1591052",
                        "threadId" : "48008022501120",
                        "connectionId" : 1591052,
                        "client" : "47.[redacted]:40602",
                        "appName" : "MongoDB Shell",
                        "clientMetadata" : {
                                "application" : {
                                        "name" : "MongoDB Shell"
                                },
                                "driver" : {
                                        "name" : "MongoDB Internal Client",
                                        "version" : "4.0.5"
                                },
                                "os" : {
                                        "type" : "Linux",
                                        "name" : "Ubuntu",
                                        "architecture" : "x86_64",
                                        "version" : "16.04"
                                }
                        },
                        "active" : true,
```

Note If you log on to the ApsaraDB for MongoDB instance using other methods, you can run the following command to query information about all clients:

```
db.runCommand({currentOp: 1, "active" : true})
```

4. Add the IP address obtained in the preceding step to the whitelist of the ApsaraDB for MongoDB instance, and remove the 0.0.0.0/0 entry that you added in Step 1 from the whitelist.

References

If the public IP address of your local client dynamically changes, you can use one of the following methods to connect to an ApsaraDB for MongoDB instance:

- Connect to the instance by using an ECS instance.
- Connect to the instance by using a VPN. For more information, see Connect a local client to an ApsaraDB for MongoDB instance through an SSL VPN tunnel.

4.8. Connect a local client to an ApsaraDB for MongoDB instance through an SSL VPN tunnel

This topic describes how to connect a local client to an ApsaraDB for MongoDB instance through an SSL VPN tunnel, which provides a secure connection between the local client and the VPC housing the ApsaraDB for MongoDB instance. With this tunnel, you can manage the ApsaraDB for MongoDB instance from the local client with ease. SSL is short for Secure Sockets Layer, VPN for virtual private network, and VPC for Virtual Private Cloud.

Scenarios

- The public IP address of the local client changes dynamically. As a result, you must frequently update the whitelist that contains the public IP address of the local client on the ApsaraDB for MongoDB console. If you do not delete expired IP addresses at the earliest opportunity, security risks may arise.

- A higher level of security is required when you connect to an ApsaraDB for MongoDB instance over the Internet.
- You need to log on to the ApsaraDB for MongoDB instance from an ECS instance over the Internet. This may cause security risks. Therefore, you must separate ECS management permissions from ApsaraDB for MongoDB database permissions.

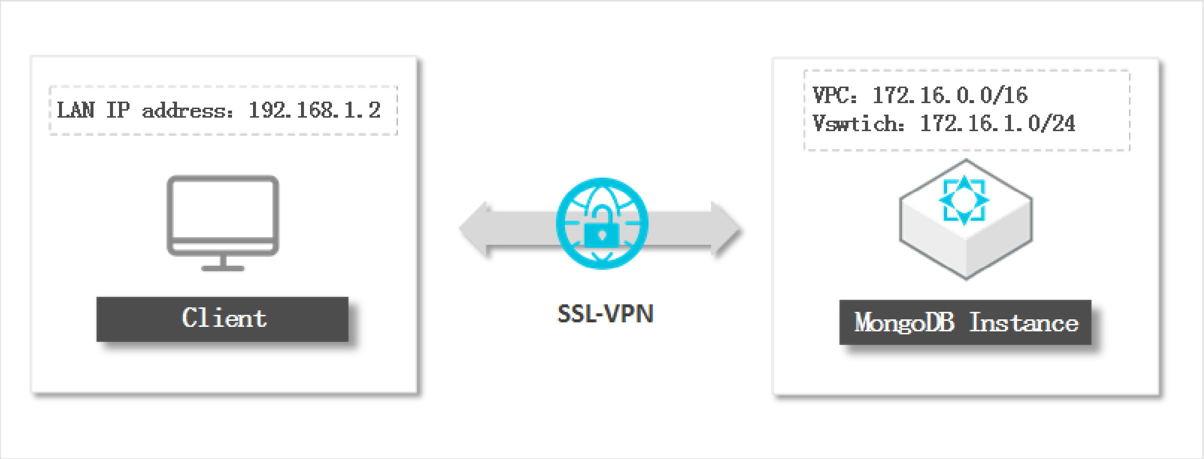
Billing

You are charged to create a VPN gateway. For more information, see Billing.

Prerequisites

- VPC is the network type of the ApsaraDB for MongoDB instance. For more information about how to switch the network type from Classic Network to VPC, see Switch from Classic Network to VPC.
- The Classless Inter-Domain Routing (CIDR) block of the local client is different from that of the ApsaraDB for MongoDB instance.
- The local client can access the Internet.

Networking



Step 1 Create a VPN gateway

See Create a VPN gateway.

Step 2 Create an SSL server

See Create an SSL server.

Step 3 Create an SSL client

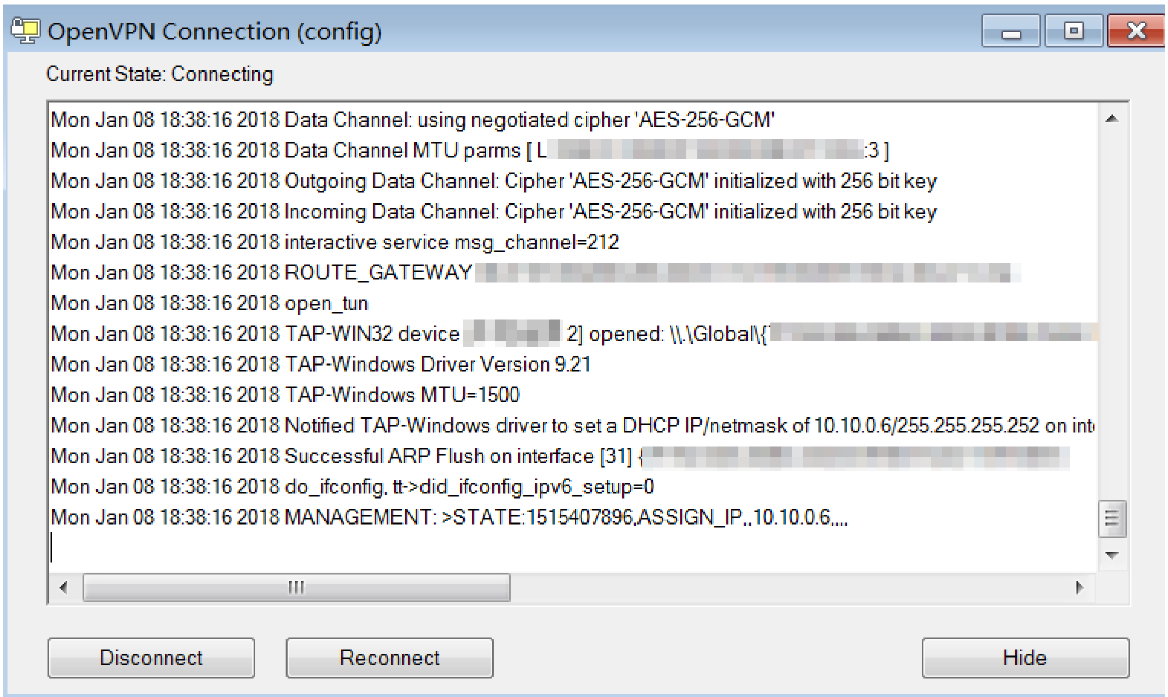
See Create an SSL client certificate.

Log on to the ApsaraDB for MongoDB instance from the client through the SSL VPN tunnel

This section uses Windows as an example. For more information about other operating systems, see Remote access from a Linux client and Remote access from a Mac client.

1. Log on to the VPC console.

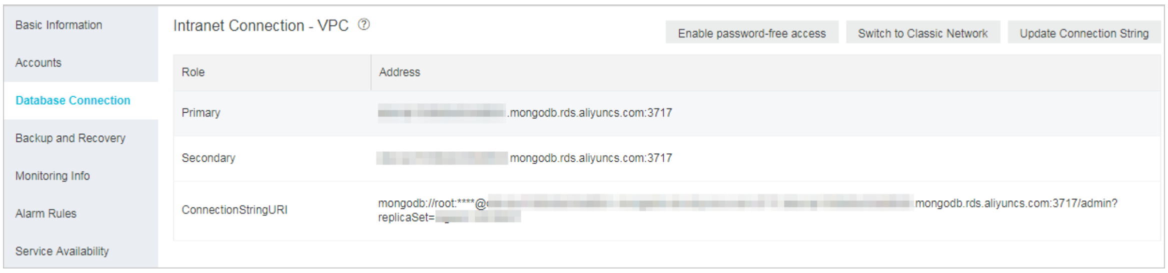
2. In the upper-left corner of the page, select a region.
3. In the left-side navigation pane, choose **VPN > SSL Clients**.
4. On the right of the SSL client you have created, click **Download** to download the generated client certificate package.
5. Download the OpenVPN software package and install OpenVPN on the client you want to connect through the SSL VPN tunnel.
6. Decompress the client certificate package that you downloaded and copy the client certificate file to the config folder of the OpenVPN installation directory.
7. Click **Connect**.



Current State: Connecting

```
Mon Jan 08 18:38:16 2018 Data Channel: using negotiated cipher 'AES-256-GCM'
Mon Jan 08 18:38:16 2018 Data Channel MTU parms [ L [redacted]:3 ]
Mon Jan 08 18:38:16 2018 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Mon Jan 08 18:38:16 2018 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Mon Jan 08 18:38:16 2018 interactive service msg_channel=212
Mon Jan 08 18:38:16 2018 ROUTE_GATEWAY [redacted]
Mon Jan 08 18:38:16 2018 open_tun
Mon Jan 08 18:38:16 2018 TAP-WIN32 device [redacted] 2] opened: \\.\Global\{[redacted]
Mon Jan 08 18:38:16 2018 TAP-Windows Driver Version 9.21
Mon Jan 08 18:38:16 2018 TAP-Windows MTU=1500
Mon Jan 08 18:38:16 2018 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.10.0.6/255.255.255.252 on int
Mon Jan 08 18:38:16 2018 Successful ARP Flush on interface [31] {[redacted]
Mon Jan 08 18:38:16 2018 do_ifconfig, tt->did_ifconfig_ipv6_setup=0
Mon Jan 08 18:38:16 2018 MANAGEMENT: >STATE:1515407896,ASSIGN_IP,,10.10.0.6,,,,
```

8. Add the CIDR block of the VPC to which the ApsaraDB for MongoDB instance belongs to a whitelist of this instance. For this example, add the IP address 172.16.1.0/24 to the whitelist.
9. Log on to the ApsaraDB for MongoDB console.
10. Obtain the internal endpoints of the ApsaraDB for MongoDB instance. For more information, see Connect to a replica set instance through the mongo shell.



Intranet Connection - VPC

Role	Address
Primary	[redacted].mongodb.rds.aliyuncs.com:3717
Secondary	[redacted]mongodb.rds.aliyuncs.com:3717
ConnectionStringURI	mongodb://root:****@[redacted]mongodb.rds.aliyuncs.com:3717/admin? replicaSet=[redacted]

11. Use the mongo shell or other management tools to log on to the ApsaraDB for MongoDB instance.

> **Note** Log on using an internal endpoint of the ApsaraDB for MongoDB instance.

5.Account management

5.1. Reset the password for an ApsaraDB for MongoDB instance

This topic describes how to reset the password for an ApsaraDB for MongoDB instance. If you forget your password or did not set the password when you created an instance, you can reset the password of the instance.

Limits

You can only reset the password of the root user or an account for a shard or config server.

> ? **Note** If you want to manage database users created by running the `db.createUser` command, you can use DMS or the mongo shell. For more information, see Manage user permissions on MongoDB databases or Log on to the MongoDB instance through the mongo shell.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Accounts**.
6. Perform one of the following operations as needed.
   - Reset the password of the root user.

     Find the root user and click **Reset Password** in the **Actions** column.
   - Reset the password of an account for a shard or config server in a sharded cluster instance.

     > ? **Note** If no endpoints are obtained for the shard or config server, this operation cannot be performed. For more information about how to obtain the endpoints, see Apply for a connection string of a shard or Configserver node.

     Find the account created when you apply for an endpoint for a shard or config server, and click **Reset Password** in the **Actions** column. For this example, find shardaccount. For more information, see Apply for a connection string of a shard or Configserver node.
7. In the dialog box that appears, enter a new password and confirm it.

ⓘ Note

- The password must be 8 to 32 characters in length.
- The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! # $ % ^ & * ( ) _ + - =

8. Click **OK**.

5.2. Manage user permissions on MongoDB databases

In Data Management (DMS), you can manage users for MongoDB databases and grant the users the permissions of different roles. The roles are **Common operation role**, **Administrator action role**, **Instance-level role**, **Cluster administrator role**, **Backup and Recovery roles**, and Super role.

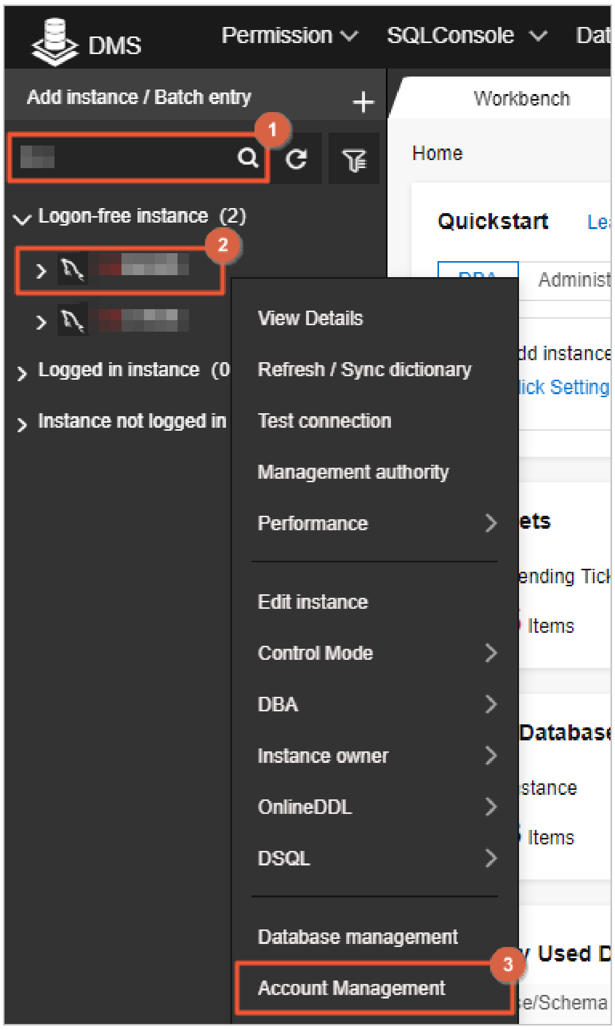
Prerequisites

- A MongoDB database is registered in DMS.
- You are assigned a required role for the MySQL database instance that is registered in DMS. The role varies based on the control mode of the instance. The following table describes the details.

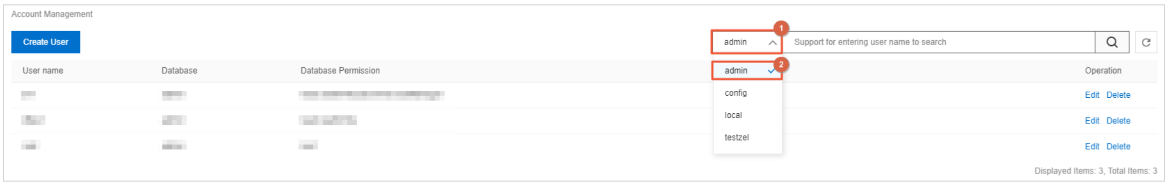
Control mode	Role requirement
Security Collaboration	You must be a DMS administrator, a database administrator (DBA), or the owner of the relevant database instance.
Stable Change	No specific role is required.
Flexible Management	No specific role is required.

Create a user

1. Log on to the DMS console.
2. In the search box at the top of the left-side navigation pane, enter the name of the MySQL database whose permissions you want to manage. From the matched result, find the instance to which the database belongs.
3. Right-click the instance and select **Account Management**.



4. On the **Account Management** page, select a database from the drop-down list.

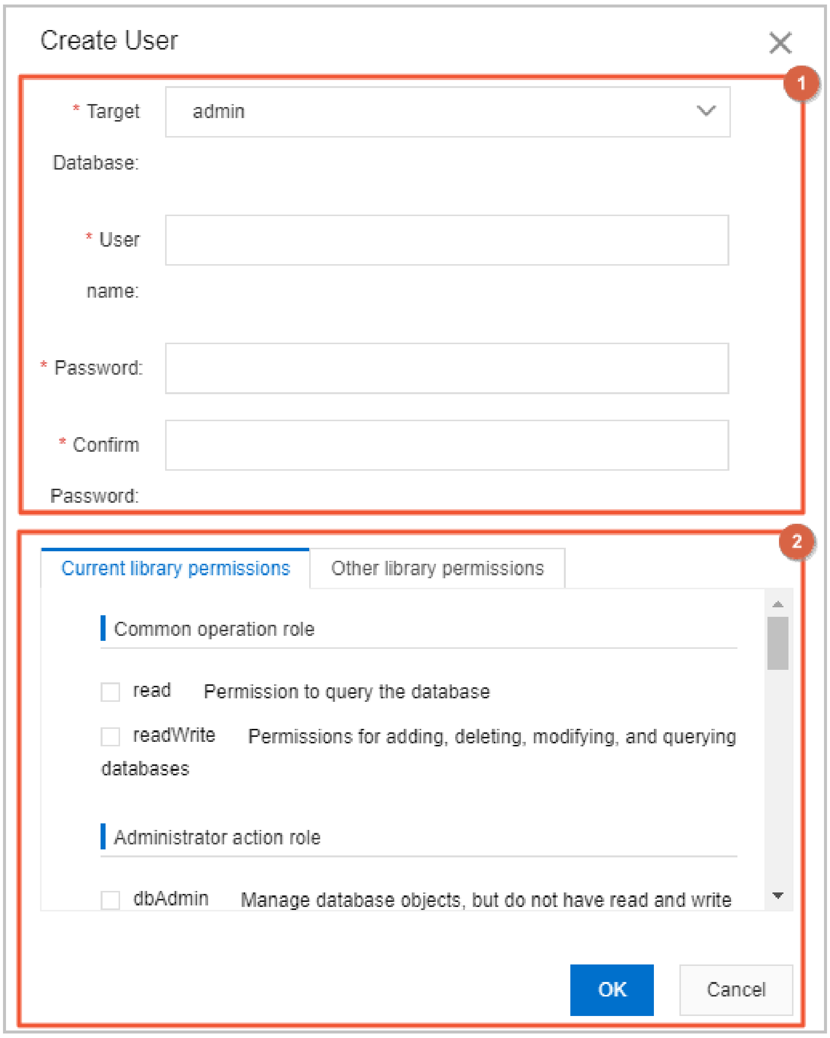


User name	Database	Database Permission	Operation
[redacted]	[redacted]	[redacted]	Edit Delete
[redacted]	[redacted]	[redacted]	Edit Delete
[redacted]	[redacted]	[redacted]	Edit Delete

Displayed Items: 3, Total Items: 3

5. Click **Create User** in the upper-left corner.

6. In the **Create User** dialog box, set relevant parameters.



Create User

* Target Database: admin

* User name:

* Password:

* Confirm Password:

Common operation role

□ read Permission to query the database

□ readWrite Permissions for adding, deleting, modifying, and querying databases

Administrator action role

□ dbAdmin Manage database objects, but do not have read and write

i. Configure user information. Set the parameters as described in the following table.

Parameter	Description
Target Database	The database for which you want to create the user. Note - If you select a database other than the admin database, the user to be created is a common user. - If you select the **admin** database, the user to be created is a privileged user.
User name	The name of the user. - The name cannot contain Chinese characters. - It can contain letters, digits, and special characters. - The name can contain the following special characters: !#$%^&*()_+-=
Password	The password that the user can use to log on to the database. To ensure data security, the password can be 8 to 32 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The password can contain the following special characters: !#$%^&*()_+-=
Confirm Password	The password that the user can use to log on to the database. Enter the same password again to confirm the password.

ii. Grant permissions to the user.

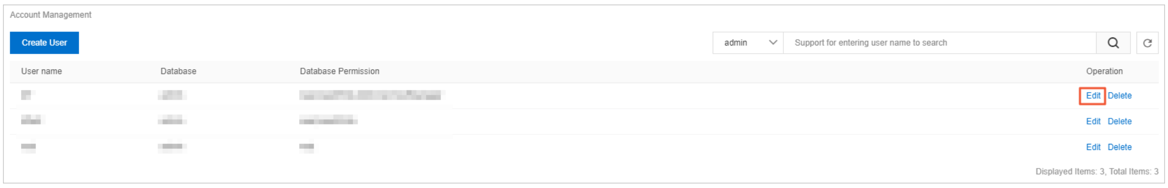
> Note
>
> - Assume that you select the admin database:
>
>   On the **Current library permissions** tab, you can grant permissions of different roles to the user. The roles are **Common operation role**, **Administrator action role**, **Instance-level role**, **Cluster administrator role**, **Backup and Recovery roles**, and **Super role**. For more information, see Permissions of different roles.
>
>   On the **Other library permissions** tab, you can grant permissions on other databases in the current instance to the user.
>
> - Assume that you select a database other than the admin database:
>
>   On the **Current library permissions** tab, you can grant permissions of **Common operation role** and **Administrator action role** to the user. For more information, see Permissions of different roles.
>
>   You cannot grant permissions on other databases to the user on the **Other library Permissions** tab.

7. In the Create User dialog box, click **OK**.

> Note If the database instance is in Security Collaboration mode, SQL statements can be generated based on the parameters you set. However, the SQL statements may fail to be executed due to security rules. In this case, you can perform operations as prompted or contact the DBA or DMS administrator.

Edit a user

1. Log on to the DMS console.
2. In the search box at the top of the left-side navigation pane, enter the name of the MySQL database whose permissions you want to manage. From the matched result, find the instance to which the database belongs.
3. Right-click the instance and select **Account Management**.
4. On the **Account Management** page, select a database from the drop-down list.
5. Find the user whose information you want to edit and click **Edit** in the **Operation** column.



Account Management

Create User

admin

Support for entering user name to search

User name	Database	Database Permission	Operation
[redacted]	[redacted]	[redacted]	Edit Delete
[redacted]	[redacted]	[redacted]	Edit Delete
[redacted]	[redacted]	[redacted]	Edit Delete

Displayed Items: 3, Total Items: 3

Delete a user

1. Log on to the DMS console.
2. In the search box at the top of the left-side navigation pane, enter the name of the MySQL database whose permissions you want to manage. From the matched result, find the instance to which the database belongs.

3. Right-click the instance and select **Account Management**.
4. On the **Account Management** page, select a database from the drop-down list.
5. Find the user who you want to delete and click **Delete** in the **Operation** column.
6. In the Prompt message, click **OK**.

Permissions of different roles

The following table describes the permissions of different roles. For more information, see MongoDB official website.

Role	Permission	Description
Common operation role	read	Enables a user to query data in the database.
	readWrite	Enables a user to insert, delete, update, or query data in the database.
Administrator action role	dbAdmin	Enables a user to manage data in the database, but not to read data from or write data to the database.
	userAdmin	Enables a user to create users for the database.
	dbOwner	Enables a user to perform all operations on the database.
Instance-level role	readAnyDatabase	Enables a user to query data in all databases of the instance.
	readWriteAnyDatabase	Enables a user to insert, delete, update, or query data in all databases of the instance.
	userAdminAnyDatabase	Enables a user to create users for all databases of the instance.
	dbAdminAnyDatabase	Enables a user to manage data in all databases of the instance, but not to read data from or write data to the databases.
Cluster administrator role	hostManager	Enables a user to manage data in the database, but not to read data from or write data to the database.
	clusterMonitor	Enables a user to query clusters and replication sets.
	clusterManager	Enables a user to manage and monitor clusters and replication sets.
	clusterAdmin	Enables a user to perform all operations on clusters.
Backup and Recovery roles	backup	Enables a user to query data in all databases of the instance.
	restore	Enables a user to insert, delete, update, or query data in all databases of the instance.
Super role	Root	Enables a user to perform all operations on all resources in an instance.

6.Instance management

6.1. Changing Instance Configuration

6.1.1. Configuration change overview

ApsaraDB for MongoDB allows you to change configurations to meet your needs in most scenarios. It also provides solutions to the configuration items that you cannot change.

For more information about instance specifications, see Instance types.

For more information about correspondence and restrictions between versions and storage, see MongoDB versions and storage engines.

? **Note** For more information about precautions and procedure for version upgrade, see Upgrade MongoDB versions.

Standalone instances

Item	Configuration change supported	Description
Specifications	Yes	For more information, see Change the configuration of a standalone or replica set instance.

Item	Configuration change supported	Description
Storage space	Yes	For more information, see Change the configuration of a standalone or replica set instance. If the billing method is subscription, you cannot downgrade the storage space. You must perform the following operations: 1. Create a pay-as-you-go standalone instance and select the required storage space. For more information, see Create a standalone instance. Note The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Use DTS to migrate data from the original instance to the new instance. For more information, see Overview. 3. Test and verify the new instance. If it runs normally, switch business to the new instance. Note If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.

Item	Configuration change supported	Description
Number of nodes	No	You cannot change the number of nodes, architecture, and storage engine for a standalone instance. You must perform the following operations: 1. Create a pay-as-you-go instance. Select the required number of nodes, architecture, and storage engine. Note - To increase the number of nodes, you must create a replica set instance. - The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Use DTS to migrate data from the original instance to the new instance. For more information, see Overview. 3. Test and verify the new instance. If it runs normally, switch business to the new instance. Note If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.
Instance architecture		
Storage engines		

Replica set instances

Item	Configuration change supported	Description
Number of nodes	Yes	For more information, see 变更副本集实例节点数.
Specifications	Yes	For more information, see Change the configuration of a standalone or replica set instance.

Item	Configuration change supported	Description
Storage space	Yes	For more information, see Change the configuration of a standalone or replica set instance. If the billing method is subscription, you cannot downgrade the storage space. You must perform the following operations: 1. Restore data to a new ApsaraDB for MongoDB instance by point in time Create a pay-as-you-go standalone instance and select the required storage space. **Note** The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Test and verify the new instance. If it runs normally, switch business to the new instance. **Note** If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 3. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.

Item	Configuration change supported	Description
Instance architecture	No	You cannot change the architecture and storage engine of a replica set instance. You must perform the following operations: 1. Create a pay-as-you-go instance. Select the required architecture and storage engine. **Note** The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Use DTS to migrate data from the original instance to the new instance. For more information, see Overview. 3. Test and verify the new instance. If it runs normally, switch business to the new instance. **Note** If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.
Storage engines		

Sharded cluster instances

You can change the specifications and storage space of a sharded cluster instance.

Note You cannot change the architecture and storage engine of a sharded cluster instance. Creating a new instance will cause a long-period shutdown and have a great impact on the business. Therefore, we do not recommend this method.

Component	Item	Description
Mongos node	Specifications	For more information, see 变更分片集群实例配置.

Component	Item	Description
Shard node	Specifications	For more information, see 变更分片集群实例配置.
	Storage space	If the billing method is subscription, you cannot downgrade the storage space. You must perform the following operations: 1. Restore data to a new ApsaraDB for MongoDB instance by point in time Create a pay-as-you-go standalone instance and select the required storage space. **Note** The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Test and verify the new instance. If it runs normally, switch business to the new instance. **Note** If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 3. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.
Configserver node	Specifications and storage space	A Configserver node uses a fixed three-node replica set architecture. By default, 1 core, 2 GB memory, and 20 GB storage space are selected. You cannot change these items.

6.1.2. Change the configuration of a standalone or replica set instance

You can change the configuration of a standalone or replica set instance if the configuration is excessive or cannot meet the performance requirements of your application.

Precautions

- When you change the configuration, the new storage space must be larger than the storage space occupied by the current instance.
- If the billing method is subscription, the interval between two configuration downgrades cannot be less than 60 days.
- When the billing method is subscription, you cannot downgrade the storage space. You can use other methods to reduce the storage space. For more information, see Configuration change overview.
- You cannot change the storage engine or the instance type. For example, you can change a standalone instance to a replica set instance. You can use other methods to change these items. For more information, see Configuration change overview.

Billing rules

For more information, see Configuration change fees.

Impacts

- Configuration changes do not cause data loss.
- Pre-operations for configuration changes to an instance do not affect the running of the instance. However, when configuration changes are formally executed on the instance, most operations related to databases, accounts, and network cannot be performed. One or two transient connections of up to 30 seconds may occur. For more information, see Select the switching time.
- The duration of a configuration change depends on multiple factors such as network conditions, task queues, and data volume. We recommend that you change configurations during off-peak hours and make sure that your applications are configured with automatic reconnection policies.

Select the switching time

On the Change Configuration page, you can specify the switching time. The following table describes details about switching time.

Parameter	Instance status	Impact
Switch Within Maintenance Window	The instance immediately changes to the **Changing Configuration** state.	The system performs pre-operations, which do not affect the running of the instance or cause transient connections. Configuration changes are formally executed within the maintenance period you set. For example, if the preset maintenance period is 2:00 to 3:00, configuration changes are performed during this window. Most operations related to databases, accounts, and network cannot be performed. One or two transient connections of up to 30 seconds may occur. **Note** For more information about how to modify a maintenance window, see Specify a maintenance period.
Switch Immediately After Data Migration		Configuration changes are immediately performed. Most operations related to databases, accounts, and network cannot be performed. One or two transient connections of up to 30 seconds may occur.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Change the configuration of the instance.

   If the billing method of the instance is pay-as-you-go, perform the following operations:

   i. Find the instance and click its ID.

   ii. In the **Basic Information** section, click **Change Configuration**.

If the billing method of the instance is subscription, perform the following operations:

i. Find the instance and click its ID.

ii. In the **Basic Information** section, click **Upgrade** or **Downgrade**.

5. On the **Configuration Upgrade** page, specify **Replication Factor**, **Plan**, **Storage Space**, and **Switching Time** of the instance.

   For more information about instance specifications, see Instance types.

   > Note
   >
   > - For more information about the limits on the parameters, see Precautions.
   > - For more information about the selection and impacts of **Switching Time**, see Select the switching time.

6. Read and select ApsaraDB for MongoDB Agreement of Service and complete the payment.

Result

When the instance status changes to **Running**, the configuration is changed.

6.2. Upgrading Database Version

6.2.1. Upgrade the minor version of an ApsaraDB for MongoDB instance

This topic describes how to upgrade the minor version of an ApsaraDB for MongoDB instance to the latest in the ApsaraDB for MongoDB console.

Prerequisites

- The instance is a replica set or sharded cluster instance.
- The instance is not running the latest minor version. When you use the latest minor version, the ApsaraDB for MongoDB console does not display the **Upgrade Minor Version** button for the instance.

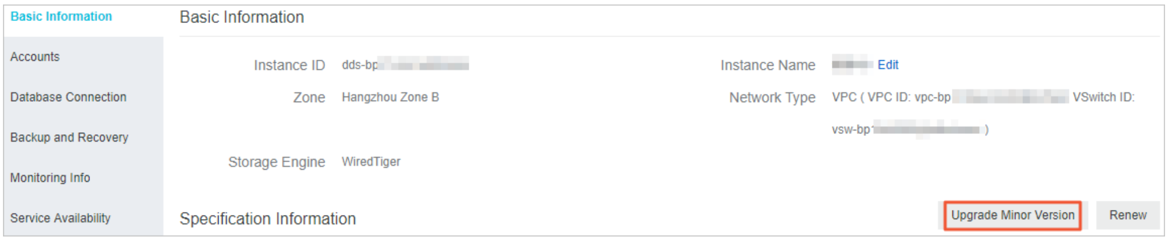
Precautions

- You cannot downgrade an instance after you upgrade it.
- When an instance undergoes a minor version upgrade, it is restarted and has a brief disconnection of less than 30 seconds. We recommend that you perform the upgrade during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected.
- To ensures better performance and stability of the instance, the system will upgrade the minor version to the latest version by default If the minor version of your instance expires or is not included in the maintenance list and the instance is upgraded, migrated, changed, Created from a backup, Created by point-in-time, or performed Restore data to a new ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the target instance and click its ID.
5. On the **Basic Information** page, click **Upgrade Minor Version**.



Basic Information

Instance ID dds-bp[redacted]

Instance Name [redacted] Edit

Zone Hangzhou Zone B

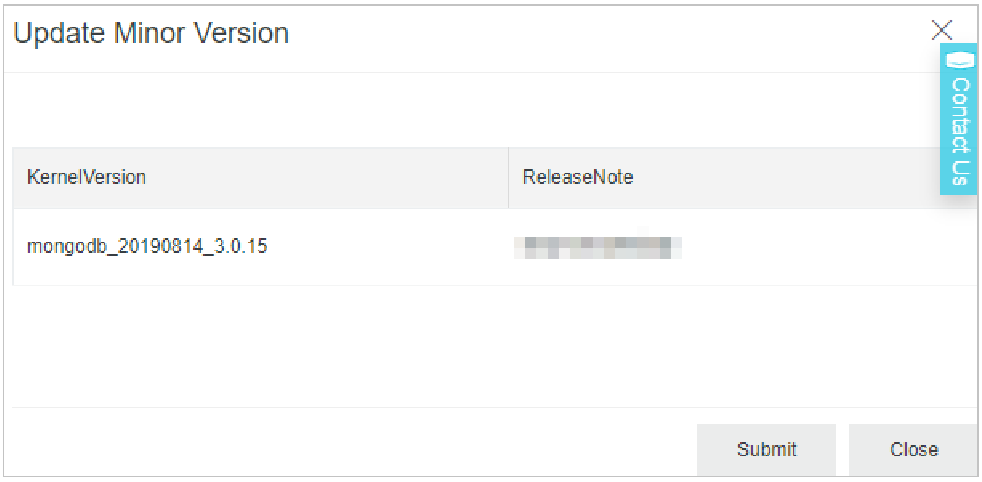
Network Type VPC (VPC ID: vpc-bp[redacted] VSwitch ID: vsw-bp[redacted])

Storage Engine WiredTiger

Specification Information

> **Note** When you use the latest minor version, the console does not display the **Upgrade Minor Version** button for the instance.

6. In the **Upgrade Minor Version** message that appears, view the version release log and determine whether to upgrade the instance.



Update Minor Version

KernelVersion	ReleaseNote
mongodb_20190814_3.0.15	[redacted]

> **Note** If you want to upgrade the instance, click **Submit**. Otherwise, click **Close**.

7. Wait until the instance status changes from **Upgrading** to **Running**.

6.2.2. Upgrade MongoDB versions

This topic describes how to upgrade the MongoDB version of an instance in the ApsaraDB for MongoDB console. ApsaraDB for MongoDB supports MongoDB 4.4, 4.2, 4.0, and 3.4.

MongoDB versions

For more information about the MongoDB versions supported by ApsaraDB for MongoDB, see Versions and storage engines.

MongoDB versions for upgrade

Instance architecture	Before upgrade	After upgrade	Description
Standalone instance	3.4 and 4.0	N/A	Upgrade to MongoDB 4.0 is not supported. If you need an ApsaraDB for MongoDB instance with MongoDB 4.0, select the version when you create the instance. For more information, see Create a standalone instance.
Replica set instance	3.2 (retired), 3.4, 4.0, 4.2, and 4.4	3.4, 4.0, and 4.2	Upgrade to MongoDB 4.4 is not supported. If you need an ApsaraDB for MongoDB instance with MongoDB 4.4, select the version when you create the instance. For more information, see 创建副本集实例.
Sharded cluster instance	3.2 (phased-out), 3.4, 4.0, and 4.2	3.4, 4.0, and 4.2	N/A

Note For more information about the features in each version, see MongoDB versions and descriptions.

Precautions

- The duration of upgrading the MongoDB version of an instance is related to the data volume of databases in this instance. We recommend that you schedule your upgrade task during off-peak hours.
- You cannot downgrade the MongoDB version of an instance after you upgrade it.

Notes

- Nodes in an instance are upgraded in turn. An instance is automatically restarted two or three times during an upgrade. We recommend that you perform the upgrade during off-peak hours or make sure that your application is configured to connect to the instance after it is disconnected.

  Note If your application runs in a production environment, we recommend that you use a connection string URI to connect your application to the instance. This way, the read and write operations of your application remain available even if a node fails as a result of a primary/secondary switchover. For more information, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

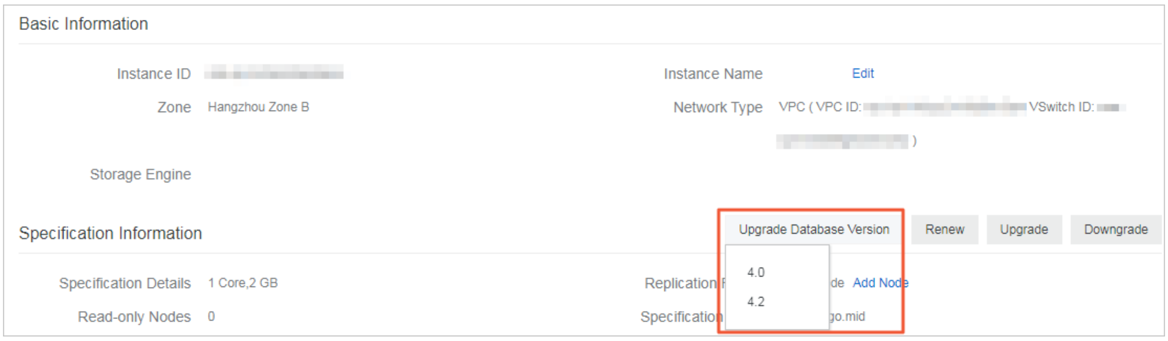
- The balancer of a sharded cluster instance is disabled during an upgrade and is enabled after the upgrade.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances**

based on the instance type.

4. Find the target instance and click its ID.
5. On the **Basic Information** page, click **Upgrade Database Version** and select the version for upgrade.



Basic Information

Instance ID [redacted]

Instance Name

Zone Hangzhou Zone B

Network Type VPC (VPC ID: [redacted] VSwitch ID: [redacted])

Storage Engine

Specification Information

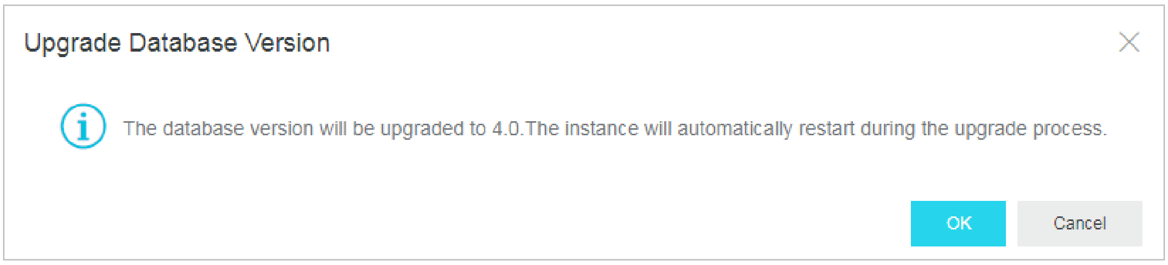
Specification Details 1 Core,2 GB

Replication

Read-only Nodes 0

Specification

6. In the **Upgrade Database Version** message, click **OK**.



Upgrade Database Version

The database version will be upgraded to 4.0.The instance will automatically restart during the upgrade process.

The instance status changes to **Upgrading**. When the instance is changed to the **Running** state, the upgrade is complete.

6.3. Specify a maintenance period

To guarantee stability, Alibaba Cloud maintains ApsaraDB for MongoDB instances at irregular intervals. You can specify a maintenance period in which you allow Alibaba Cloud to maintain your instances. We recommend that instances be maintained during off-peak hours to avoid an impact on business.

Context

Before maintenance, Alibaba Cloud sends an SMS message and an email to the respective phone number and email address that you have specified for your Alibaba Cloud account. Please check in a timely manner.

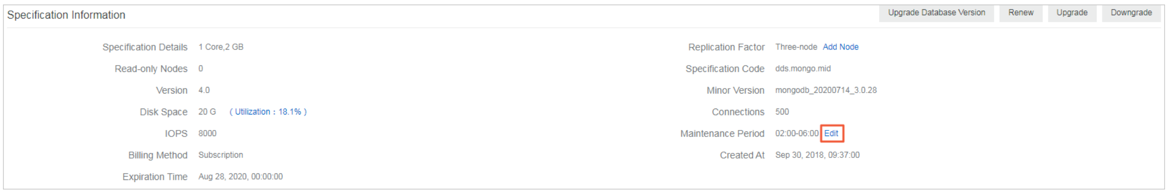
On the day of maintenance, instances enter the **Instance being maintained** status ahead of the specified maintenance period to guarantee the stability of the maintenance process. You can still connect to instances in this status. In the ApsaraDB for MongoDB console, you cannot change these instances, for example, upgrade or downgrade their configuration or restart them. However, you can manage accounts, manage ApsaraDB for MongoDB instances, or configure IP address whitelists for these instances. You can also use query features, such as performance monitoring, in the console.

During the maintenance period, instances may be disconnected transiently once or twice. You need to ensure that your applications can automatically re-establish a connection. After intermittent disconnection, instances can immediately return to normal.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, **Sharded Cluster Instances**, or **Serverless Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the **Specification Information** area, click **Edit** to the right of **Maintenance Period**.



Specification Information

Specification Details 1 Core,2 GB

Read-only Nodes 0

Version 4.0

Disk Space 20 G (Utilization : 18.1%)

IOPS 8000

Billing Method Subscription

Expiration Time Aug 28, 2020, 00:00:00

Replication Factor Three-node Add Node

Specification Code dds.mongo.mid

Minor Version mongodb_20200714_3.0.28

Connections 500

Maintenance Period 02:00-06:00 Edit

Created At Sep 30, 2018, 09:37:00

6. Specify a maintenance period for the instance and click **OK**.

6.4. View zones of nodes

ApsaraDB for MongoDB provides the zone distribution of nodes. You can view the zone distribution information in the console.

Prerequisites

Replica set or sharded cluster instances are created.

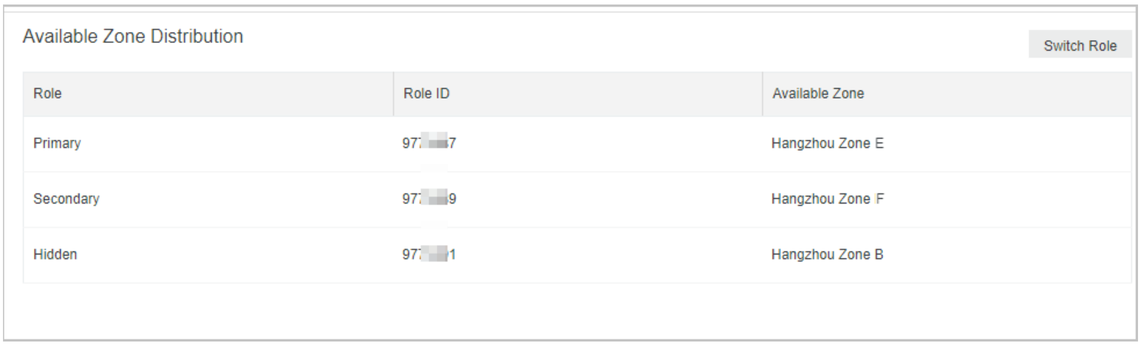
Deployment tips

ApsaraDB for MongoDB provides a zone-disaster recovery solution for replica set instances to meet the high reliability and data security requirements in business scenarios. This solution deploys the nodes of a replica set instance or the components of a sharded cluster instance in three different zones. When one of the three zones loses communication due to force majeure factors such as power failure or network failure, the high-availability system automatically triggers a switchover. This ensures the continuous availability and data security of the entire instance. For more information, see Create a multi-zone replica set instance and Create a multi-zone sharded cluster instance.

> **Note** For more information about comparison of node deployment policies for single and multiple zones, see Node deployment policies and Create a multi-zone sharded cluster instance.

View zones of nodes

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Service Availability** to view the current zone distribution.
   - Replica set instances

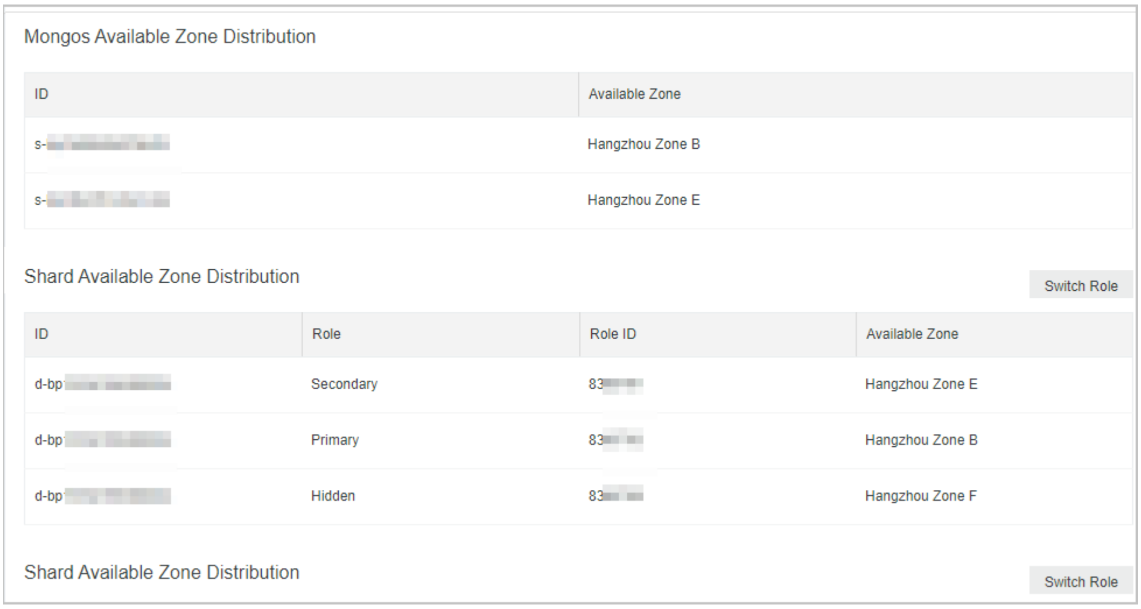


Available Zone Distribution

Switch Role

Role	Role ID	Available Zone
Primary	97[redacted]7	Hangzhou Zone E
Secondary	97[redacted]9	Hangzhou Zone F
Hidden	97[redacted]1	Hangzhou Zone B

- Sharded cluster instances



Mongos Available Zone Distribution

ID	Available Zone
s-[redacted]	Hangzhou Zone B
s-[redacted]	Hangzhou Zone E

Shard Available Zone Distribution

Switch Role

ID	Role	Role ID	Available Zone
d-bp[redacted]	Secondary	83[redacted]	Hangzhou Zone E
d-bp[redacted]	Primary	83[redacted]	Hangzhou Zone B
d-bp[redacted]	Hidden	83[redacted]	Hangzhou Zone F

Shard Available Zone Distribution

Switch Role

References

- Migrate an ApsaraDB for MongoDB instance across zones in the same region

  Migrate instances to other zones within the same region. For example, you can migrate instances from a single zone to multiple zones. After instances are migrated to other zones, the attributes, specifications, and connection strings of the instances remain unchanged.

- Switch node roles

  You can switch the node roles of an ApsaraDB for MongoDB instance based on your business deployment. This allows your applications to connect to the nearest nodes.

6.5. Migrate an ApsaraDB for MongoDB instance across zones in the same region

This topic describes how to migrate an ApsaraDB for MongoDB instance across zones in the same region. After the instance is migrated, its attributes, specifications, and connection addresses remain unchanged.

Prerequisites

- A replica set or sharded cluster instance is created.

- The destination and source zones are in the same region.
- If the instance is in a VPC, make sure that a vSwitch is created in the destination zone before you start migration. For more information about how to create a vSwitch, see Create a VSwitch.
- The instance does not have a public endpoint. If you have applied for a public endpoint, you must release it before migration. For more information, see Release a public connection string.

Precautions

- If the instance is in a VPC, you cannot change the VPC when the instance is in the migration process.
- The time required varies based on factors such as the network conditions, task queue status, and data volume. We recommend that you migrate the instance across zones during off-peak hours.
- During the migration, a transient connection of 30 seconds occurs. Make sure that your application is configured to reconnect to the instance after it is disconnected.
- The virtual IP addresses (VIPs) of the instance, such as 172.16.88.60, are changed when the instance is migrated across zones. If your application uses the original VIP, the application cannot connect to the instance after the migration.

> **Note** We recommend that you use a connection string URI to connect to the instance, which ensures high availability. For more information, see Overview of replica set instance connections or Overview of sharded cluster instance connections.

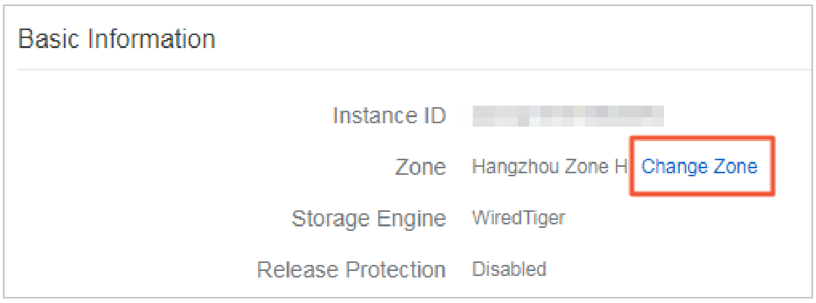
Supported migration types and scenarios

Migration type	Scenario
Migrate an ApsaraDB for MongoDB instance from one zone to another	The ApsaraDB for MongoDB instance is migrated to the zone where an ECS instance resides. Then, the ECS instance can connect to the ApsaraDB for MongoDB instance over an internal network with lower network latency.
Migrate an ApsaraDB for MongoDB instance from one zone to multiple zones	The ApsaraDB for MongoDB instance provides disaster recovery across data centers. The three nodes of a replica set instance are deployed to three different zones in the same region. This enables the instance to tolerate disasters at higher levels. For example, a replica set instance in a single zone can tolerate only server- and rack-level faults, whereas a replica set instance in multiple zones can tolerate server-, rack-, and data center-level faults. **Note** For more information about the node deployment policy of a replica set instance or a sharded cluster instance in multiple zones, see Node deployment policies or Deployment policy for the components in a multi-zone sharded cluster instance.

Migration type	Scenario
Migrate an ApsaraDB for MongoDB instance from multiple zones to one zone	Special user requirements are met.

Migrate a replica set instance across zones in the same region

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the **Basic Information** section, click **Change Zone**.



Basic Information

Instance ID [redacted]

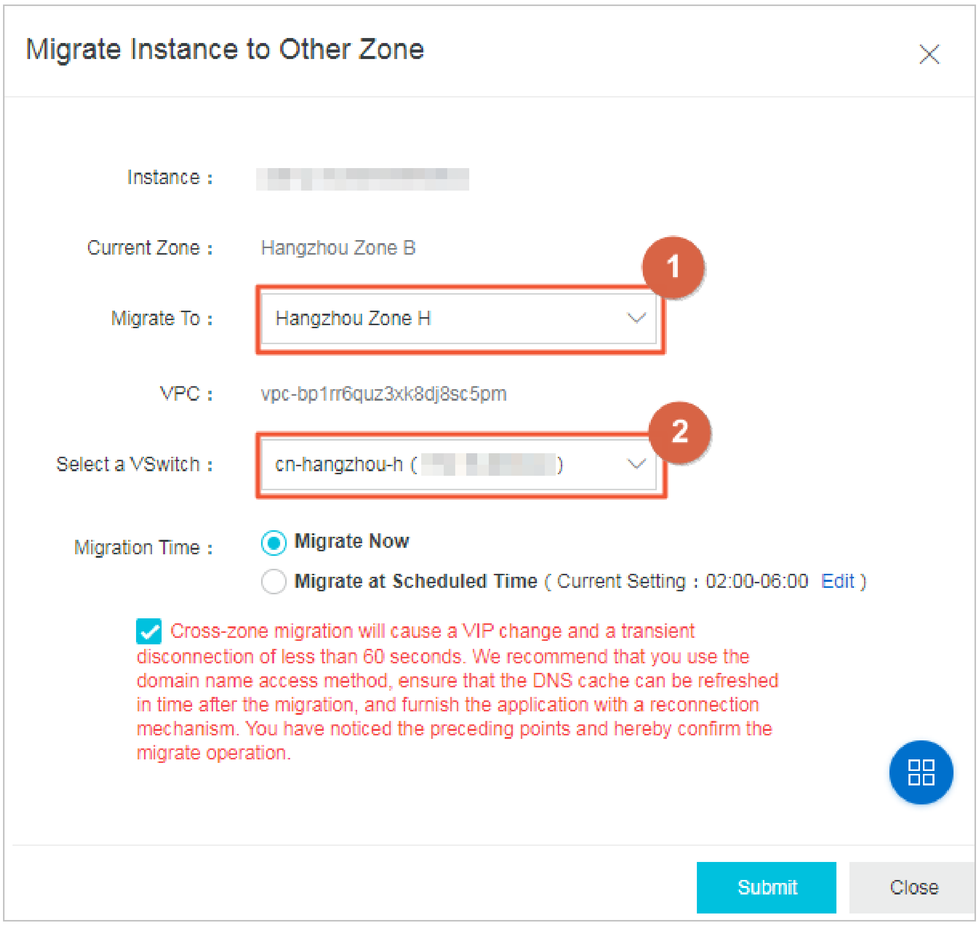
Zone Hangzhou Zone H

Storage Engine WiredTiger

Release Protection Disabled

6. In the Migrate Instance to Other Zone panel, configure parameters based on the network type of the instance.
   - If the instance is in a VPC or is in hybrid network access mode, perform the following operations:

a. Select the destination zone and vSwitch.



Migrate Instance to Other Zone

Instance : [redacted]

Current Zone : Hangzhou Zone B

Migrate To : Hangzhou Zone H

VPC : vpc-bp1rr6quz3xk8dj8sc5pm

Select a VSwitch : cn-hangzhou-h ([redacted])

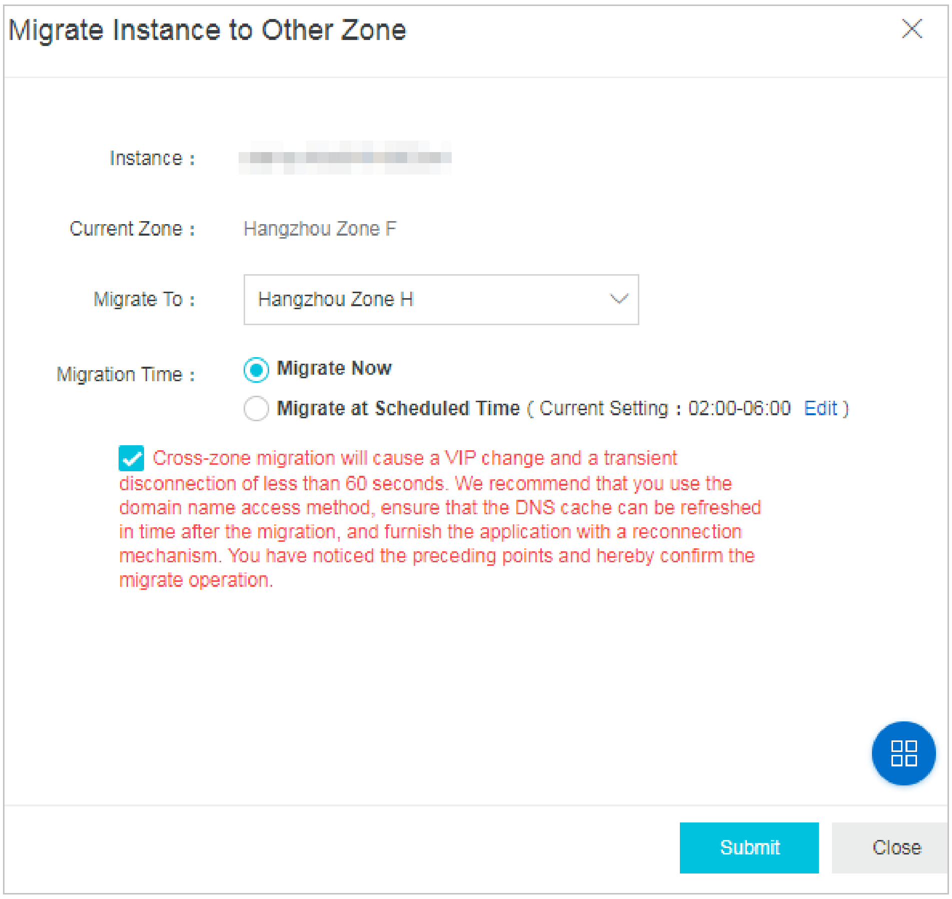
Migration Time : ● Migrate Now ○ Migrate at Scheduled Time (Current Setting : 02:00-06:00 Edit)

☑ Cross-zone migration will cause a VIP change and a transient disconnection of less than 60 seconds. We recommend that you use the domain name access method, ensure that the DNS cache can be refreshed in time after the migration, and furnish the application with a reconnection mechanism. You have noticed the preceding points and hereby confirm the migrate operation.

b. Specify Migration Time and select the check box of the warning message.

- If the instance is in the classic network, perform the following operations:

a. Select the destination zone.



Migrate Instance to Other Zone

Instance : [redacted]

Current Zone : Hangzhou Zone F

Migrate To : Hangzhou Zone H

Migration Time : ● Migrate Now
○ Migrate at Scheduled Time (Current Setting : 02:00-06:00 Edit)

☑ Cross-zone migration will cause a VIP change and a transient disconnection of less than 60 seconds. We recommend that you use the domain name access method, ensure that the DNS cache can be refreshed in time after the migration, and furnish the application with a reconnection mechanism. You have noticed the preceding points and hereby confirm the migrate operation.

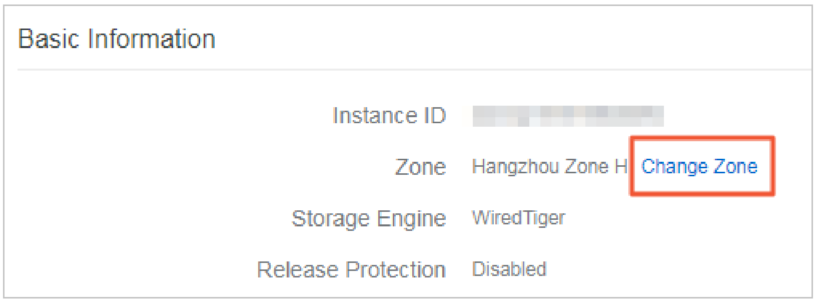
b. Specify Migration Time and select the check box of the warning message.

> Note
>
> - **Migrate Now**: The migration immediately starts. When the instance status changes to **Running**, the migration is complete.
> - **Migrate at Scheduled Time**: The migration starts during the specified period. You can click **Edit** to change the period.
>
>   After you select this option, the system prepares for the migration task, changes the instance status to **Migrating**, and then starts the task in the specified period.

7. Click **Submit**.

Migrate a sharded cluster instance across zones in the same region

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. In the **Basic Information** section, click **Change Zone**.



Basic Information

Instance ID [redacted]

Zone Hangzhou Zone H Change Zone

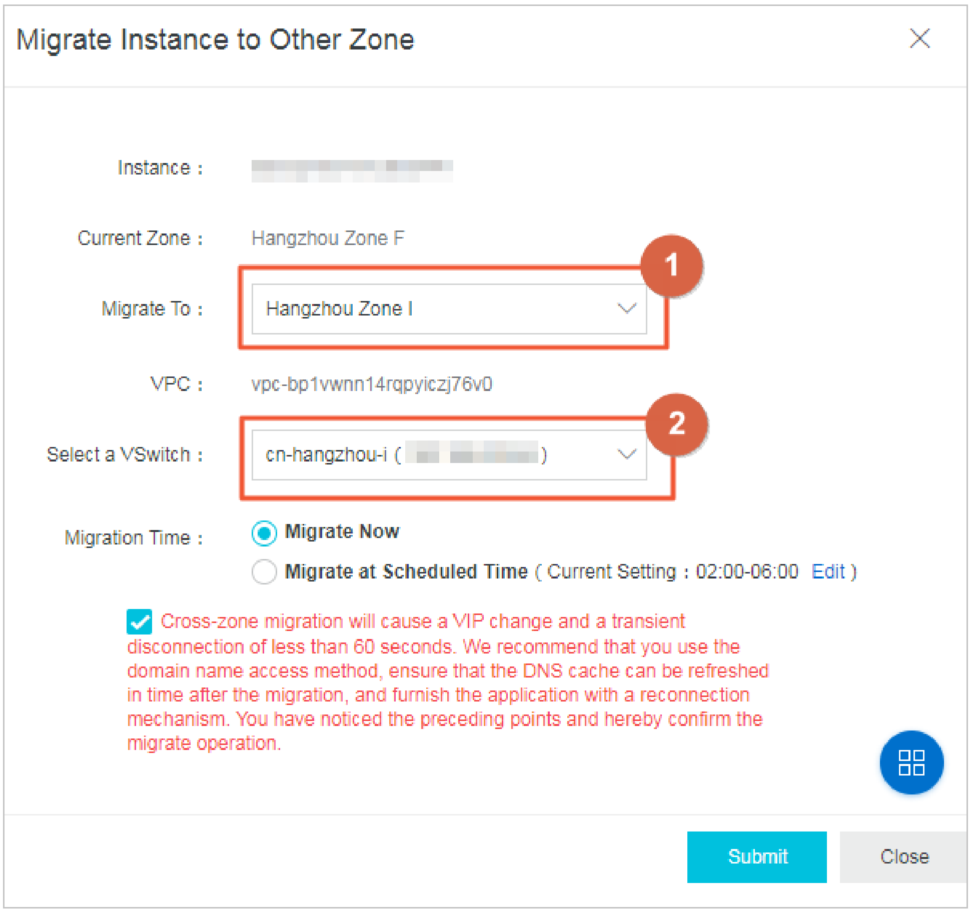
Storage Engine WiredTiger

Release Protection Disabled

6. In the Migrate Instance to Other Zone panel, configure parameters based on the network type of the instance.

   - If the instance is in a VPC or is in hybrid network access mode, perform the following operations:

     a. Select the destination zone and vSwitch.



Migrate Instance to Other Zone

Instance : [redacted]

Current Zone : Hangzhou Zone F

Migrate To : Hangzhou Zone I

VPC : vpc-bp1vwnn14rqpyiczj76v0

Select a VSwitch : cn-hangzhou-i ([redacted])

Migration Time : ● Migrate Now

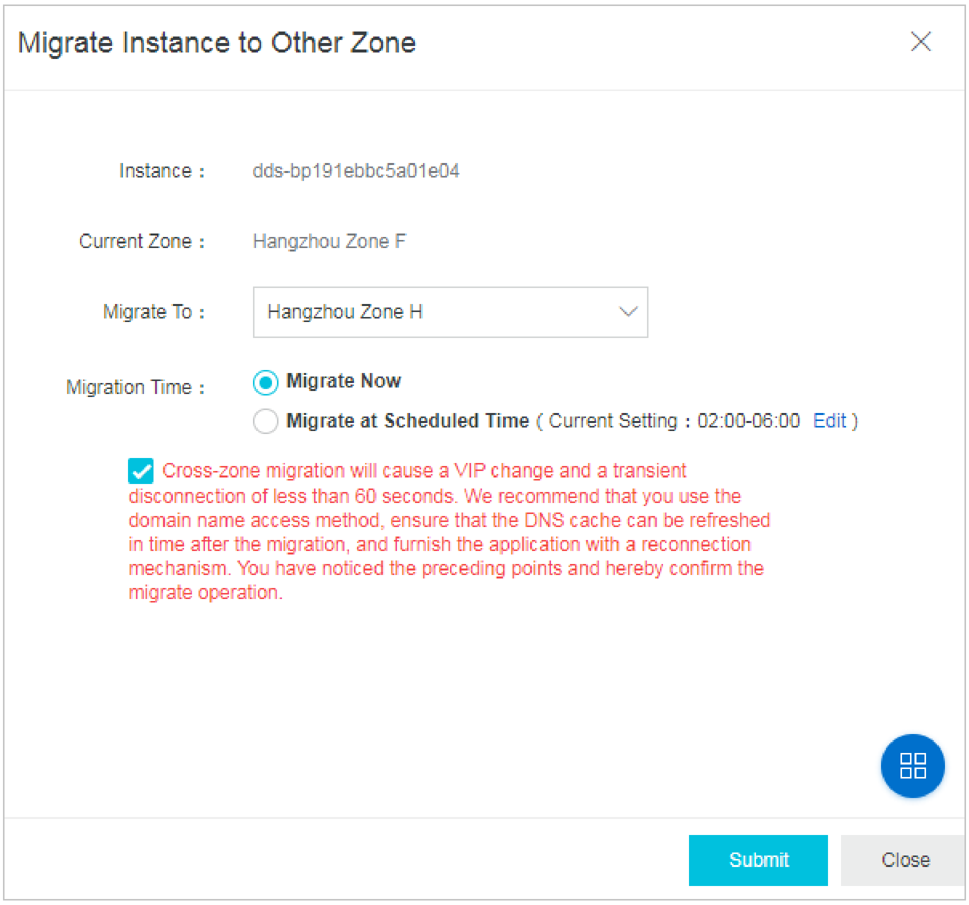
○ Migrate at Scheduled Time (Current Setting : 02:00-06:00 Edit)

☑ Cross-zone migration will cause a VIP change and a transient disconnection of less than 60 seconds. We recommend that you use the domain name access method, ensure that the DNS cache can be refreshed in time after the migration, and furnish the application with a reconnection mechanism. You have noticed the preceding points and hereby confirm the migrate operation.

     b. Specify Migration Time and select the check box of the warning message.

   - If the instance is in the classic network, perform the following operations:

a. Select the destination zone.



Migrate Instance to Other Zone

Instance : dds-bp191ebbc5a01e04

Current Zone : Hangzhou Zone F

Migrate To : Hangzhou Zone H

Migration Time : ● Migrate Now
○ Migrate at Scheduled Time (Current Setting : 02:00-06:00 Edit)

☑ Cross-zone migration will cause a VIP change and a transient disconnection of less than 60 seconds. We recommend that you use the domain name access method, ensure that the DNS cache can be refreshed in time after the migration, and furnish the application with a reconnection mechanism. You have noticed the preceding points and hereby confirm the migrate operation.

b. Specify Migration Time and select the check box of the warning message.

Note

- **Migrate Now**: The migration immediately starts. When the instance status changes to **Running**, the migration is complete.
- **Migrate at Scheduled Time**: The migration starts during the specified period. You can click **Edit** to change the period.

  After you select this option, the system prepares for the migration task, changes the instance status to **Migrating**, and then starts the task in the specified period.

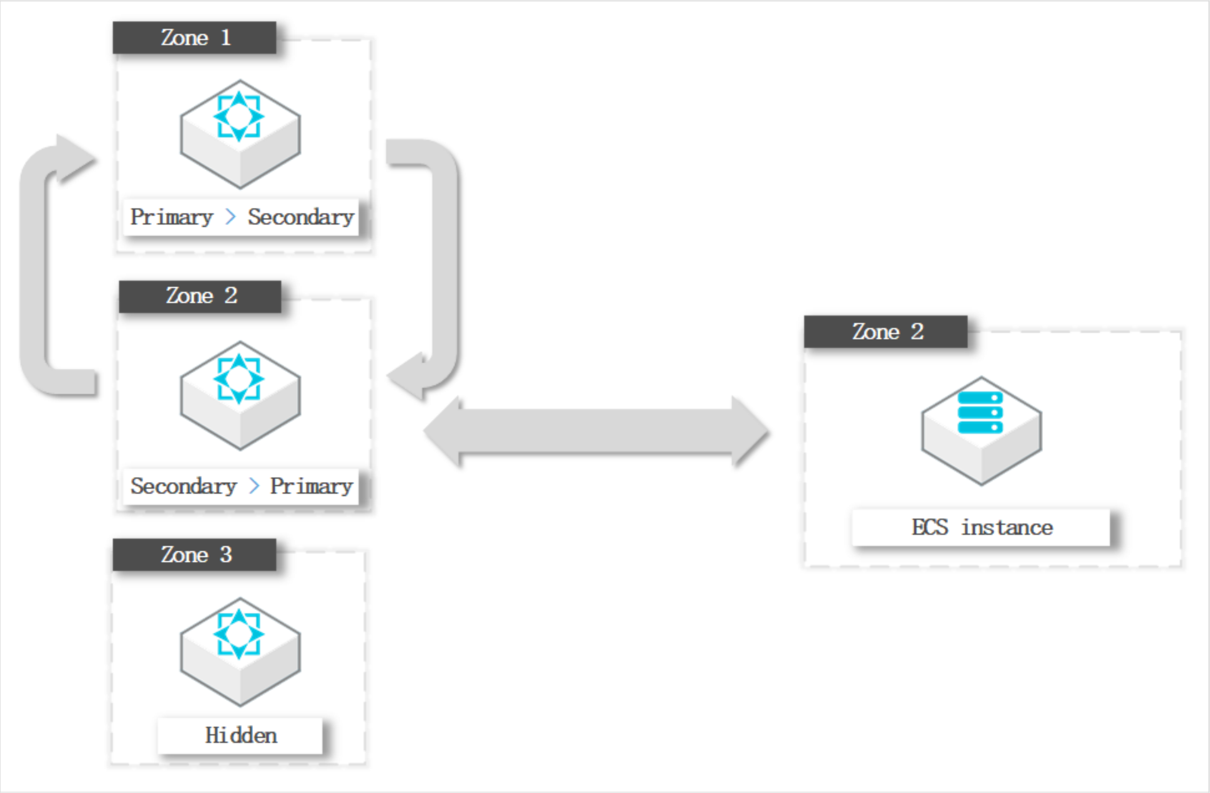
7. Click **Submit**.

6.6. 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 ECS instance and an ApsaraDB for MongoDB instance are in the same zone and connected over the internal network, the latency is minimal. If they are connected across different zones, the latency increases and the performance of ApsaraDB for MongoDB instances and your business will be 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 the roles of 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. Note that only the node roles are changed. ECS and ApsaraDB for MongoDB instances can be connected in the same zone without changing the actual zones and role IDs.

Prerequisites

Replica set or sharded cluster instances must be used.

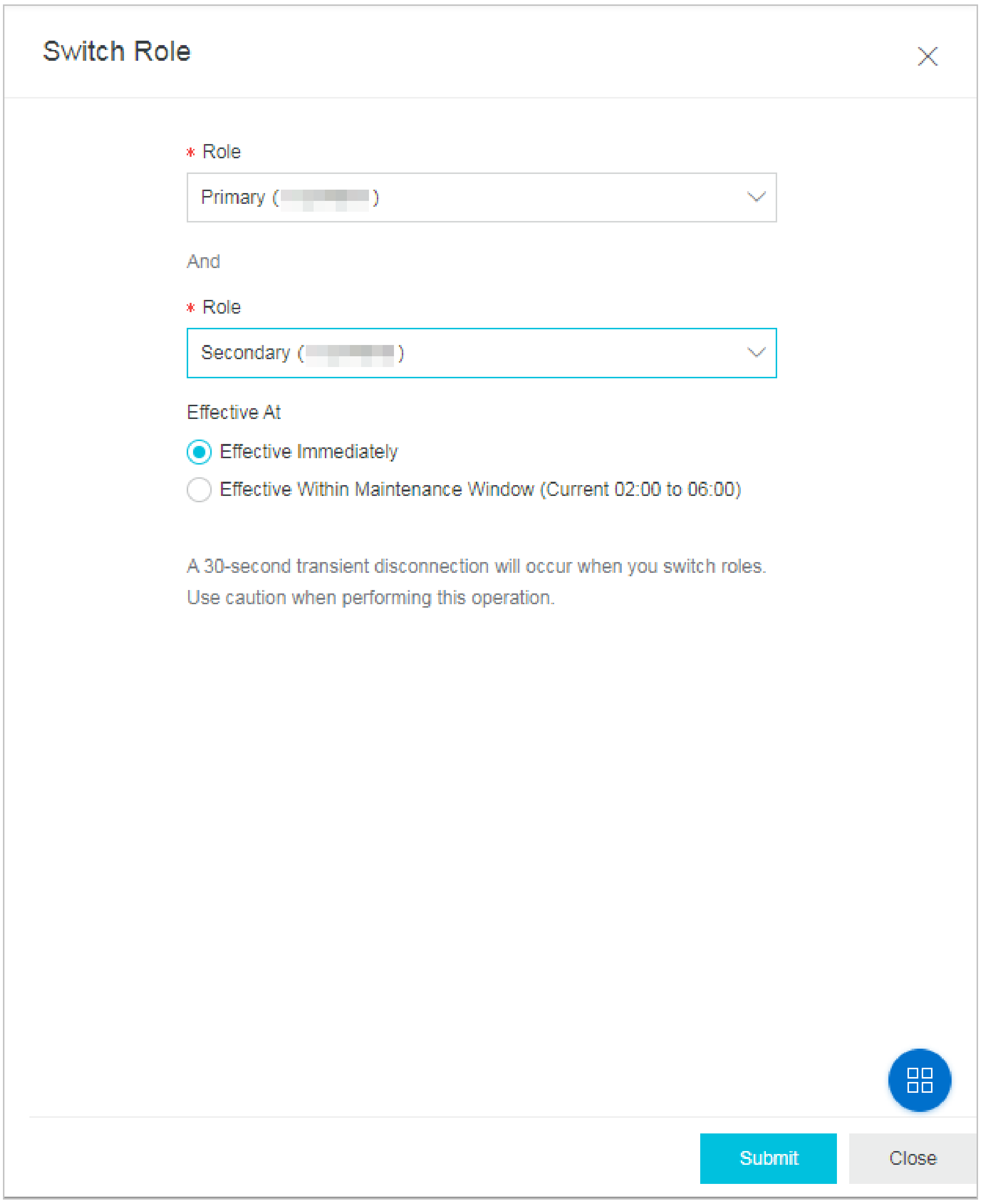
Precautions

- Switching node roles will cause a transient disconnection of up to 30 seconds. Perform this operation during off-peak hours or ensure that your application has a reconnection mechanism.
- Switching node roles only changes the roles of nodes, but not the zones and role IDs of nodes.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Service Availability**.

6. Subsequent steps on the **Service Availability** page vary depending on instance types.
   - Replica set instances
     a. Click **Switch Role** in the upper-right corner of the page.
     b. In the **Switch Role** dialog box that appears, select the roles.



Switch Role

* Role

Primary ([redacted])

And

* Role

Secondary ([redacted])

Effective At

● Effective Immediately

○ Effective Within Maintenance Window (Current 02:00 to 06:00)

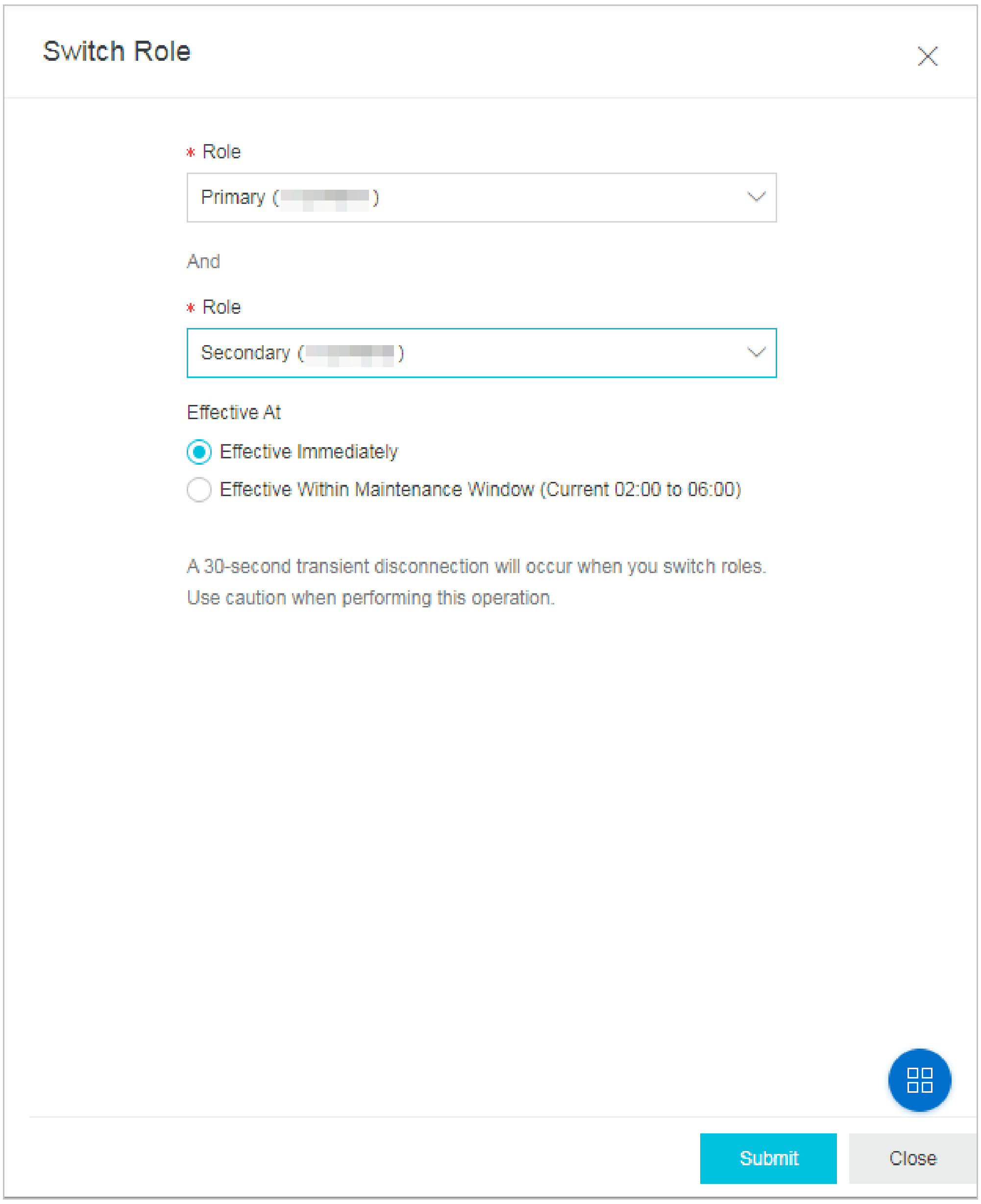
A 30-second transient disconnection will occur when you switch roles. Use caution when performing this operation.

     c. Select the **Effective At**.
        - **Effective Immediately**: The system will switch the roles of the nodes immediately.
        - **Effective Within Maintenance Window**: The system will switch the roles of the nodes during the maintenance period. How to specify a maintenance period, see Specify a maintenance period.
   - Sharded cluster instances

> **Note** For sharded cluster instances, you can only manage 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** dialog box that appears, select the nodes.



Switch Role

* Role

Primary ([redacted])

And

* Role

Secondary ([redacted])

Effective At

● Effective Immediately

○ Effective Within Maintenance Window (Current 02:00 to 06:00)

A 30-second transient disconnection will occur when you switch roles. Use caution when performing this operation.

c. Select the **Effective At**.

- **Effective Immediately**: The system will switch the roles of the nodes immediately.
- **Effective Within Maintenance Window**: The system will switch the roles of the nodes during the maintenance period. How to specify a maintenance period, see Specify a maintenance period.

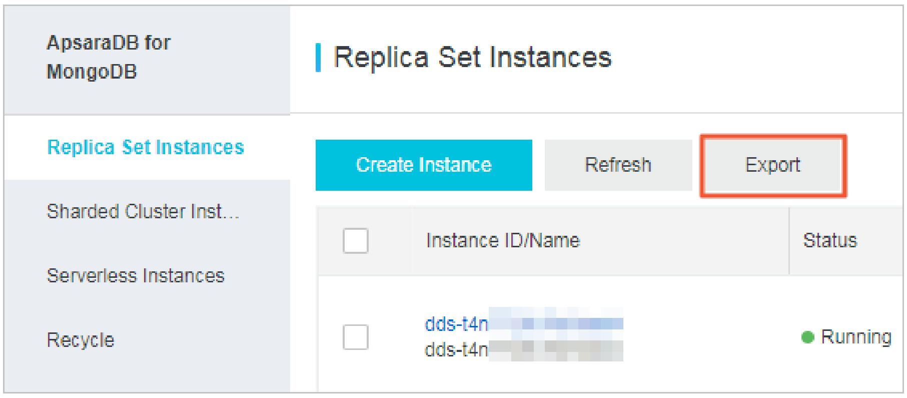
7. Click **Submit**.

6.7. Export the list of instances

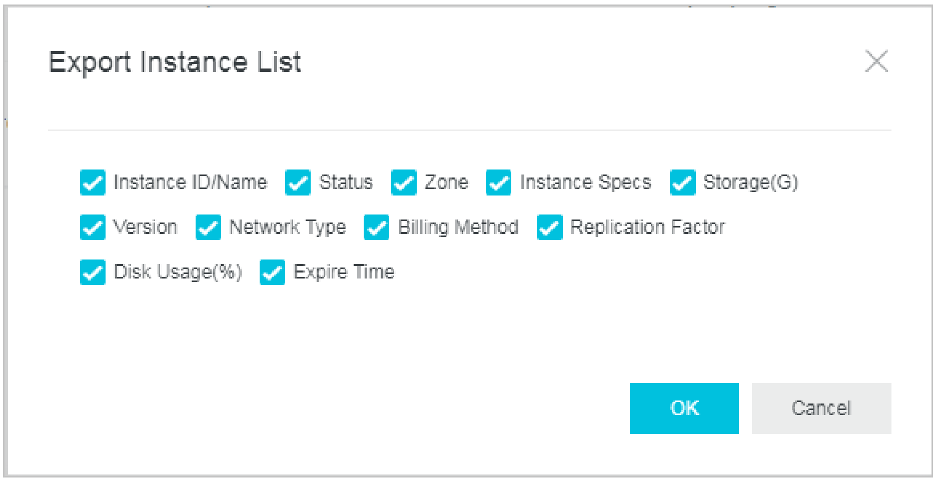
You can export a list of instances of a specified type and region by using the ApsaraDB for MongoDB console to manage cloud instances offline.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region and resource group for the instance.
3. In the left-side navigation pane, click **Replica Set Instances**, **Sharded Cluster Instances**, or **Serverless Instances** based on the instance type.
4. On the Replica Set Instances page of the required instance type, click **Export**.



5. In the **Export Instance List** dialog box, select the instance information you want to export to the list.


Export Instance List

☑ Instance ID/Name ☑ Status ☑ Zone ☑ Instance Specs ☑ Storage(G)
☑ Version ☑ Network Type ☑ Billing Method ☑ Replication Factor
☑ Disk Usage(%) ☑ Expire Time

6. Click **OK**.

> **Note** After you click **OK**, the browser begins to download the CSV file. You can use Excel or a text editor to view this file.

6.8. Release an ApsaraDB for MongoDB instance

This topic describes how to manually release an ApsaraDB for MongoDB instance that uses pay-as-you-go billing. After an instance is released, its data cannot be restored.

Prerequisites

The billing method of the instance is pay-as-you-go.

> **Note** You cannot manually release a subscription instance. A subscription instance is automatically released when it expires.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find your instance and choose

   ⋮

   > **Release** in the **Actions** column.

   Alternatively, you can click the ID of your instance. On the **Basic Information** page that appears, click **Release**.
5. In the **Release Instance** message that appears, click **OK**.

6.9. Restart an ApsaraDB for MongoDB instance

This topic describes how to restart an ApsaraDB for MongoDB instance when the number of connections exceeds the upper limit or the performance of the instance deteriorates.

Precautions

- When you restart an instance, all its nodes are restarted in turn and each node has a brief disconnection of about 30 seconds. If the instance houses more than 10,000 collections, the brief disconnections last longer. Therefore, we recommend that you restart an instance during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected.
- When you restart a replica set instance, a primary/secondary switchover may occur and cause the roles of connected nodes to change. We recommend that you use a connection string URI to connect to the instance to avoid impact on the read/write operations of your application. For more information, see Overview of replica set instance connections.
- You can restart a sharded cluster instance, or a mongos or shard in the instance. This kind of nodes are inaccessible until restarted.

Restart an instance

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.

4. Find the target instance and choose ⋮ > **Restart** in the **Actions** column.

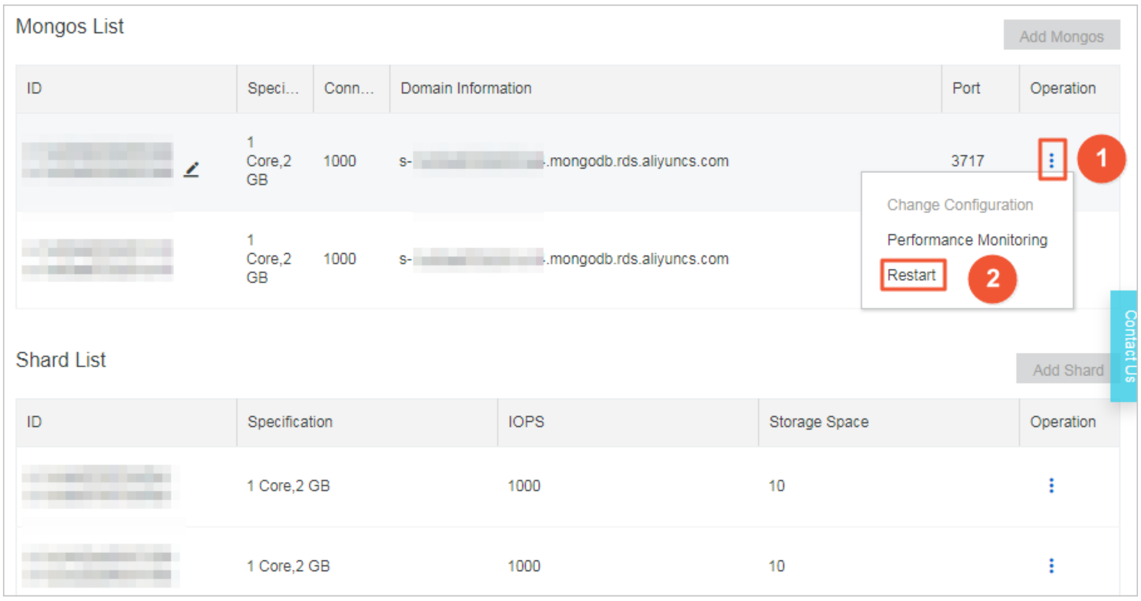
> **Note** Alternatively, you can click the ID of the target instance. On the **Basic Information** page that appears, click **Restart Instance** in the upper-right corner.

5. In the **Restart Instance** message that appears, click **OK**.
   The instance status changes to **Rebooting**. When the instance enters the **Running** state, the restart is complete.

Restart a node in a sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. Follow these steps to restart a node:
   - Restart a mongos.

     In the **Mongos List** section, find the target mongos and choose ⋮ > **Restart** in the Operation column.



Mongos List

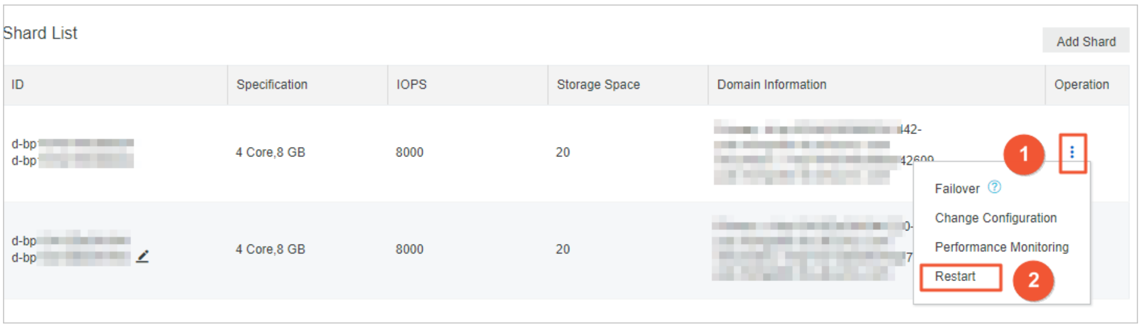
ID	Speci...	Conn...	Domain Information	Port	Operation
[redacted]	1 Core,2 GB	1000	s-[redacted].mongodb.rds.aliyuncs.com	3717	⋮
[redacted]	1 Core,2 GB	1000	s-[redacted].mongodb.rds.aliyuncs.com		

Shard List

ID	Specification	IOPS	Storage Space	Operation
[redacted]	1 Core,2 GB	1000	10	⋮
[redacted]	1 Core,2 GB	1000	10	⋮

   - Restart a shard.

     In the **Shard List** section, find the target shard and choose ⋮ > **Restart** in the Operation column.



Shard List

ID	Specification	IOPS	Storage Space	Domain Information	Operation
d-bp[redacted] d-bp[redacted]	4 Core,8 GB	8000	20	[redacted]	⋮
d-bp[redacted] d-bp[redacted]	4 Core,8 GB	8000	20	[redacted]	

6. In the **Restart Node** message that appears, click **OK**.
   The instance status changes to **Rebooting**. When the instance enters the **Running** state, the restart is complete.

7.Tag management

7.1. Create a tag

You can create and bind multiple tags to a large number of instances to classify and filter instances by tag.

Prerequisites

You must log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account. RAM users do not have permissions to manage or use tags.

Precautions

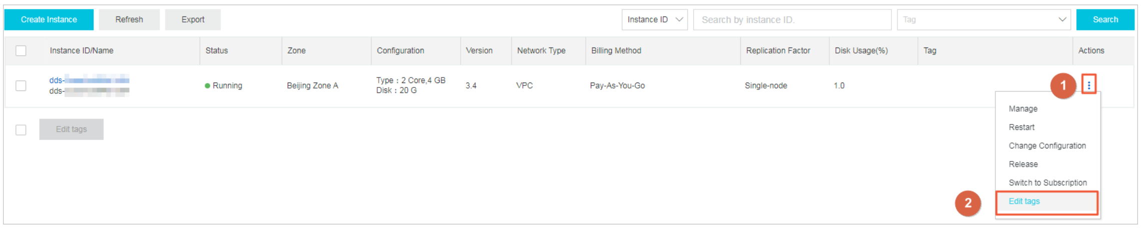
- A tag consists of a key-value pair. The key must be unique in the same region of the same account, while the value is not limited.

  > **Note** A key can have zero to multiple values.

- You can edit tags for a maximum of 50 instances at a time.
- You can bind up to 20 tags to each instance.
- You can bind or unbind up to 20 tags at a time.

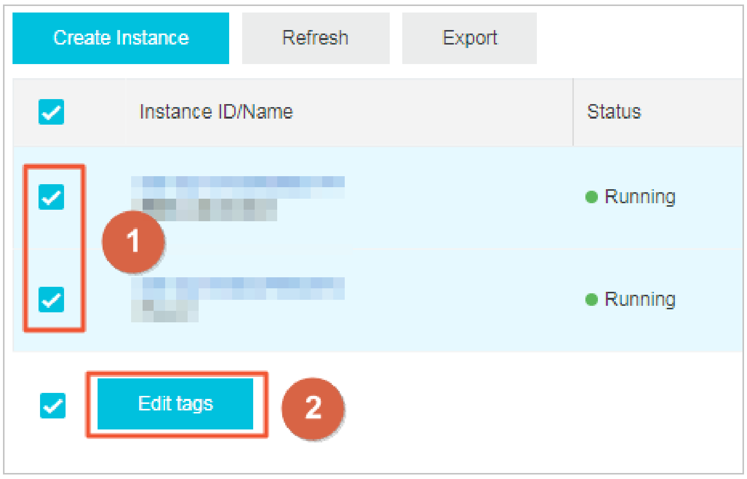
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. To create a tag, perform the following steps:
   - Create tags for an instance: Find the instance. Choose ⋮ > **Edit tags** in the **Actions** column.



□	Instance ID/Name	Status	Zone	Configuration	Version	Network Type	Billing Method	Replication Factor	Disk Usage(%)	Tag	Actions
□	dds-[redacted] dds-[redacted]	● Running	Beijing Zone A	Type：2 Core,4 GB Disk：20 G	3.4	VPC	Pay-As-You-Go	Single-node	1.0		⋮

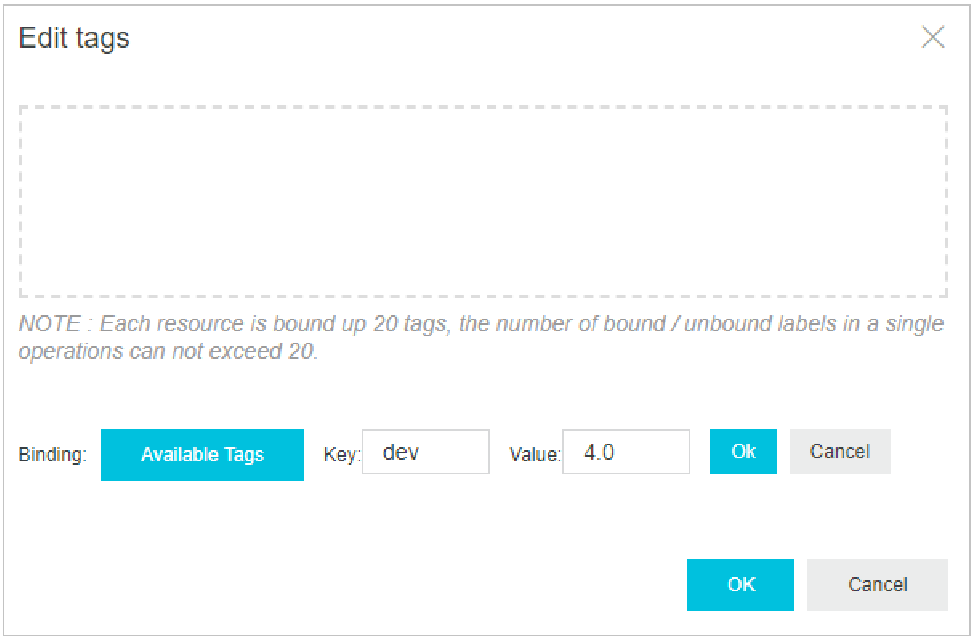
   - Create tags for multiple instances: Select the instances for which you want to create tags and then click **Edit tags** at the lower part of the page.



5. In the dialog box that appears, click **Create**.

> Note If you have created tags, click **Available Tags** to bind the tags to the instances. For more information, see Bind existing tags.

6. Set the key and value of the tag and then click **Ok**.



NOTE : Each resource is bound up 20 tags, the number of bound / unbound labels in a single operations can not exceed 20.

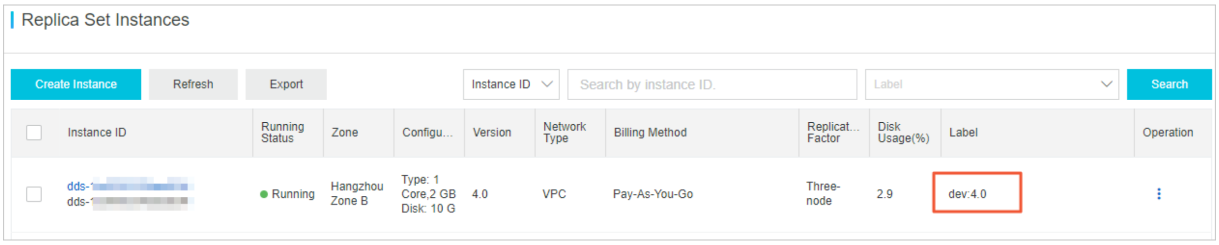
Binding: Key: dev Value: 4.0

7. Repeat the preceding steps to create all the tags. Then click **OK** in the lower-right corner of the dialog box.

> Note After you create tags, you can bind them to other instances.

Result

After you create a tag, you can view the tag information of the instance in the instance list.



Replica Set Instances

□	Instance ID	Running Status	Zone	Configu...	Version	Network Type	Billing Method	Replicat... Factor	Disk Usage(%)	Label	Operation
□	dds-[redacted] dds-[redacted]	● Running	Hangzhou Zone B	Type: 1 Core,2 GB Disk: 10 G	4.0	VPC	Pay-As-You-Go	Three-node	2.9	dev:4.0	⋮

7.2. Bind existing tags

After you create tags, you can bind different tags to instances and filter instances by tag.

Prerequisites

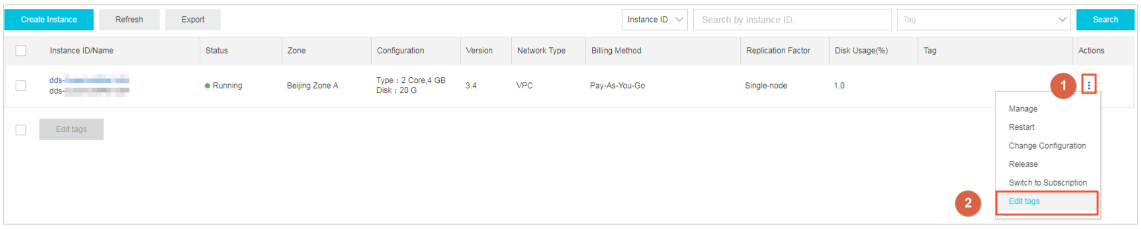
- You must log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account. RAM users do not have permissions to manage or use tags.
- You have performed the Create a tag operation.

Precautions

- You can select a maximum of 50 instances to bind tags at a time.
- You can bind up to 20 tags to each instance.
- You can bind or unbind up to 20 tags at a time.

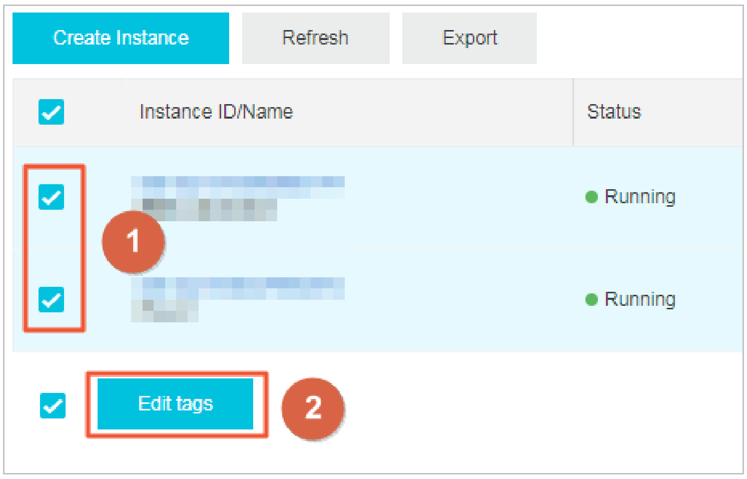
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Perform the following operations to bind a tag:
   - Bind tags to an instance: Find the instance. In the **Actions** column, choose ⋮ **> Edit tags**.

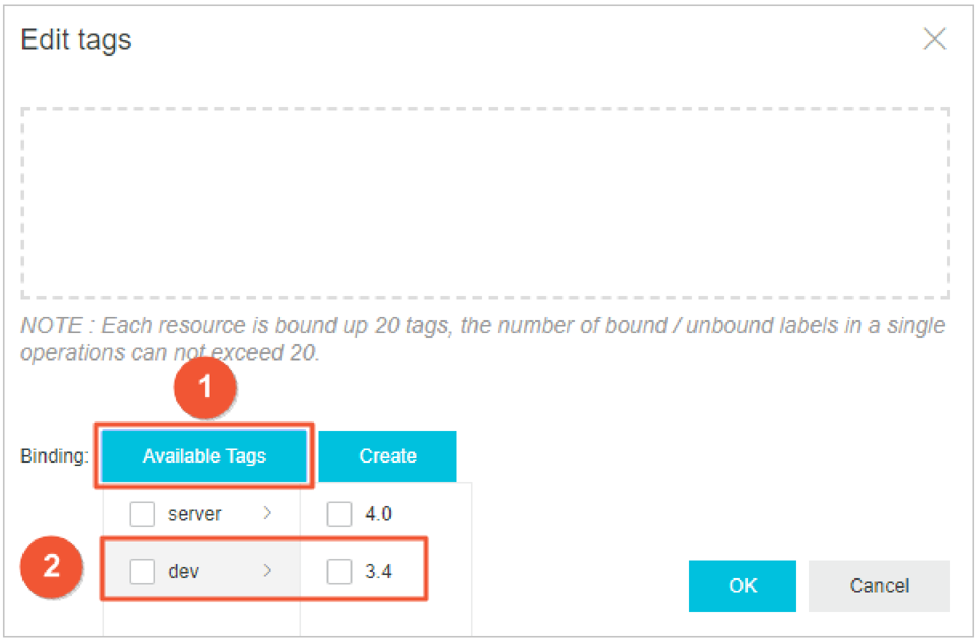


□	Instance ID/Name	Status	Zone	Configuration	Version	Network Type	Billing Method	Replication Factor	Disk Usage(%)	Tag	Actions
□	dds-[redacted] dds-[redacted]	● Running	Beijing Zone A	Type : 2 Core,4 GB Disk : 20 G	3.4	VPC	Pay-As-You-Go	Single-node	1.0		⋮

   - Bind tags to multiple instances at a time: Select the instances to which you want to bind tags and then click **Edit tags** at the bottom of the instance list.



5. In the dialog box that appears, click **Available Tags**. Then, select the desired tag information.


Edit tags
NOTE : Each resource is bound up 20 tags, the number of bound / unbound labels in a single operations can not exceed 20.
1
Binding:
Available Tags
Create
server
4.0
2
dev
3.4
OK
Cancel

> **Note** If tag information is not displayed after you click **Available Tags**, you must Create a tag first.

6. Click **OK**.

7.3. Filter instances by tag

After you bind tags to ApsaraDB for MongoDB instances, you can filter instances by tag in the instance list to quickly find instances of the specified category.

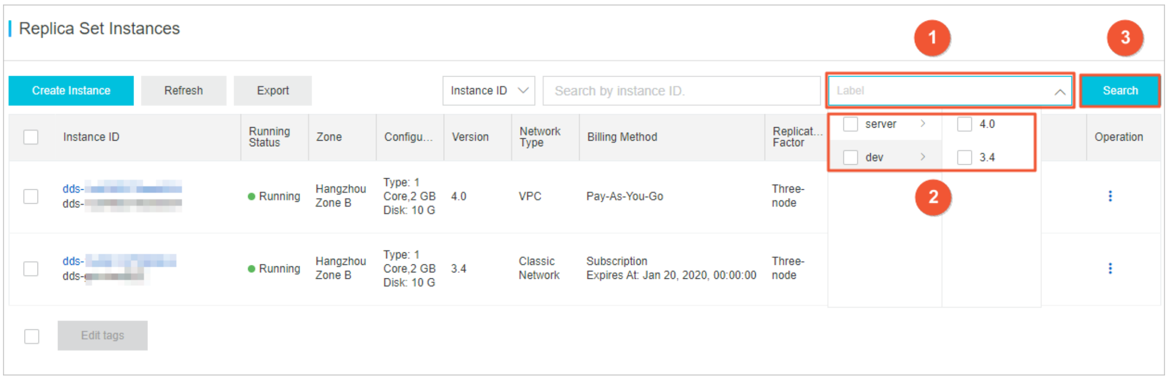
Prerequisites

You must log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account. RAM users do not have permissions to manage or use tags.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Click the Label search box, select the tag key or tag value to be filtered, and then click **Search**.



	Instance ID	Running Status	Zone	Configu...	Version	Network Type	Billing Method	Replicat... Factor	Operation
□	dds-[redacted] dds-[redacted]	● Running	Hangzhou Zone B	Type: 1 Core,2 GB Disk: 10 G	4.0	VPC	Pay-As-You-Go	Three-node	⋮
□	dds-[redacted] dds-[redacted]	● Running	Hangzhou Zone B	Type: 1 Core,2 GB Disk: 10 G	3.4	Classic Network	Subscription Expires At: Jan 20, 2020, 00:00:00	Three-node	⋮

Note

- After you create a tag or update an existing tag, you need to refresh the page to see the new tag in the tag list.
- To clear the filter criteria, click ☒.

7.4. Unbind or delete a tag

When a tag is no longer needed in an ApsaraDB for MongoDB instance, you can unbind the tag from the instance. If the tag is not bound to any other instance, it will be deleted.

Prerequisites

You must log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account. RAM users do not have permissions to manage or use tags.

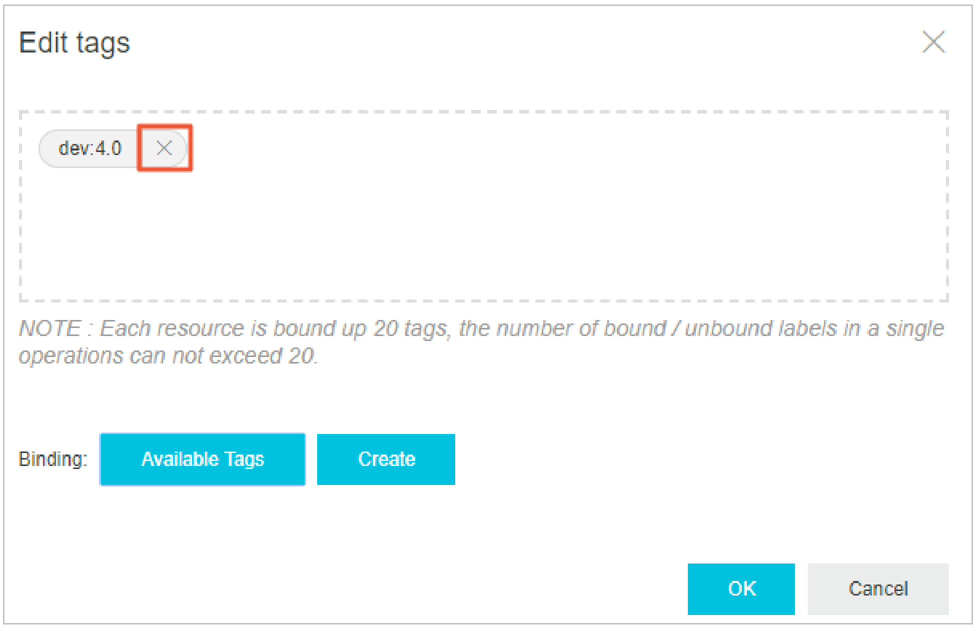
Precautions

- You can unbind up to 20 tags at a time.
- When a tag is unbound from all instances, it is automatically deleted.
- Unbinding a tag does not affect the normal operation of the instance. After all tags of an instance are unbound, the instance cannot be filtered by tag.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the instance. Choose ⋮ > **Edit tags** in the **Actions** column.

5. In the dialog box that appears, click ×.



Edit tags

dev:4.0 ×

NOTE : Each resource is bound up 20 tags, the number of bound / unbound labels in a single operations can not exceed 20.

Note To delete a tag, unbind the tag from all instances.

6. Click **OK**.

8.Network connection management

8.1. Connection String of a Shard or Configserver Node

8.1.1. Apply for a connection string of a shard or Configserver node

A sharded cluster instance consists of Mongos, shard, and Configserver nodes. Typically, you need only to connect to the Mongos node to read and write data. In some special scenarios such as data synchronization between clusters, you must read the oplog of a shard node or the configuration information of a Configserver node. You can apply for a connection string of the corresponding node.

Prerequisites

Sharded cluster instances must be used.

Usage notes

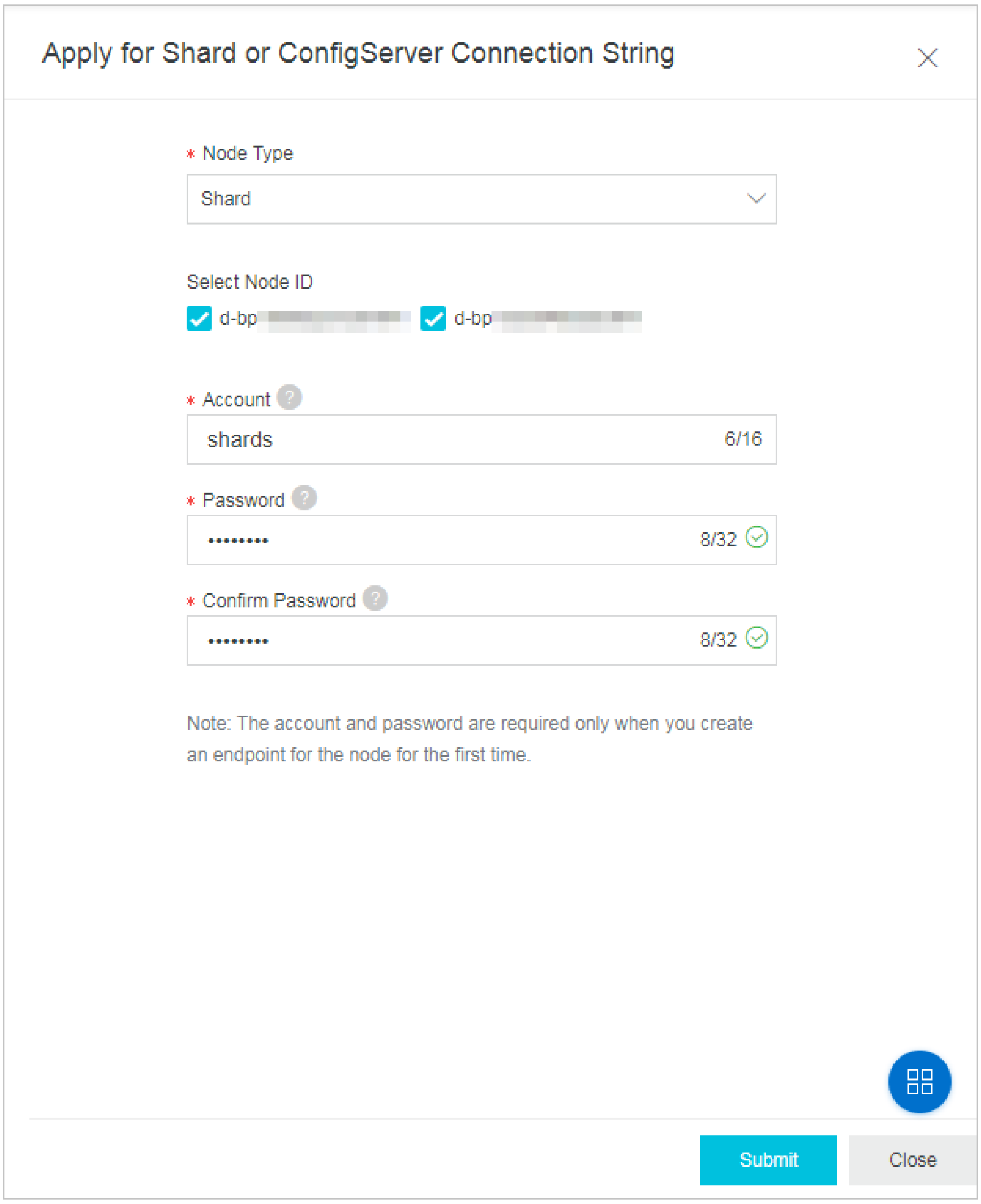
- After you apply for a connection string, two connection strings are allocated to each node, one for the primary node and one for the secondary node.
- The network type of the connection strings must be the same as that of the current Mongos node.
- You cannot modify the connection string of a shard or Configserver node.
- The connection strings allocated here can only be used to access the node over the internal network. If you want to access the node over the Internet, you can apply for a public endpoint for a sharded cluster instance. For more information, see Apply for a public endpoint for a sharded cluster instance.

Introduction to the sharded cluster architecture and nodes

For more information, see Architecture of sharded cluster instances.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the upper-right corner, choose **More > Apply for Shard or ConfigServer Connection String**.
7. In the dialog box that appears, apply for a connection string for the shard or Configserver node.



Apply for Shard or ConfigServer Connection String

* Node Type

Shard

Select Node ID

☑ d-bp[redacted] ☑ d-bp[redacted]

* Account

shards 6/16

* Password

•••••••• 8/32

* Confirm Password

•••••••• 8/32

Note: The account and password are required only when you create an endpoint for the node for the first time.

Parameter	Description
Node Type	◦ **Shard**: the shard node. ◦ **CS**: the Configserver node.
Select Node ID	Select a check box corresponding to the ID of the node for which you want to create a connection string.

Parameter	Description
Account	The account name must be 4 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter. **Note** ○ You must set the account and password only when you apply for the connection string of a shard or Configserver node for the first time. The account and password are required for all shard and Configserver nodes. ○ The permissions of this account are fixed to read-only.
Password	○ 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. **Note** If you forget your password, you can reset the password. For more information, see Reset the password for an ApsaraDB for MongoDB instance.
Confirm Password	Enter the account password again.

8. Click **Submit**.
9. Wait until the instance status changes from **Creating Connection** to **Running**.

Node types

After you apply for the connection string of a shard or Configserver node, you can view the connection string on the **Database Connection** page. The following table describes node types.

Node type	Description
Shard	The shard node.
CS	The Configserver node.
Mongos	The mongos node.

References

If the connection string of a shard or Configserver node is no longer needed, you can Release the connection string of a shard or Configserver node.

8.1.2. Release the connection string of a shard or Configserver node

When you no longer need to connect to a shard or Configserver node, you can release its connection string.

Precautions

- The connection string of a Mongos node cannot be released.
- After the connection string of a shard or Configserver node is released, the connection strings of the primary and secondary nodes are released and you cannot use this connection string to connect to the released node. Proceed with caution.
- This operation releases the internal connection string of the node. If the node also has a public connection string and the connection string is no longer needed, you can Release a public connection string.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. Find the node and click **Release** in the **Actions** column.

   > **Note** **Shard** indicates a shard node. **CS** indicates a Configserver node.

7. In the dialog box that appears, click **OK**.
8. Wait until the instance status changes from **Releasing Network Connection** to **Running**.

8.2. Public IP Connection

8.2.1. Apply for a public endpoint for an ApsaraDB for MongoDB instance

This topic describes how to apply for a public endpoint for an ApsaraDB for MongoDB instance so that you can connect to the instance over the Internet.

Apply for a public endpoint for a standalone instance

Apply for a public endpoint for a replica set instance

Apply for a public endpoint for a sharded cluster instance

References

Connect to an ApsaraDB for MongoDB instance over the Internet

8.2.2. Release a public connection string

To ensure data security, you can release a public connection string that is no longer needed in the console.

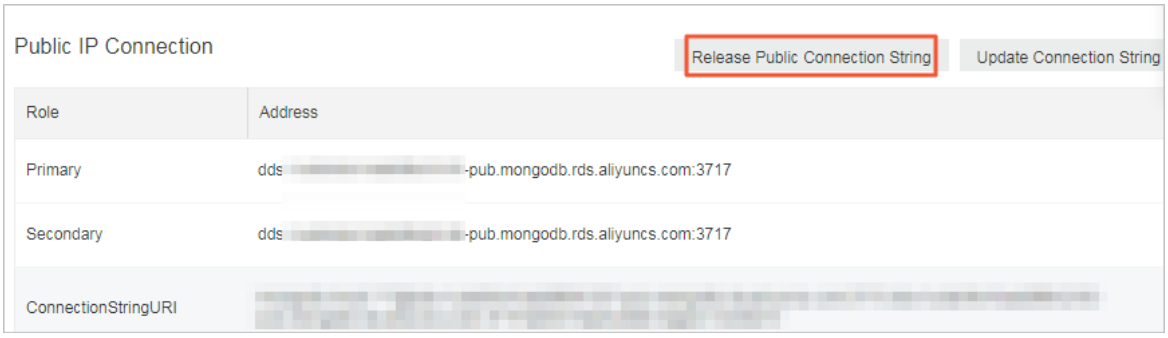
Precautions

- You can release one or more public connection strings of the Mongos, shard, and Configserver nodes for a sharded cluster instance.
- After the public connection string is released for an instance or node, you cannot connect to the instance or node through the original public connection string.
- After the public connection string 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 or an ECS security group for an ApsaraDB for MongoDB instance.

Standalone and replica set instances

> ? **Note** After the public connection string of a replica set instance is released, the public connection strings of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Public IP Connection** section, click **Release Public Connection String**.



Public IP Connection

Role	Address
Primary	dds[redacted]-pub.mongodb.rds.aliyuncs.com:3717
Secondary	dds[redacted]-pub.mongodb.rds.aliyuncs.com:3717
ConnectionStringURI	[redacted]

7. In the dialog box that appears, click **OK**.

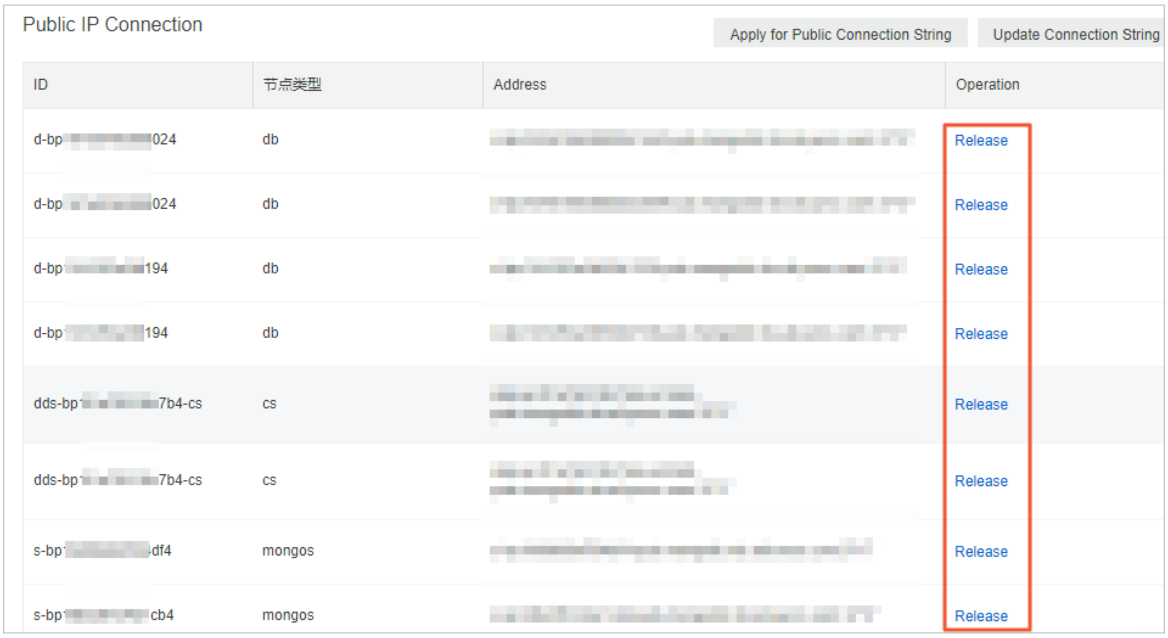
Sharded cluster instances

You can release one or more public connection strings of the Mongos, shard, and Configserver nodes for a sharded cluster instance.

Note

- For more information about node types, see Architecture of sharded cluster instances.
- After the public connection string of a shard or Configserver node is released, the public connection strings of the primary and secondary nodes are released.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Public IP Connection** section, find the Mongos, shard, or Configserver node for which you want to release the public connection string.
7. In the **Actions** column corresponding to the instance, click **Release**.



Public IP Connection

ID	节点类型	Address	Operation
d-bp[redacted]024	db	[redacted]	Release
d-bp[redacted]024	db	[redacted]	Release
d-bp[redacted]194	db	[redacted]	Release
d-bp[redacted]194	db	[redacted]	Release
dds-bp[redacted]7b4-cs	cs	[redacted]	Release
dds-bp[redacted]7b4-cs	cs	[redacted]	Release
s-bp[redacted]df4	mongos	[redacted]	Release
s-bp[redacted]cb4	mongos	[redacted]	Release

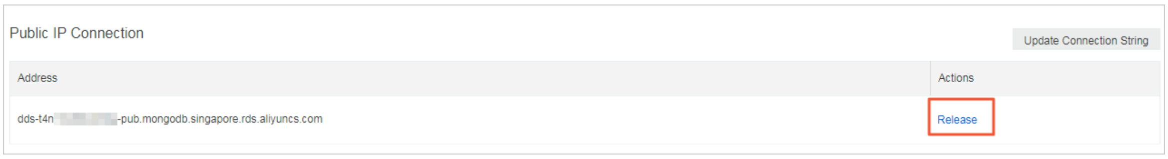
Node type	Description
db	The shard node.
cs	The Configserver node.
mongos	The Mongos node.

Note You can repeat this step to release the public connection strings of other nodes as needed. To release the public connection string of the next node, you must wait until the public connection string of the current node is released or the status of the current node becomes **Running**.

8. In the dialog box that appears, click **OK**.

Serverless instances

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Serverless Instances**
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Public IP Connection** section, click **Release** under the **Actions** column.



Public IP Connection

Address	Actions
dds-t4n[redacted]-pub.mongodb.singapore.rds.aliyuncs.com	Release

7. In the dialog box that appears, click **OK**.

8.3. Enable or disable password-free access for an ApsaraDB for MongoDB instance

This topic describes how to enable or disable password-free access over a VPC for an ApsaraDB for MongoDB instance. This makes database connections easy and secure. After password-free access is enabled, the ECS instance that shares the same VPC with the ApsaraDB for MongoDB instance can connect to a database of this ApsaraDB for MongoDB instance without a password. You can still use a database username and its password to connect to this database.

Prerequisites

- The instance is a replica set or sharded cluster instance.
- The database version of the instance is 4.0 (with the minor version of mongodb_20190408_3.0.11 or later). If the version is earlier than the required version, upgrade the instance. For more information, see Upgrade MongoDB versions and Upgrade the minor version of an ApsaraDB for MongoDB instance.

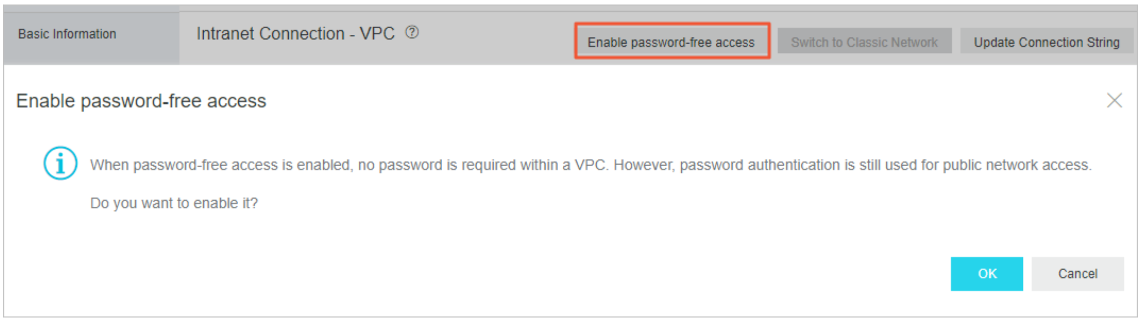
  > **Note** You can view the database version and the minor version on the **Basic Information** page in the ApsaraDB for MongoDB console.

- The instance is in a VPC. If the network type is Classic Network, switch it to VPC. For more information, see Switch from Classic Network to VPC.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.

3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the upper-right corner of the **Intranet Connection - VPC** section, click **Enable password-free access** or **Disable password-free access**.
   - Enable password-free access.


Basic Information
Intranet Connection - VPC
Enable password-free access
Switch to Classic Network
Update Connection String
Enable password-free access
OK
Cancel

When password-free access is enabled, no password is required within a VPC. However, password authentication is still used for public network access.

Do you want to enable it?

After password-free access is enabled, the ECS instance that shares the same VPC with the ApsaraDB for MongoDB instance can connect to a database of this ApsaraDB for MongoDB instance without a password. You can still use a database username and its password to connect to this database.

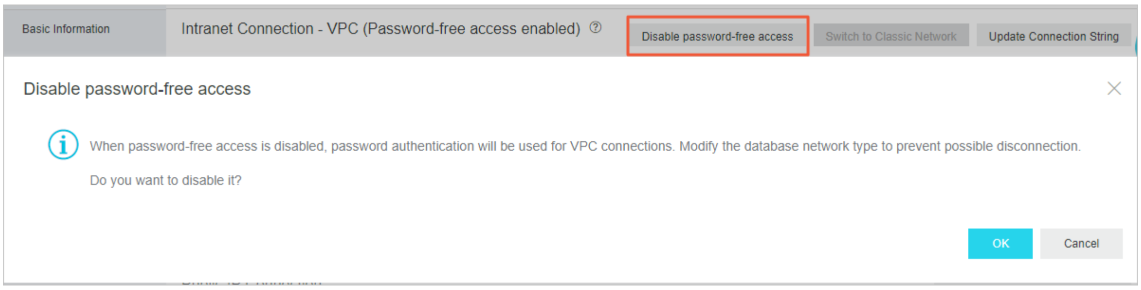
> **Note** If you want to connect to a database of an ApsaraDB for MongoDB instance without entering a password, add the IP address of your client to a whitelist of this instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

The following command provides an example of a password-free connection by using the mongo shell:

```
mongo --host dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717
```

   - Disable password-free access.

> **Note** After password-free access is disabled, the applications that have established connections with a database of this ApsaraDB for MongoDB instance are disconnected. You must change the database connection mode for your application before you disable password-free access.


Basic Information
Intranet Connection - VPC (Password-free access enabled)
Disable password-free access
Switch to Classic Network
Update Connection String
Disable password-free access
OK
Cancel

When password-free access is disabled, password authentication will be used for VPC connections. Modify the database network type to prevent possible disconnection.

Do you want to disable it?

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

Related operations

Operation	Description
ModifyInstanceVpcAuthMode	Enables or disables password-free access over a VPC for an ApsaraDB for MongoDB instance.

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

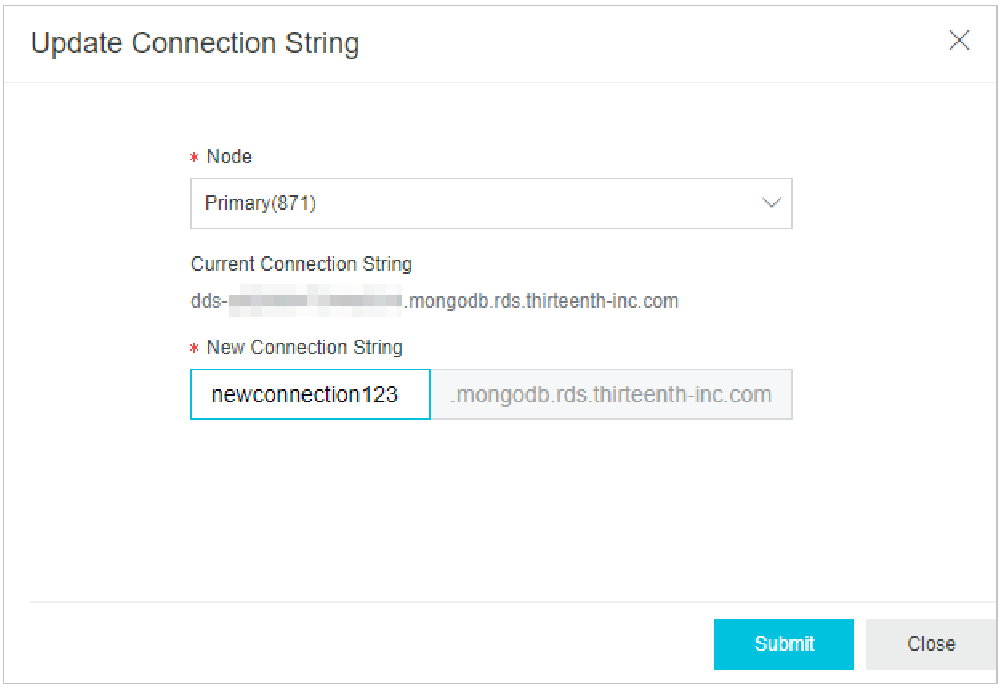
This topic describes how to modify the public and internal endpoints of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console.

Limits

Instance	Limit
Standalone instances	You can only modify the public and internal endpoints of a primary node.
Replica set instances	You can modify the public and internal endpoints of both primary and secondary nodes.
Sharded cluster instances	You can only modify the public and internal endpoints of a mongos.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Intranet Connection** or **Public IP Connection** section, click **Update Connection String**.
7. In the dialog box that appears, enter a new endpoint.



Update Connection String

* Node

Primary(871)

Current Connection String

dds-[redacted].mongodb.rds.thirteenth-inc.com

* New Connection String

newconnection123 .mongodb.rds.thirteenth-inc.com

Note

- You can only modify the prefix of the endpoint.
- The new endpoint must be 8 to 64 characters in length and can contain lowercase letters, digits, and hyphens (-). It must start with a lowercase letter and end with a lowercase letter or digit.

8. 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.

8.5. Switch the network type of an ApsaraDB for MongoDB instance

This topic describes how to switch the network type of an ApsaraDB for MongoDB instance between Classic Network and VPC in the ApsaraDB for MongoDB console.

Prerequisites

The instance is a replica set or sharded cluster instance.

Note For a standalone instance, its network type is always VPC and cannot be changed.

Precautions

Switching the network type of an instance causes a brief disconnection of the instance. We recommend that you perform this operation during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected. This protects your business against the brief disconnection.

> ? **Note** You can choose to retain the internal endpoints on the classic network. This way, you can switch the network type without a brief disconnection. For more information, see Configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC.

Internal connection addresses

- Intranet Connection - Classic Network: Cloud services on a classic network are not isolated. Unauthorized access can only be blocked by the security groups or whitelists of the cloud services.
- Intranet Connection - VPC: A VPC is an isolated virtual network with better security and performance than a classic network. By default, an ApsaraDB for MongoDB instance provides internal endpoints on a VPC.

Switch from Classic Network to VPC

1. Create a VPC in the same region as the target ApsaraDB for MongoDB instance. For more information, see Create a VPC.
2. Log on to the ApsaraDB for MongoDB console.
3. In the upper-left corner of the page, select the resource group and the region of the target instance.
4. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
5. Find the target instance and click its ID.
6. In the left-side navigation pane, click **Database Connection**.
7. In the **Intranet Connection - Classic Network** section, click **Switch to VPC**.
8. In the **VPC** dialog box that appears, specify **VPC** and **VSwitch**.

   > ? **Note**
   >
   > - You can turn on **Retain the connection address of the classic network** to generate new internal endpoints on the VPC and keep the existing internal endpoints on the classic network within a specified period. When an internal endpoint on a classic network expires, it is automatically released.
   > - If you do not turn on **Retain the connection address of the classic network**, there may be a brief disconnection while you switch the network type. In this case, Alibaba Cloud services (such as ECS) on the classic network cannot connect to this instance.

9. Click **OK**.

Switch from VPC to Classic Network

After you switch the network type of the instance to Classic Network, the internal endpoints on the VPC are released and ECS instances in the VPC can no longer connect to this instance with these endpoints. ApsaraDB for MongoDB generates new internal endpoints on the classic network and retains the same public endpoints. You must modify the connection information for your application.

> ⓘ **Note** After you switch the network type of the instance to Classic Network, ECS instances in the VPC can no longer connect to this instance. While you switch the network type of the instance, there may be a brief disconnection. We recommend that you perform this operation during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected. This protects your business against the brief disconnection.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Intranet Connection - VPC** section, click **Switch to Classic Network**.
7. In the message that appears, click **OK**.

8.6. Configure a VPC for a new instance

ApsaraDB for MongoDB supports two network types: classic network and VPC. This topic describes how to configure a VPC for a new ApsaraDB for MongoDB instance.

Context

On the Alibaba Cloud platform, a classic network and a VPC have the following differences:

- On the classic network, cloud services are not isolated. You can configure a security group or whitelist policy for them to block unauthorized access.
- A VPC helps you build an isolated network environment in Alibaba Cloud, where you can customize its routing table, IP address range, and gateway. In addition, you can use a physical connection or VPN to combine your user-created IDC with cloud resources in Alibaba Cloud VPC to create a virtual IDC, so that you can smoothly migrate your applications to the cloud.

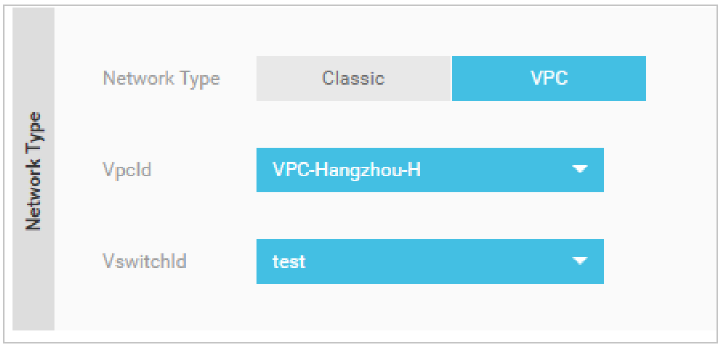
ApsaraDB for MongoDB uses VPC by default. To this end, you need to create an ApsaraDB for MongoDB instance and a VPC in the same region as follows:

- If you have not created an ApsaraDB for MongoDB instance, you can create a VPC first and create an ApsaraDB for MongoDB instance in the VPC following the procedure described in this topic.
- If you have created an ApsaraDB for MongoDB instance, you can create a VPC in the same region and add the ApsaraDB for MongoDB instance to the VPC. For more information, see Switch the network type of an instance.

Procedure

1. Create a VPC. For more information, see Create a VPC.
2. Create an ApsaraDB for MongoDB instance in the same region as the VPC.

3. When creating the ApsaraDB for MongoDB instance, select **VPC** as the network type on the instance creation page.
4. Under **VPC**, select the configured VPC and VSwitch for **VPC** and **VSwitch**, respectively, as shown in the following figure.



Network Type

Network Type: Classic | VPC

VpcId: VPC-Hangzhou-H

VswitchId: test

5. On the instance creation page, specify other configuration items as required. For more information, see the following links.
   - Create a standalone instance
   - Create a replica set instance
   - Create a sharded cluster instance

8.7. Configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC

This topic describes how to configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC without a brief disconnection or network disconnection.

Prerequisites

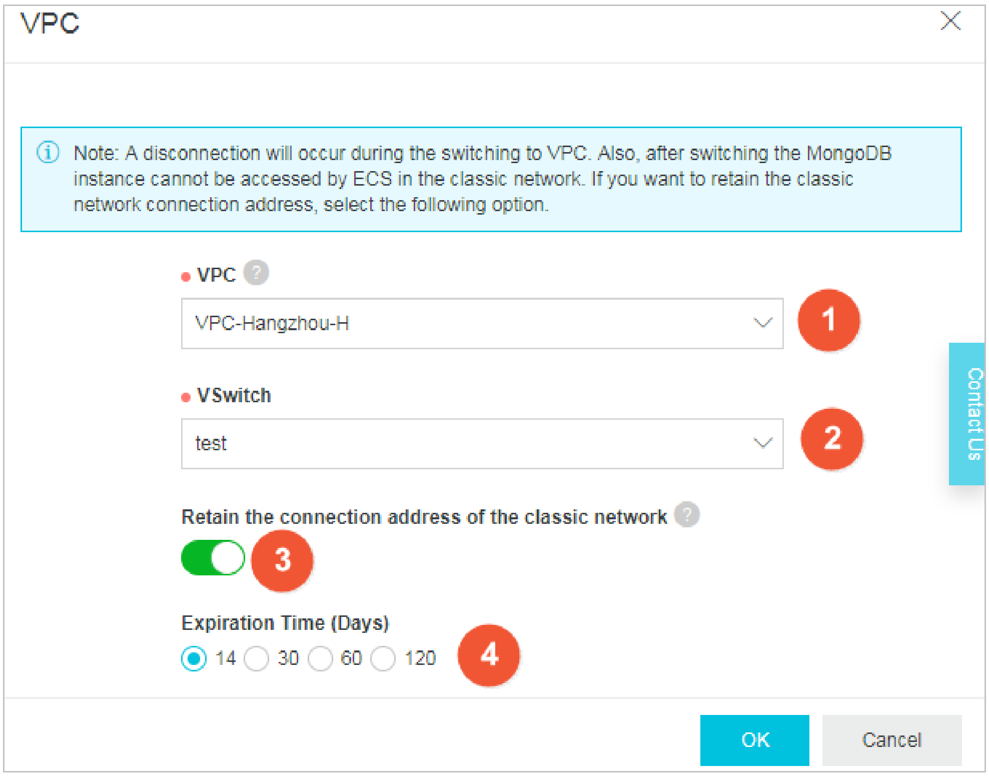
- The instance is a replica set or sharded cluster instance.
- The instance is in a classic network.
- A VPC is created in the same region as the instance. For more information, see Create a VPC and Create a VSwitch.

Background information

- In hybrid network access mode, you cannot switch the network type to Classic Network.
- While you switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC, you can choose to retain the internal endpoints on the classic network for up to 120 days. In hybrid network access mode, the instance supports access from ECS instances in both the classic network and VPC.
- In hybrid network access mode, you can switch the network types of ECS instances and other Alibaba Cloud services from Classic Network to VPC until all services are deployed in a VPC.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Intranet Connection - Classic Network** section, click **Switch to VPC**.
7. In the **VPC** dialog box that appears, configure related parameters.



VPC

Note: A disconnection will occur during the switching to VPC. Also, after switching the MongoDB instance cannot be accessed by ECS in the classic network. If you want to retain the classic network connection address, select the following option.

VPC: VPC-Hangzhou-H

VSwitch: test

Retain the connection address of the classic network: on

Expiration Time (Days): ● 14 ○ 30 ○ 60 ○ 120

   i. Specify **VPC** and **VSwitch**.
   ii. Turn on **Retain the connection address of the classic network**.
   iii. Set **Expiration Time (Days)**.

8. Click **OK**.

8.8. Change the retention period of internal endpoints on the classic network of an ApsaraDB for MongoDB instance

This topic describes how to change the retention period of internal endpoints on the classic network of an ApsaraDB for MongoDB instance.

Prerequisites

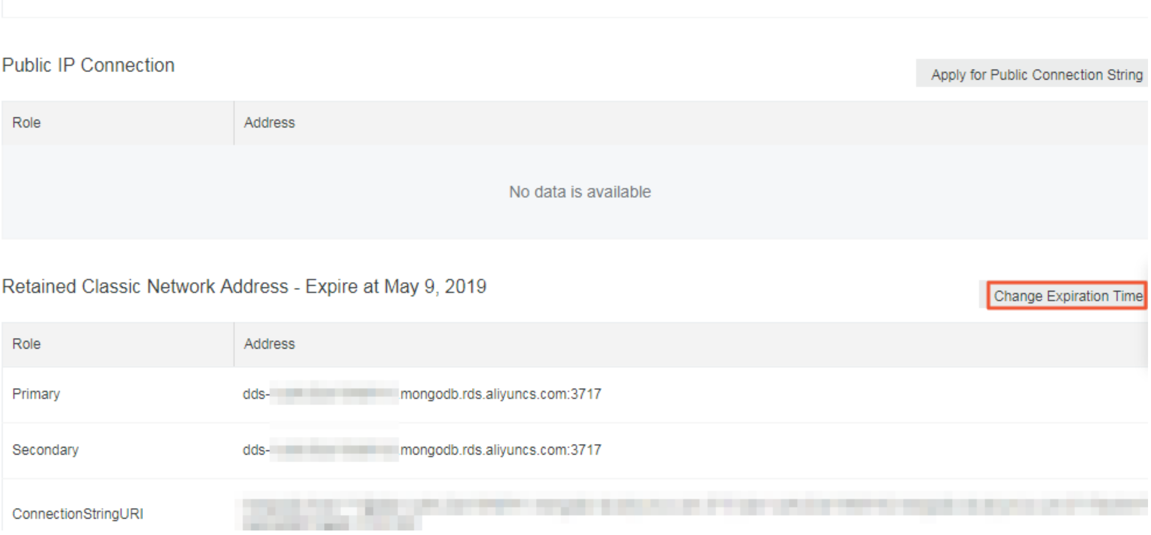
The instance is in hybrid network access mode. For more information, see Configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC.

Context

After an internal endpoint on the classic network expires, it is released. You can change the retention period of such an internal endpoint before it expires.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Database Connection**.
6. In the **Retained Classic Network Address** section, click **Change Expiration Time** .



Public IP Connection

Role	Address
No data is available	

Retained Classic Network Address - Expire at May 9, 2019

Role	Address
Primary	dds-[redacted]mongodb.rds.aliyuncs.com:3717
Secondary	dds-[redacted]mongodb.rds.aliyuncs.com:3717
ConnectionStringURI	[redacted]

7. In the Change Expiration Time panel, select a retention period.

> **Note** You can set the retention period to 14, 30, 60, or 120 days.

8. Click **OK**.

9.Data security

9.1. Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance

This topic describes how to configure an IP address whitelist or an ECS security group for an ApsaraDB for MongoDB instance. After you create an ApsaraDB for MongoDB instance, you must configure an IP address whitelist or add an ECS security group to allow access only from authorized devices. The default whitelist contains only the IP address 127.0.0.1, which indicates that no devices can access the ApsaraDB for MongoDB instance.

Prerequisites

When you add an ECS security group, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instances in the ECS security group. If both the ApsaraDB for MongoDB instance and ECS instances are of the VPC type, make sure that they reside in the same VPC.

Context

- Before you use an ApsaraDB for MongoDB instance for the first time, you must configure a whitelist for the instance. After you configure the whitelist, the endpoints of the instance appear on the **Basic Information** and **Database Connection** pages.
- Whitelists make your ApsaraDB for MongoDB instances more secure. We recommend that you maintain the whitelists on a regular basis.

Configure an IP address whitelist for a standalone instance, replica set instance, or sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
6. Find the IP address whitelist that you want to configure, and choose ⋮ > **Manually Modify** or **Import ECS Intranet IP** in the **Actions** column.

   Manually Modify

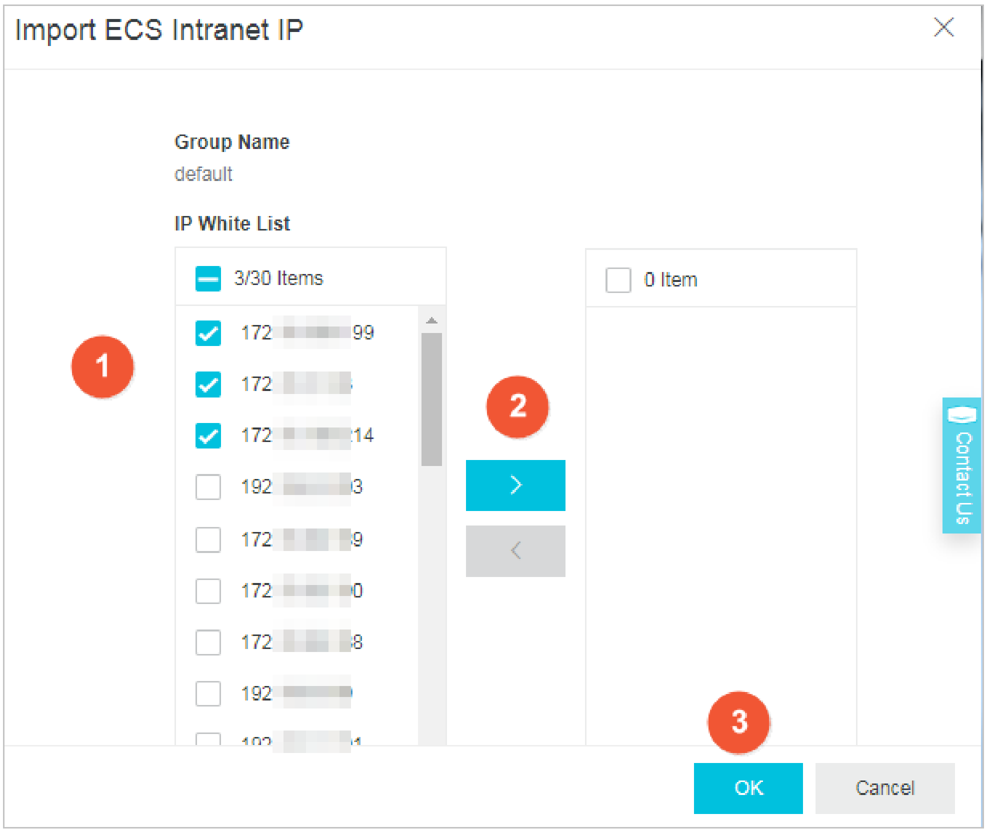
i. In the **Manually Modify** panel, click the **IPv4** or **IPv6** tab based on your network connection.

> Note
>
> - Limits for **IPv4** addresses:
>   - 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 IP addresses can access the ApsaraDB for MongoDB instance. This may introduce security risks to the instance. Proceed with caution.
> - Limits for **IPv6** addresses:
>   - Separate multiple IP addresses with commas (,). A maximum of 1,000 different IP addresses can be added.
>
>   Supported formats include `::` and `0:0:0:0:0:0:0:1`. Only IP addresses are supported. CIDR blocks will be supported later.
>   - If the whitelist is empty or contains only `::`, all IP addresses can access the ApsaraDB for MongoDB instance. This may introduce security risks to the instance. Proceed with caution.
>   - You can specify **IPv6** address whitelists only if the instance resides in Zone G of the China (Hangzhou) region.
>   - You can specify **IPv6** address whitelists only if the version of the database engine is 4.2.
> - You cannot specify both **IPv4** and **IPv6** addresses in a single whitelist. If you want to specify both IPv4 and IPv6 addresses, specify them in separate whitelists.

ii. Click **OK**.

Import ECS Intranet IP

i. Click **Import ECS Intranet IP**. In the Import ECS Intranet IP panel, the internal IP addresses of ECS instances created in the current account are displayed. Select one or more IP addresses and add them to the IP address whitelist.



Import ECS Intranet IP

Group Name
default

IP White List

3/30 Items

- ☑ 172[redacted]99
- ☑ 172[redacted]
- ☑ 172[redacted]14
- □ 192[redacted]3
- □ 172[redacted]9
- □ 172[redacted]0
- □ 172[redacted]8
- □ 192[redacted]
- □ 192[redacted]1

□ 0 Item

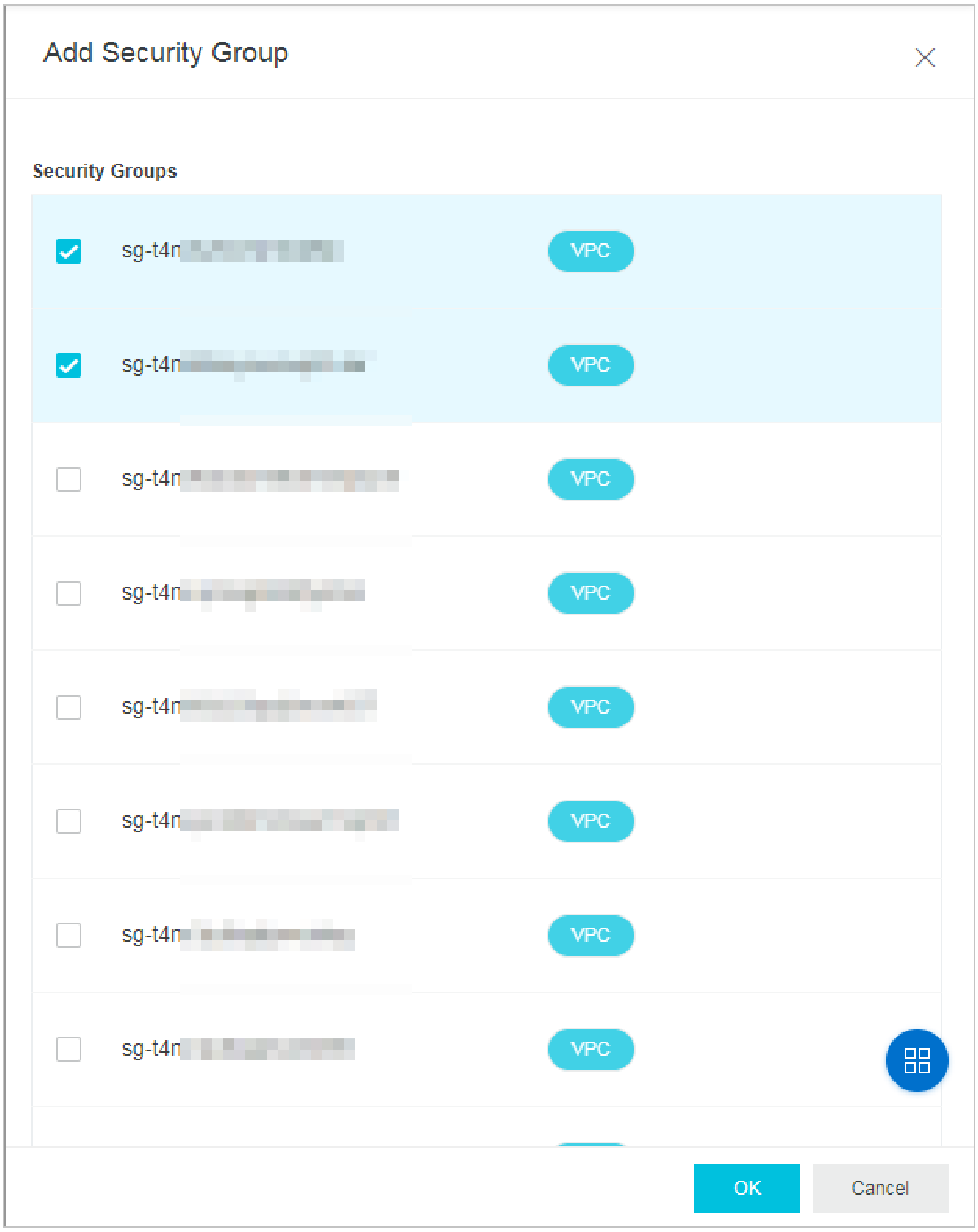
ii. Click **OK**.

> ⓘ **Note** For easy O&M and access control, we recommend that you add an ECS security group. For more information, see Configure an ECS security group for a standalone instance, replica set instance, or sharded cluster instance.

Configure an ECS security group for a standalone instance, replica set instance, or sharded cluster instance

An ECS security group relieves you from the tedious work of adding IP addresses or CIDR blocks. It makes database O&M easier.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5.
6. Click **Add Security Group**.
7. In the Add Security Group panel, select one or more ECS security groups that you want to add.



Add Security Group

Security Groups

☑ sg-t4n[redacted] VPC

☑ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

□ sg-t4n[redacted] VPC

Note

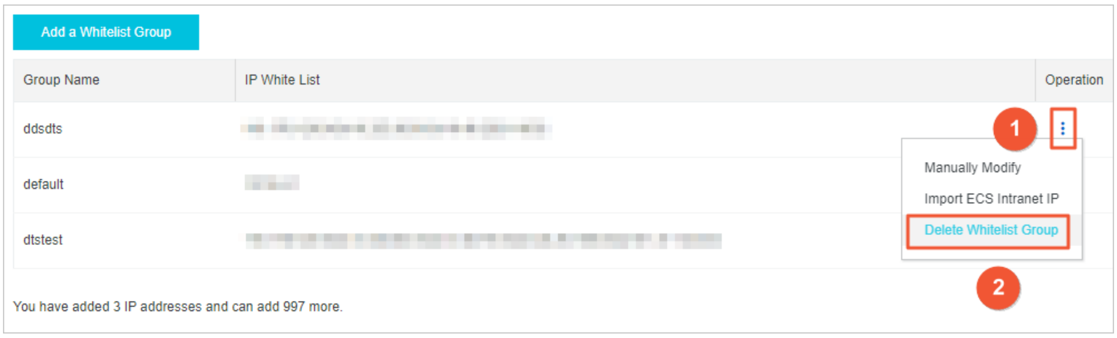
- Each ApsaraDB for MongoDB instance can be added to up to 10 security groups. After you add an ECS security group, all its ECS instances can access the ApsaraDB for MongoDB instance either over an internal network or over the Internet. For access over an internal network, the two types of instances must have the same network type. If the network type is VPC, the two types of instances must be in the same VPC. For access over the Internet, you must have applied for a public endpoint for the ApsaraDB for MongoDB instance.
- If you move your pointer over an ECS security group, you can view its name and description. If you move your pointer over VPC, you can view the VPC ID. This way, you can quickly find an ECS security group.

Delete a whitelist or an ECS security group of a standalone instance, replica set instance, or sharded cluster instance

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5.
6. Delete a whitelist or an ECS security group.To delete a whitelist, perform the following steps:
   i. Find the whitelist that you want to delete, and choose

      ⋮

      > **Delete Whitelist Group** in the **Actions** column.



Group Name	IP White List	Operation
ddsdts	[redacted]	⋮
default	[redacted]	
dtstest	[redacted]	

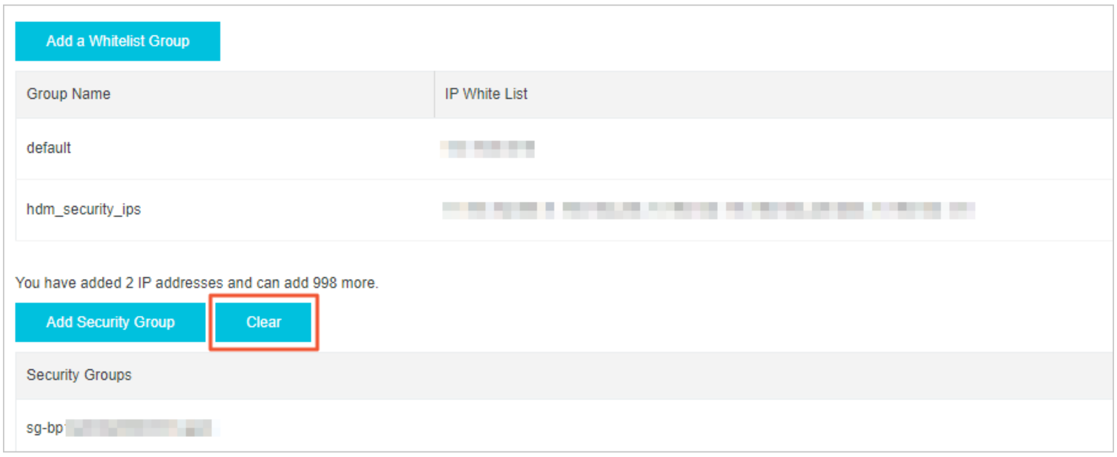
You have added 3 IP addresses and can add 997 more.

> **Note** You cannot delete the default whitelist.

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

To clear all ECS security groups, perform the following steps:

   i. Click **Clear**.



Group Name	IP White List
default	[redacted]
hdm_security_ips	[redacted]

You have added 2 IP addresses and can add 998 more.

Security Groups
sg-bp[redacted]

   ii. In the Clear Security Groups message, click **OK**.

Common connection scenarios

- Connect to an ApsaraDB for MongoDB instance over the Internet
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions
- Connect an ECS instance with an ApsaraDB for MongoDB instance in another Alibaba Cloud account

9.2. Configure SSL encryption for an ApsaraDB for MongoDB instance

This topic describes how to enhance link security by enabling Secure Sockets Layer (SSL) encryption and installing SSL CA certificates on your application services. The SSL encryption feature encrypts network connections at the transport layer to improve data security and ensure data integrity during communication. This topic describes operations related to SSL encryption.

Prerequisites

- The instance is a replica set instance.
- The database 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 houses more than 10,000 collections, the transient connections last longer.

Precautions

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

  > ? **Note** 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.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.

4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > SSL**.
6. Perform one of the following operations:

> ⓘ **Note** When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. 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 status is **Disabled**.	Turn on the switch next to **SSL Status**. In the message that appears, click **OK**.
Update an SSL CA certificate	The SSL encryption status is **Enabled**.	Click **Update Certificate**. In the message that appears, click **OK**.
Download an SSL CA certificate file	The SSL encryption status is **Enabled**.	Click **Download Certificate** to download an SSL CA certificate file to your computer.
Disable SSL encryption	The SSL encryption status is **Enabled**.	Turn off the switch next to **SSL Status**. In the message that appears, click **OK**.

References

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

9.3. 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. You can enable TDE for an instance in the ApsaraDB for MongoDB console to improve data security.

Prerequisites

- The instance is a replica set instance or a sharded cluster instance.
- The storage engine of the instance is WiredTiger.
- The database version of the instance is MongoDB 4.0 or 4.2. If the database version of the instance is earlier than MongoDB 4.0, upgrade the database version. For more information, see Upgrade MongoDB versions.

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

If the architecture or storage engine of your instance does not meet the requirements, you can create an instance that meets the requirements and migrate data of your instance to the new instance. For more information, see Configuration change overview.

Impacts

- When you enable TDE, your instance is restarted and a transient connection occurs between your application and the instance that connects to the application. We recommend that you enable TDE during off-peak hours and ensure that your application can reconnect to the instance in case of a transient connection.
- TDE increases the CPU usage of your instance.
- You cannot restore TDE-encrypted collections to a user-created ApsaraDB for MongoDB database from physical backup files. To restore TDE-encrypted collections to a user-created ApsaraDB for MongoDB database, use logical backup files.

Notes

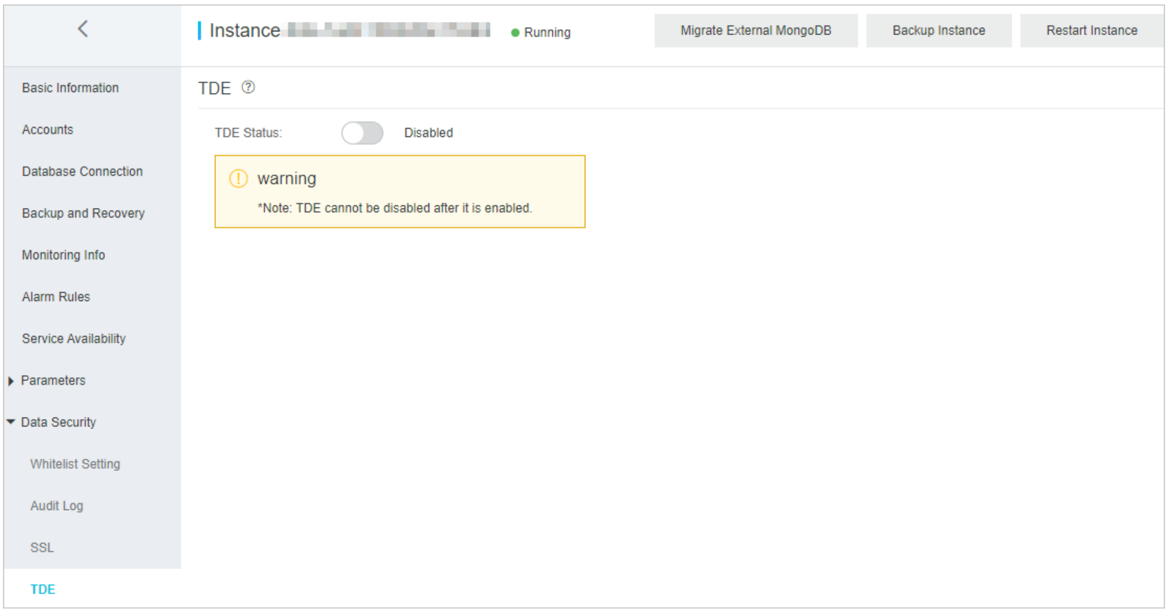
- After you enable TDE, you cannot disable TDE.
- You can enable TDE for an instance. You can also enable or disable encryption for a collection as required. If you need a filed-level encryption, see Explicit (Manual) Client-Side Field Level Encryption(only MongoDB 4.2 version instances are supported).

> ⓘ **Note** When you create a collection, you can disable encryption for the collection. For more information, see Disable encryption for a collection.

- After you enable TDE, only newly created collections are encrypted. Collections that are created before you enable TDE are not encrypted.
- Key Management Service (KMS) generates and manages the keys of TDE. ApsaraDB for MongoDB does not provide keys and certificates that are required for encryption.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > TDE**.
6. Turn on the switch next to **TDE Status:** to enable TDE.



TDE Status: Disabled

warning

*Note: TDE cannot be disabled after it is enabled.

7. In the **Enable TDE** dialog box, you can select **Use Automatically Generated Key** or **Use Custom Key**. Then, click **OK**. The instance status changes to **Modifying TDE**. After the status changes to **Running**, TDE is enabled.

   > Note You can use KMS to manage custom keys. For more information, see KMS.

Disable encryption for a collection

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

1. Connect to your instance through the mongo shell. For more information, see Connect to a replica set instance or Connect to a sharded cluster instance.
2. Run the following command to create a collection and disable the encryption feature:

```
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)" } } })
```

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

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

Prerequisites

- The instance is a replica set instance, and the database version of the instance is 3.4 or 4.0..

  > **Note** If the database version of the instance is earlier than required versions, you must upgrade the database version. For more information, see Upgrade MongoDB versions.

- SSL encryption is enabled for the instance. For more information, see Configure SSL encryption for an ApsaraDB for MongoDB instance.
- Mongo shell 3.0 or later is installed on the local server or ECS instance from which you want to connect to the database. For more information about the installation procedure, visit Install MongoDB.
- The IP address of the local server or the ECS instance is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

Precautions

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

Procedure

A local server with a Linux operating system is used in the following example:

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

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

3. On the local server or in the ECS instance, run the following command to connect to a database of the ApsaraDB for MongoDB instance:

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

Note

- <host>: the connection string (including the port number) of the primary or secondary node in the ApsaraDB for MongoDB instance. For more information, see Overview of replica set instance connections.
    - If you want to connect to a database of the ApsaraDB for MongoDB instance over the Internet, apply for a public endpoint for this instance. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.
    - If you want to connect to a database of the ApsaraDB for MongoDB instance over an internal network, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instance. If the network type is VPC, make sure that the two instances are in the same VPC.
- <username>: the username you use to log on to a database of the ApsaraDB for MongoDB instance. The initial username is root. We recommend that you do not log on to a database as the root user in a production environment. You can create users and grant permissions to the users as needed. For more information, see Manage user permissions on MongoDB databases.
- <database>: the name of database corresponding to the username if authentication is enabled. If the username is root, enter admin.
- <sslCAFile_path>: the path of the SSL CA certificate files.

Example:

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

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

Note

- The password characters are not explicitly displayed when you enter the password.
- If you forget the password of the root user, you can reset the password. For more information, see Set a password.

Common connection scenarios

- Connect to an ApsaraDB for MongoDB instance over the Internet
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions
- Connect an ECS instance with an ApsaraDB for MongoDB instance in another Alibaba Cloud account

9.5. Audit logs (new version)

9.5.1. Enable the new audit log feature

The new audit log feature integrates the features of Log Service with ApsaraDB for MongoDB. The audit log feature allows you to filter log data, analyze logs online, and export logs. This helps you discover security and performance issues of your ApsaraDB for MongoDB instances at your earliest opportunity.

Prerequisites

If you want to enable the new audit log feature as a RAM user, the RAM user must be granted the AliyunLogFullAccess permission. For more information, see Grant permissions.

Context

Alibaba Cloud Log Service is an all-in-one service and has been used in big data analytics scenarios. Log Service allows you to collect, consume, ship, search, and analyze log data without the need for extra code resources. This service makes O&M more efficient ApsaraDB for MongoDB integrates the features of Log Service to provide the new audit log feature, which is characterized by stability, ease of use, flexibility, and high efficiency.

This topic describes how to enable the new audit log feature.

Notes

After you enable the new audit log feature, Log Service records operations on ApsaraDB for MongoDB instances in logs to facilitate troubleshooting.

Billing

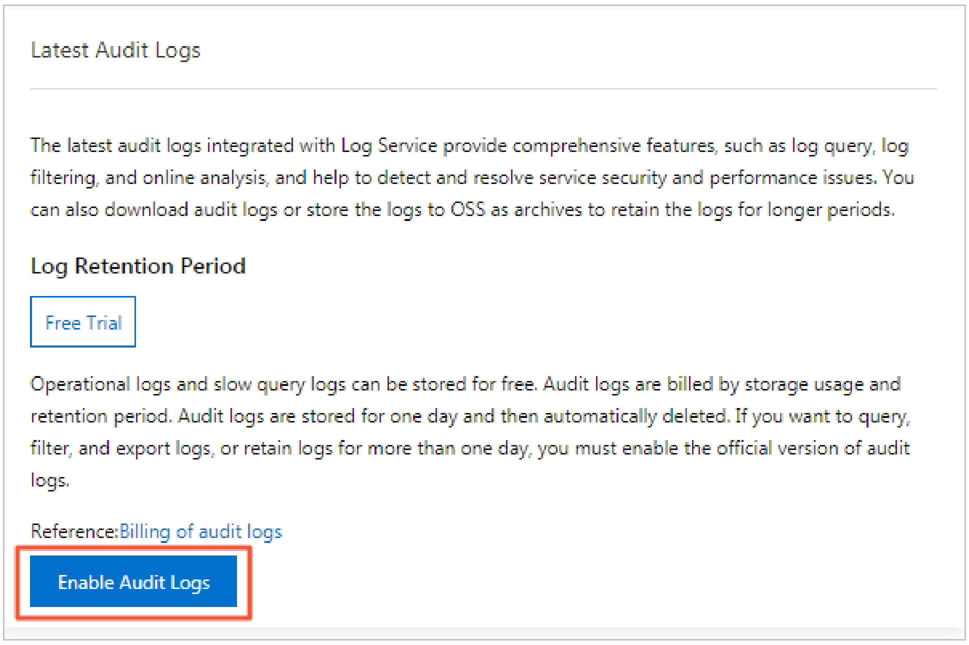
The new audit log feature has two editions: trial and official.

> ? **Note** The trial edition has been available. The official edition is planning to be available at a later date.

- In the trial edition, audit logs can be stored for one day with a maximum storage capacity of 100 GB.
- The official version is charged based on the size of audit logs and retention period. The official version supports more features than the free trial version.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > Audit Logs**.
6. Click **Enable Audit Logs**.



Latest Audit Logs

The latest audit logs integrated with Log Service provide comprehensive features, such as log query, log filtering, and online analysis, and help to detect and resolve service security and performance issues. You can also download audit logs or store the logs to OSS as archives to retain the logs for longer periods.

Log Retention Period

Operational logs and slow query logs can be stored for free. Audit logs are billed by storage usage and retention period. Audit logs are stored for one day and then automatically deleted. If you want to query, filter, and export logs, or retain logs for more than one day, you must enable the official version of audit logs.

Reference:Billing of audit logs

7. Click **OK**.

What's next

- Query audit logs
- Download audit logs
- Modify audit settings
- Subscribe to audit log reports
- Disable audit logging

Related information

- Overview
- Slow query logs

9.5.2. Query audit logs

In the ApsaraDB for MongoDB console, you can view audit logs in a specified time range and filter audit logs that match various conditions.

Prerequisites

You have enabled the new audit log feature. For more information, see Enable the new audit log feature.

Context

You can query audit logs for detailed insight when you want to view database request records, discover the cause for sudden increases in resource consumption, or find records of modify and delete operations on data.

View audit logs

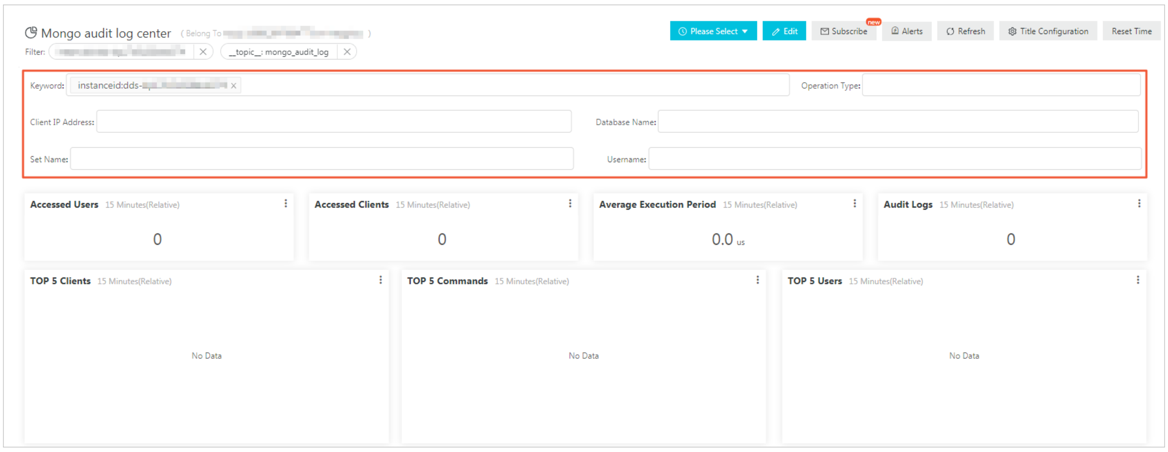
1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. On the **Mongo audit log center** page that appears, view audit log details of the ApsaraDB for MongoDB instance.

Filter audit logs

You can define different conditions to filter audit logs.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. On the **Mongo audit log center** page that appears, define conditions to filter audit logs.



Mongo audit log center (Belong To [redacted])

Filter: [redacted] ✕ __topic__: mongo_audit_log ✕

Keyword: instanceid:dds-[redacted] ×

Operation Type:

Client IP Address:

Database Name:

Set Name:

Username:

Accessed Users 15 Minutes(Relative) 0

Accessed Clients 15 Minutes(Relative) 0

Average Execution Period 15 Minutes(Relative) 0.0 us

Audit Logs 15 Minutes(Relative) 0

TOP 5 Clients 15 Minutes(Relative) No Data

TOP 5 Commands 15 Minutes(Relative) No Data

TOP 5 Users 15 Minutes(Relative) No Data

Filter conditions

Filter condition	Description

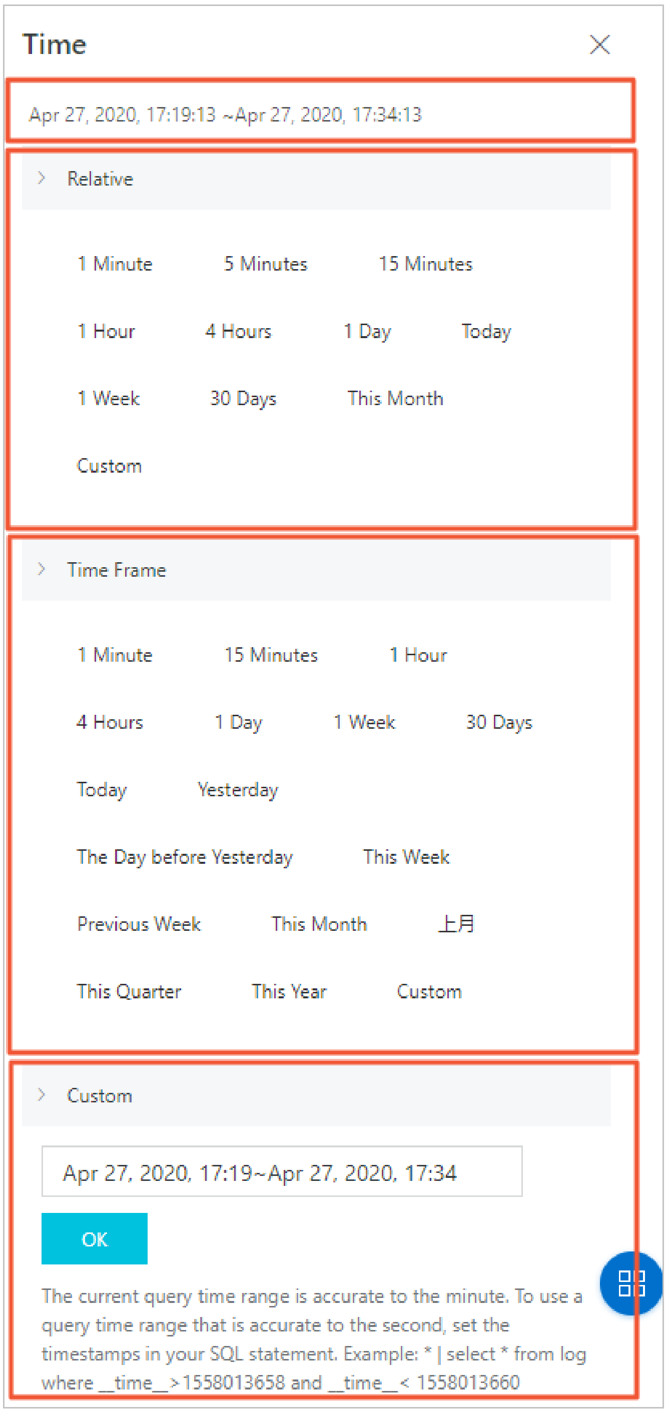
Filter condition	Description
Keyword	Filters audit logs by keywords such as the client IP address, executed commands, accounts, and extended information. Note ◦ The Keyword filed supports exact match, so you must enter complete information. ▪ For example, you must enter a complete IP address such as 192.168.1.1, instead of 192.168 or 1.1. ▪ You must enter a complete command such as AUTH or auth, instead of au. ◦ You must enclose keywords that contain colons within double quotation marks (""), such as *"userId:1"*.
Operation Type	Filters audit logs by operation type. Operation types include: ◦ query ◦ find ◦ insert ◦ update ◦ delete ◦ remove ◦ getMore: the read operation ◦ command: the protocol command such as the aggregate method
Client IP Address	The client IP address used to connect the ApsaraDB for MongoDB instance.
Database Name	The name of the ApsaraDB for MongoDB database.
Set Name	The name of the ApsaraDB for MongoDB instance set.
Username	The account used to log on to the ApsaraDB for MongoDB instance.

View audit logs within a specified time range

You can view slow query logs within a specified time range by using the time picker.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.

5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. On the **Mongo audit log center** page that appears, click **Please Select**.
7. Specify the time range in the time picker.



Apr 27, 2020, 17:19:13 ~Apr 27, 2020, 17:34:13

Apr 27, 2020, 17:19~Apr 27, 2020, 17:34

The current query time range is accurate to the minute. To use a query time range that is accurate to the second, set the timestamps in your SQL statement. Example: * | select * from log where __time__>1558013658 and __time__< 1558013660

Time picker sections

No.	Section	Description
1	Time	Information about the time range is displayed in this section when you move the pointer over a relative time or a time frame.

No.	Section	Description
2	Relative	A time period relative to the current point in time. Information about the time range is displayed in the Time section when you move the pointer over any element in this section.
2	Time Frame	A time frame period that is more than one minute. Information about the time range is displayed in the Time section when you move the pointer over any element in this section.
4	Custom	A custom time period. Specify a time period and click **OK** to confirm the time period.

FAQ

- I can only view 2,000 audit log entries in total. Where can I view the others?

  The Audit Logs page on the ApsaraDB for MongoDB console displays up to 2,000 audit log entries. To view more audit log entries, you must log on to the Log Service console. For more information, see Query logs.

- Where can I view old audit log documentation?

  See Configure audit logging for an ApsaraDB for MongoDB instance.

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

9.5.3. Modify audit settings

This topic describes how to modify the operation types to be collected in the audit logs by modifying **Audit Log Filter Setting**.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. In the upper-left corner of the page, click **Audit Log Filter Setting**.
7. In the **Audit Log Filter Setting** dialog box, select the operation types that you want to audit, and click **Submit**.

Note The following list describes the operation types:

- admin: O&M operations
- slow: slow queries
- query: queries
- insert: data insertion
- update: data updates
- delete: data deletion
- command: protocol commands, such as the aggregate method

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

9.5.4. Download audit logs

You can download the queried audit logs to your local server, and then archive, filter, or analyze the logs.

Prerequisites

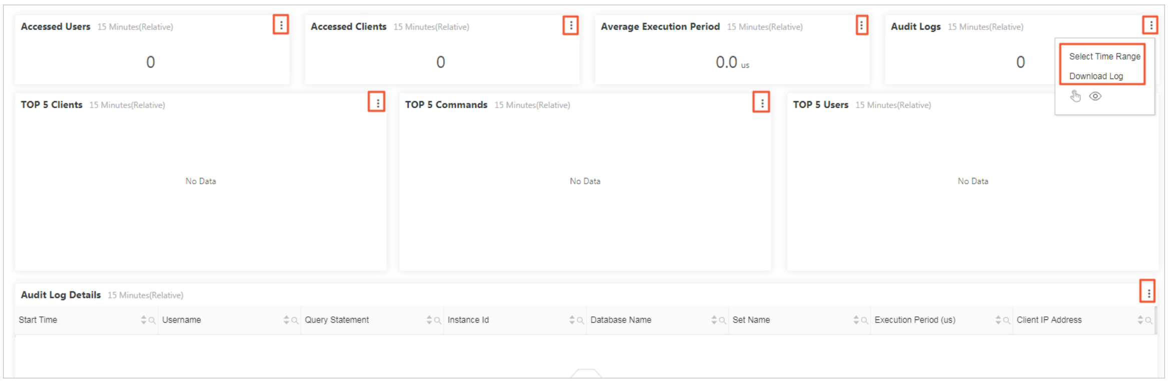
You have enabled the new audit log feature. For more information, see Enable the new audit log feature.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. In the log chart section, choose ⋮ **> Download Log** in the upper-right corner of the target chart.

Note You can filter logs by using the following methods. Then, you can download the content that meets your requirements.

- Filter log data by keyword, type, account, or client IP address. For more information, see Filter audit logs.
- Filter logs by the time when logs are generated. Click **Select Time Range** above the **Download Log** button to select a time range.


Accessed Users 15 Minutes(Relative)
0
Accessed Clients 15 Minutes(Relative)
0
Average Execution Period 15 Minutes(Relative)
0.0 us
Audit Logs 15 Minutes(Relative)
0
Select Time Range
Download Log
TOP 5 Clients 15 Minutes(Relative)
No Data
TOP 5 Commands 15 Minutes(Relative)
No Data
TOP 5 Users 15 Minutes(Relative)
No Data
Audit Log Details 15 Minutes(Relative)
Start Time
Username
Query Statement
Instance Id
Database Name
Set Name
Execution Period (us)
Client IP Address

After you click **Download Log**, the selected log entries are saved to a *.csv* file on your local server through the web browser. Then, you can view the log data by using tools such as Excel.

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

9.5.5. Subscribe to audit log reports

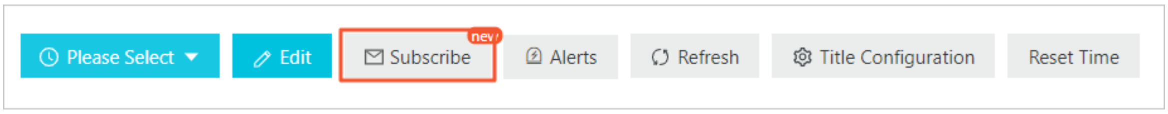
You can subscribe to audit log reports of ApsaraDB for MongoDB through emails or the DingTalk ChatBot. This allows you to retrieve information about ApsaraDB for MongoDB regularly.

Prerequisites

You have enabled the new audit log feature. For more information, see Enable the new audit log feature.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. On the **Mongo audit log center** page, click **Subscribe** in the upper-right corner.

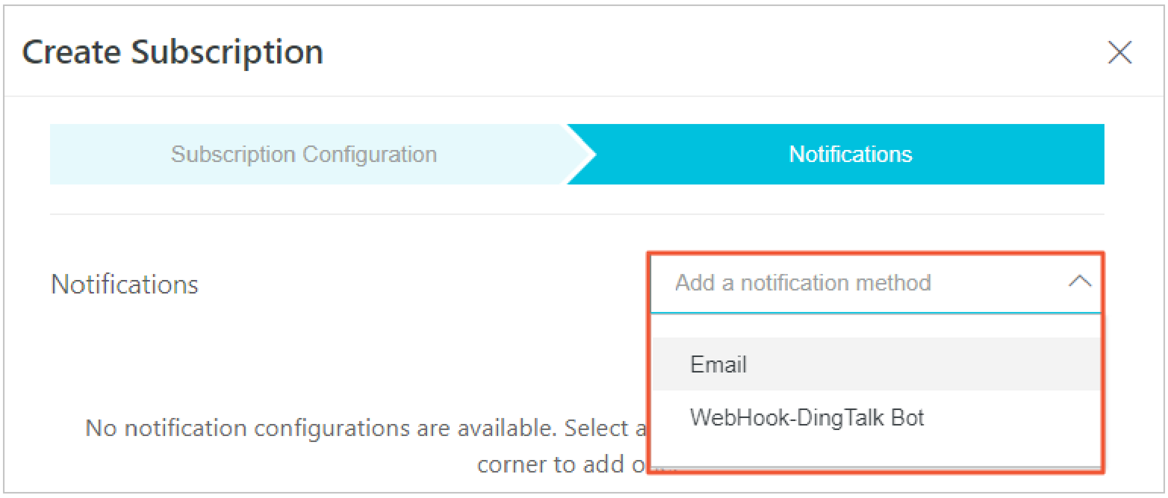


7. In the **Subscription Configuration** step, complete the settings and click **Next** at the lower part of the page.The following table describes the parameters on the Subscription Configuration page.

Parameter	Description
Subscription Name	The description of the subscription. You can customize the description.

Parameter	Description
Frequency	Specifies the frequency at which ApsaraDB for MongoDB sends the reports.
Add Watermark	After you enable this feature, the images in the reports are watermarked with the email address or the WebHook URL.

8. In the **Notifications** step, click the drop-down list on the right and select a notification method.


Create Subscription

No notification configurations are available. Select a ... corner to add o...

Note The available notification methods are **Email** and **WebHook-DingTalk Bot**. You can choose one or both of them.

9. Specify the **Recipients** of the **Email** or enter the **Request URL** of the **WebHook-DingTalk Bot**, and then click **Submit** at the lower part of the page.

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

9.5.6. Disable audit logging

This topic describes how to disable audit logging for an ApsaraDB for MongoDB instance.

Context

ApsaraDB for MongoDB does not support disabling the audit log feature. To reduce the number of audit logs and save storage space, you can disable audit logging for specific operation types.

Impacts

If you disable audit logging for a specific operation type, the system ignores the operation type in future auditing. However, the audit logs generated before you modify the settings are retained. You can use the audit logs for backtracking. For more information, see Query audit logs.

Disable audit logging

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane of the **Instance** page, choose **Data Security > Audit Logs**.
6. In the upper-left corner of the page, click **Audit Log Filter Setting**.
7. In the **Audit Log Filter Setting** dialog box, clear the target operation types, and click **Submit**.

> **Note** We recommend that you select `admin` and `slow`. The log size of the two operation types is small. You can use these audit logs for troubleshooting. For more information, see Operation types.

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

10.Monitoring and alerting

10.1. View monitoring information

The ApsaraDB for MongoDB console provides a wide range of performance monitoring information for you to check the running status of instances.

Precautions

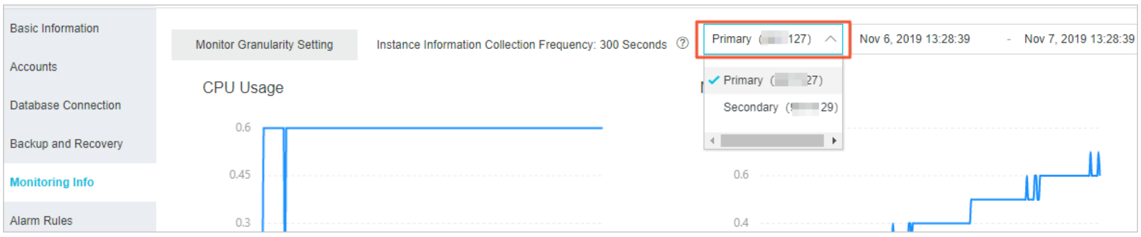
- When you receive an alert message from Alibaba Cloud, such as a message indicating that the CPU utilization of your instance exceeds 80%, you can view monitoring information about the instance on ApsaraDB for MongoDB console to troubleshoot the issue. You can filter the nodes of the instance to check the status of each node and locate the node where the issue occurs.
- Monitoring information is retained for up to seven days. You cannot view the monitoring information that was generated seven days ago.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, **Sharded Cluster Instances**, or **Serverless Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Monitoring Info**.
6. View monitoring information based on instance types:

   > ⓘ **Note** By default, the Monitoring Info page displays the monitoring information of the last day. You can also select a time range to view historical monitoring information.

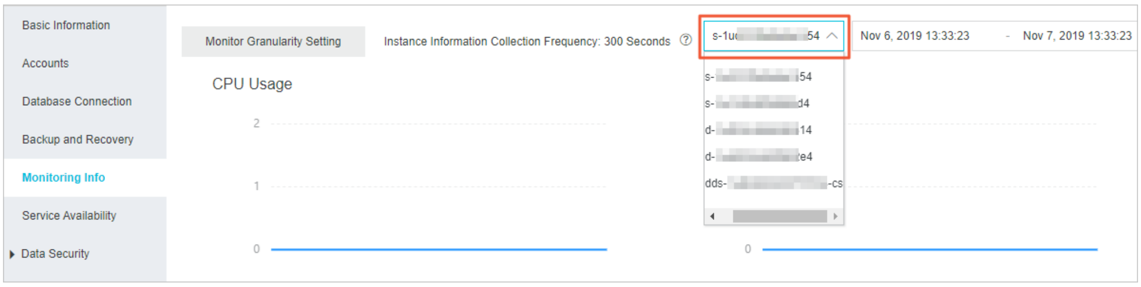
   - Standalone instances: You can only view the monitoring information about primary nodes.
   - Replica set instances: You can view the monitoring information about primary or secondary nodes by selecting a node from the drop-down list in the upper part of the Monitoring Info page.



Instance Information Collection Frequency: 300 Seconds

   - Sharded cluster instances: You can view the monitoring information about mongos, shard, or ConfigServer nodes by selecting a node from the drop-down list in the upper part of the Monitoring Info page.

   > ⓘ **Note** Mongos nodes have IDs prefixed with `s-`. Shard nodes have IDs prefixed with `d-`. ConfigServer nodes have IDs suffixed with `-cs`.



Instance Information Collection Frequency: 300 Seconds

Performance metrics

Performance metric	Description
CPU Utilization Percentage	cpu_usage: the CPU utilization of the instance.
Memory Usage Percentage	mem_usage: the memory usage of the instance.
IOPS Usage	The input/output operations per second (IOPS) of the instance. The following items are included: ● data_iops: the IOPS of the data disk. ● log_iops: the IOPS of the log disk.
IOPS Usage Percentage	The percentage 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 space used by the data disk. ● log_size: the space used on the log disk.
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.
Opcounters	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	The current number of 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.

Performance metric	Description
Network	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.
Global Lock	The length of the queues that are waiting for global locks for the instance. The following items are included: • 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. • gl_cq_total: the length of the queue that is waiting for both global read and 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 node and secondary nodes of the instance.
WT Request Queues	The number of concurrent requests that are being handled and the remaining number of concurrent requests that can be handled. The following items are included: • write_concurrent_trans_out: the number of concurrent write requests that are being handled. • read_concurrent_trans_out: the number of concurrent read requests that are being handled. • write_concurrent_trans_available: the remaining number of concurrent write requests that can be handled. • read_concurrent_trans_available: the remaining number of concurrent read requests that can be handled.
IO Latency	iocheck_cost: the latency of I/O operations, indicating the I/O response performance.

10.2. Set alert rules

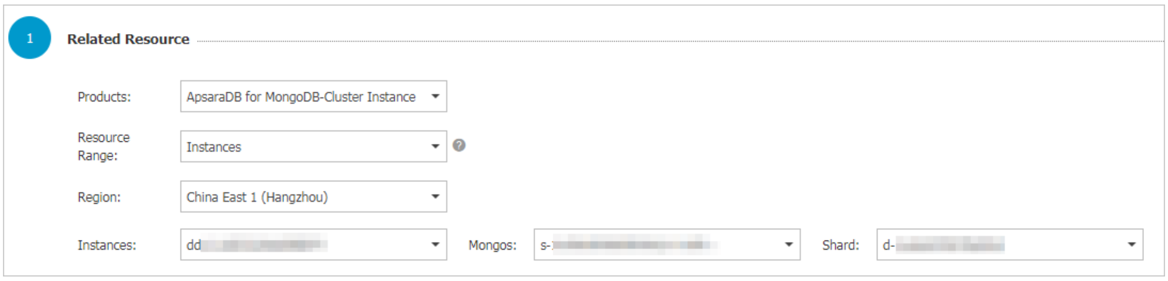
ApsaraDB for MongoDB provides the instance status monitoring and alerting feature. You can set alert rules for important metrics to help detect abnormal data in a timely manner and quickly locate and handle faults.

Procedure

1. Log on to the Cloud Monitor console.

> **Note** On this tab, you can view existing alert rules.

2. On the Alarm Rules tab of the **Cloud Monitor console**, click **Create Alarm Rule** in the upper-right corner.
3. On the **Create Alarm Rule** page, configure related resource parameters.



1 **Related Resource**

Products: ApsaraDB for MongoDB-Cluster Instance

Resource Range: Instances

Region: China East 1 (Hangzhou)

Instances: dd[redacted] Mongos: s-[redacted] Shard: d-[redacted]

Parameter	Description
Product	The architecture of the instance. Valid values: ○ ApsaraDB for MongoDB-Instance Copy ○ ApsaraDB for MongoDB-Cluster Instance ○ ApsaraDB for MongoDB-Single Node Instance **Note** If you select **ApsaraDB for MongoDB-Cluster Instance**, you must specify the **mongos** and **shard** nodes to be monitored.
Resource Range	○ **All Resources**: The alert rule is applicable to all ApsaraDB for MongoDB instances. ○ **Instances**: The alert rule is applicable to the specified ApsaraDB for MongoDB instances.
Region	The region where the instance is deployed.
Instances	The ID of the instance. You can select multiple instance IDs.

4. Set alert rules and the notification method. For more information about the parameters, see Manage alert rules.

> **Note** To create alert contacts in the Cloud Monitor console, see Create an alert contact or alert group.

5. Click **Confirm**. Alert rules automatically take effect.

For more information about the metrics, see links below:

- ApsaraDB for MongoDB-Single Node Instance
- ApsaraDB for MongoDB-Instance Copy
- ApsaraDB for MongoDB-Cluster Instance

11.Parameter settings

11.1. Configure database parameters for an ApsaraDB for MongoDB instance

This topic describes how to configure database parameters for an ApsaraDB for MongoDB instance to better fit your business needs.

Prerequisites

The instance is a standalone or replica set instance.

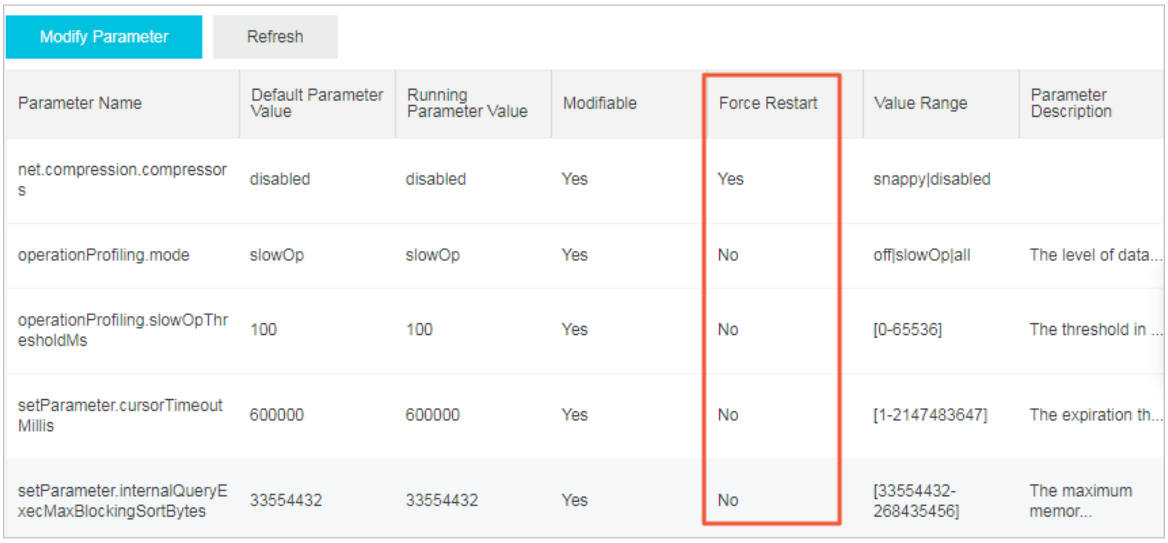
Precautions

After you save the changes to some parameters, the instance is restarted. For more information, see descriptions in the **Force Restart** column on the **Parameter List** page.

> **Note** While the instance is restarting, it is not connected. We recommend that you restart your instance during off-peak hours to minimize the impact on your business.

Procedure

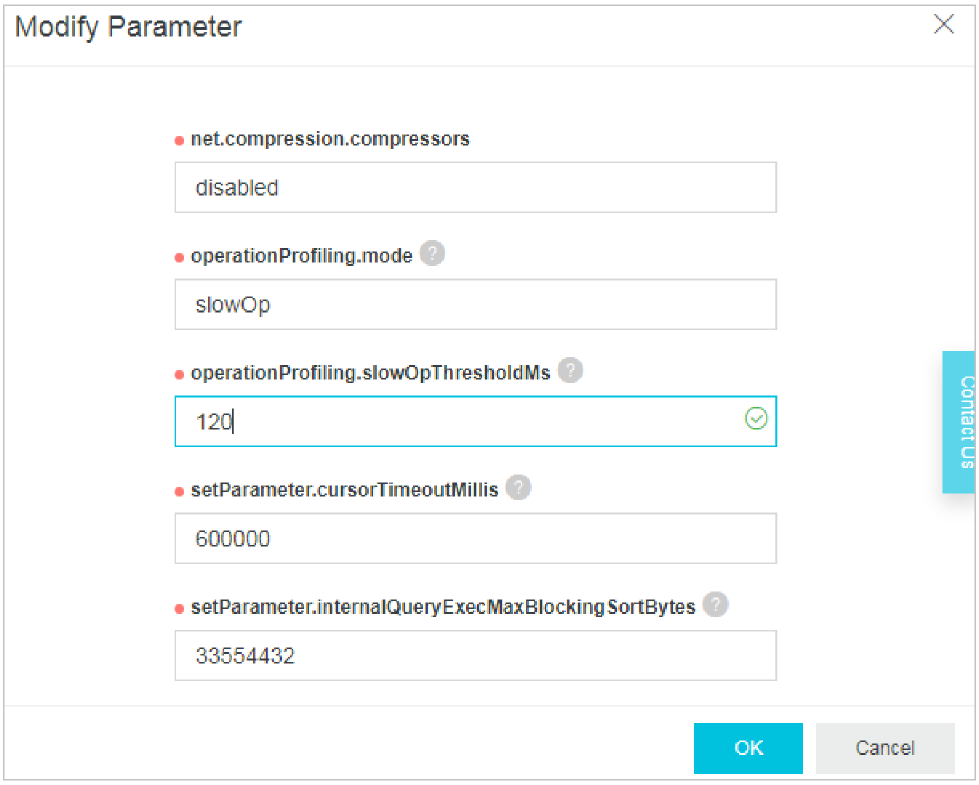
1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Parameters > Parameter List**.
6. Click **Modify Parameter**.



Parameter Name	Default Parameter Value	Running Parameter Value	Modifiable	Force Restart	Value Range	Parameter Description
net.compression.compressors	disabled	disabled	Yes	Yes	snappy|disabled	
operationProfiling.mode	slowOp	slowOp	Yes	No	off|slowOp|all	The level of data...
operationProfiling.slowOpThresholdMs	100	100	Yes	No	[0-65536]	The threshold in ...
setParameter.cursorTimeoutMillis	600000	600000	Yes	No	[1-2147483647]	The expiration th...
setParameter.internalQueryExecMaxBlockingSortBytes	33554432	33554432	Yes	No	[33554432-268435456]	The maximum memor...

> Note On the parameter list, you can check whether the instance needs to be restarted after you modify a parameter.

7. In the **Modify Parameter** dialog box that appears, modify parameters as needed.



Modify Parameter

- net.compression.compressors: disabled
- operationProfiling.mode: slowOp
- operationProfiling.slowOpThresholdMs: 120
- setParameter.cursorTimeoutMillis: 600000
- setParameter.internalQueryExecMaxBlockingSortBytes: 33554432

> Note
> - You can modify more than one parameter in this step.
> - You must configure parameters in compliance with the value ranges displayed in the console.

8. Click **OK**.

11.2. View the parameter modification history

In the ApsaraDB for MongoDB console, you can modify database parameters and then view the parameter modification history.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.

4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Parameters > Modification History**.
   On the **Modification History** page, modification records of the last 24 hours are displayed by default. You can also specify a time range to query parameter modification records.

12.Primary/Secondary failover

12.1. Trigger a primary/secondary failover for a replica set instance

An ApsaraDB for MongoDB replica set instance consists of three nodes by default. 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. If a node is faulty, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to ensue the availability of the instance. You also can manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

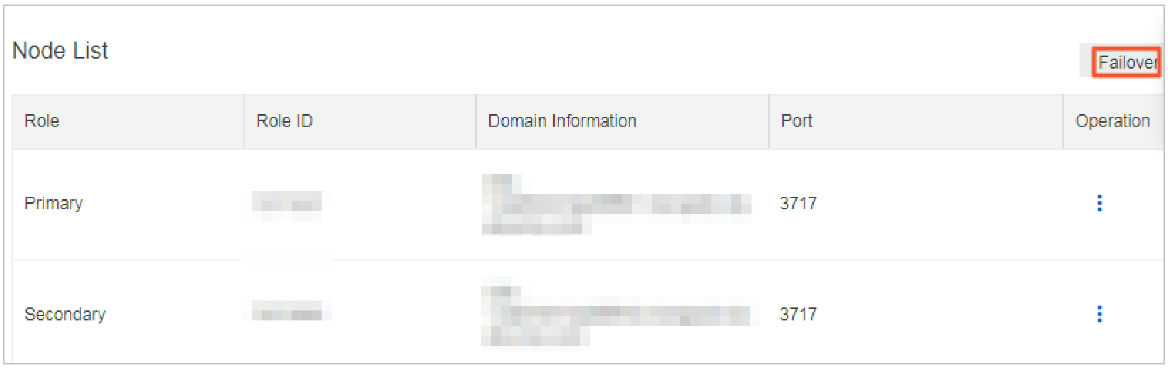
Context

After you log on to the ApsaraDB for MongoDB console or call the SwitchDBInstanceHA operation to trigger a primary/secondary failover for a replica set instance, ApsaraDB for MongoDB interchanges the roles of the primary and secondary nodes.

> Note
>
> - You can trigger a primary/secondary failover only for replica set and sharded cluster instances, but not for standalone instances due to their single-node architecture.
> - After you trigger a primary/secondary failover for an instance, a transient connection error of up to 30 seconds will occur to the instance. Ensure that your applications can automatically re-establish a connection.
> - You can trigger a primary/secondary failover only for instances in the running state.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the **Node List** section, click **Failover**, as shown in the following figure.



Node List

Failover

Role	Role ID	Domain Information	Port	Operation
Primary	[redacted]	[redacted]	3717	⋮
Secondary	[redacted]	[redacted]	3717	⋮

6. In the **Failover** dialog box, select **Effective At** and click **Submit**.

> ⓘ Note Valid values for the **Effective At** parameter in the **Failover** dialog box:
>
> - **Effective Immediately**: specifies that the system runs the failover task immediately.
> - **Effective Within Maintenance Window**: specifies that the system runs the failover task within the specified maintenance period. For more information, see Specify a maintenance period.

7. The instance status changes to **HA Switching**. The failover is successful when the instance status changes back to **Running**.

   The failover takes about one minute. Then the instance returns to normal.

> ⓘ Note If you have connected to the connection string of the primary node for an instance, you are connecting to a secondary node after a failover and you have no write permissions on the instance. In this case, you must connect to the connection string of the new primary node and obtain read and write permissions. For more information, see Overview of replica set instance connections.

12.2. Trigger a primary/secondary failover for a shard of a sharded cluster instance

Each shard of a sharded cluster instance consists of three nodes by default. If a node is faulty, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to guarantee the availability of the shard. In addition, you can manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

Notes

ApsaraDB for MongoDB provides addresses for you to connect to the primary node and a secondary node of a shard. The other secondary node is hidden as a backup to guarantee high availability. After you log on to the ApsaraDB for MongoDB console or call the SwitchDBInstanceHA operation to trigger a primary/secondary failover for a shard of a sharded cluster instance, ApsaraDB for MongoDB interchanges the roles of the primary and secondary nodes.

> ⓘ Note
>
> - You can trigger a primary/secondary failover only for replica set and sharded cluster instances, but not for standalone instances due to their single-node architecture.
> - You can trigger a primary/secondary failover only for shards in the normal running status.
> - After you trigger a primary/secondary failover for an instance, the instance may be disconnected for 30s once. We recommend that you perform this operation during off-peak hours and ensure that your applications can automatically re-establish a connection.

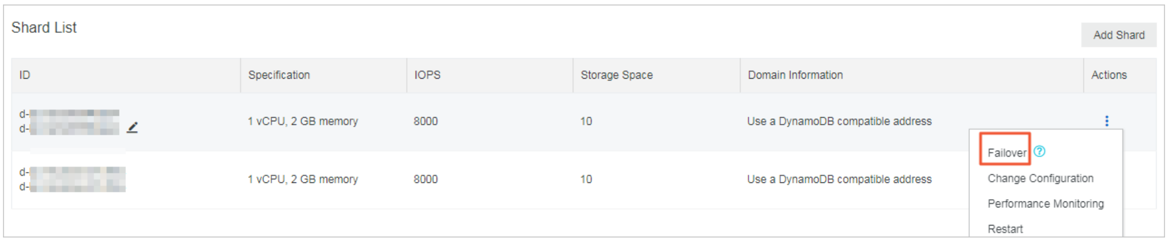
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.

3. In the left-side navigation pane, click **Sharded Cluster Instances**.
4. Find the target instance and click its ID.
5. In the **Shard List** area, locate the target shard and choose

   ⋮

   > **Failover** in the Operation column.

ID	Specification	IOPS	Storage Space	Domain Information	Actions
d-[redacted] d-[redacted]	1 vCPU, 2 GB memory	8000	10	Use a DynamoDB compatible address	⋮
d-[redacted] d-[redacted]	1 vCPU, 2 GB memory	8000	10	Use a DynamoDB compatible address	

   You can trigger a primary/secondary failover separately for each shard. The failover takes effect only for the current node and does not affect other shards of the same sharded cluster instance.
6. In the **Failover** dialog box that appears, choose **Effective At** and click **Submit**.

   > Note You can set **Effective At** to specify the time when the **Failover** task takes effect. The following options are supported:
   >
   > - **Effective Immediately**: specifies that the system runs the failover task immediately.
   > - **Effective Within Maintenance Window**: specifies that the system runs the failover task within the specified maintenance period. For more information, see Specify a maintenance period.

7. The instance status changes to **HA Switching**.The failover is successful when the instance status changes back to **Running**.

   > Note The failover takes about 1 minute. You can repeat the preceding procedure to trigger a primary/secondary failover for other shards of the same sharded cluster instance as required.

13.Log management

13.1. Overview

ApsaraDB for MongoDB enables the log management feature. This topic describes how to view the logs of ApsaraDB for MongoDB instances.

Use the log management feature to view logs

Log type	Related topics
Slow query logs	View slow query logs
Error logs	View error logs
Running logs	View operation logs

Use other methods to view logs

You can also view audit logs and slow query logs in CloudDBA. For more information, see Enable the new audit log feature and Slow query logs.

13.2. View slow query logs

You can view the slow query logs of a database in the console and optimize the database accordingly by analyzing slow query logs.

Prerequisites

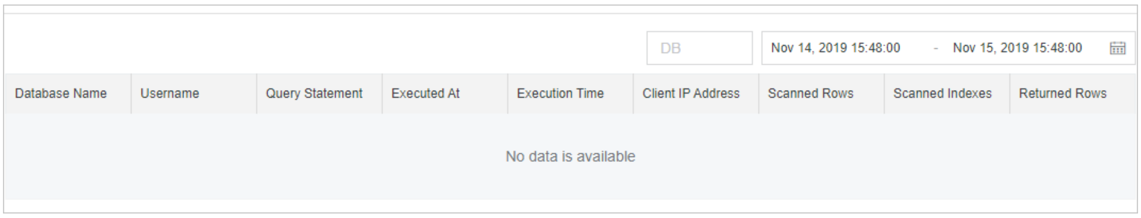
A replica set instance with more than three nodes or a sharded cluster instance is created.

Precautions

The retention period for slow query logs is 72 hours.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Logs > Slow Query Logs**.
6. View slow query logs based on instance types:
   - Replica set instances: You can select the database name and time range to query the corresponding slow query logs.

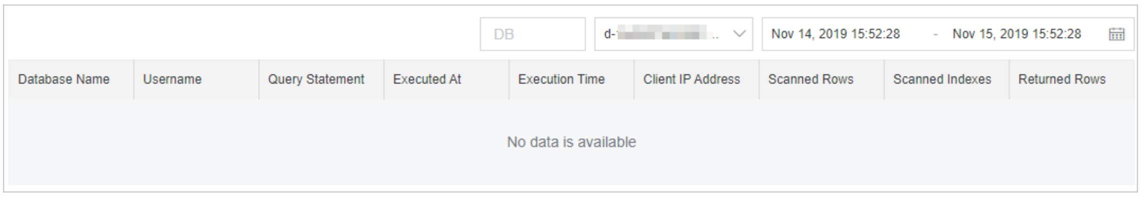


DB | Nov 14, 2019 15:48:00 - Nov 15, 2019 15:48:00

Database Name	Username	Query Statement	Executed At	Execution Time	Client IP Address	Scanned Rows	Scanned Indexes	Returned Rows

No data is available

- Sharded cluster instances: You can select the database name, shard node ID and time range to view the corresponding slow query logs.



DB | d-[redacted] | Nov 14, 2019 15:52:28 - Nov 15, 2019 15:52:28

Database Name	Username	Query Statement	Executed At	Execution Time	Client IP Address	Scanned Rows	Scanned Indexes	Returned Rows

No data is available

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

13.3. View error logs

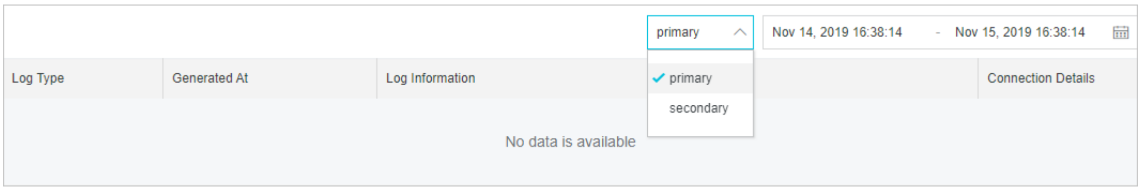
You can query the error logs of an instance in the console.

Prerequisites

A replica set instance with more than three nodes or a sharded cluster instance is created.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Logs > Error Logs**.
6. Perform the following operations based on instance types:
   - Replica set instances: You can select the node role and time range to query the corresponding error logs.



primary | Nov 14, 2019 16:38:14 - Nov 15, 2019 16:38:14

Log Type	Generated At	Log Information	Connection Details

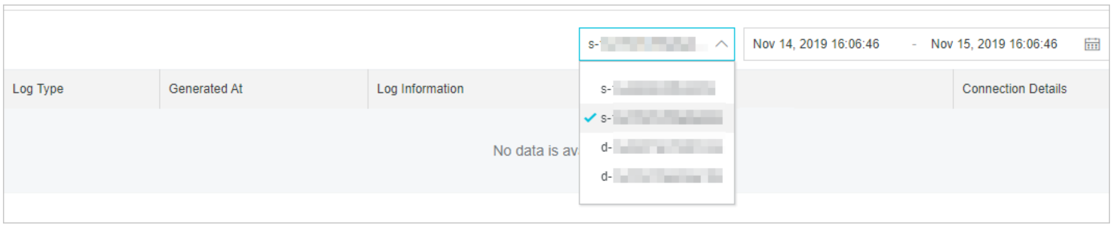
No data is available

   - Sharded cluster instances: You can query error logs of Mongos or shard nodes.

> Note Mongos nodes have IDs prefixed with `s-` . Shard nodes have IDs prefixed with `d-` .

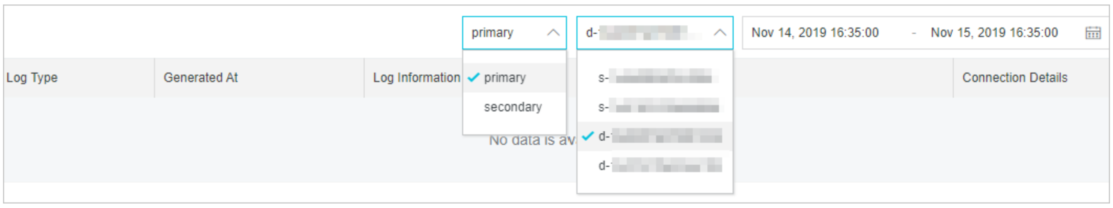
- Query the error logs of a Mongos node.

  You can select the Mongos node ID and time range to query the error logs.



- Query the error logs of a shard node.

  You can select the shard node ID, role, and time range to query the error logs.



Related information

- Enable the new audit log feature
- Overview
- Slow query logs

13.4. View operation logs

You can query the operation logs of an instance in the console to check its running status.

Prerequisites

A replica set instance with more than three nodes or a sharded cluster instance is created.

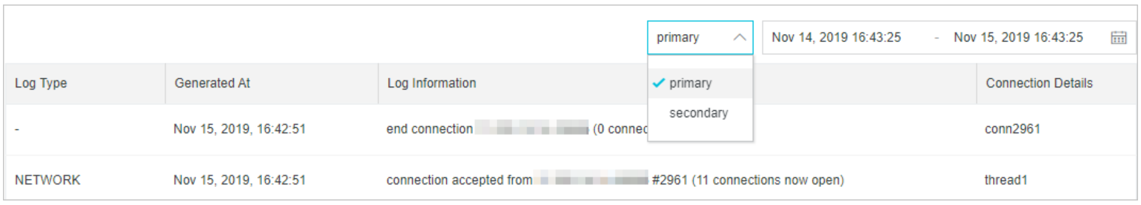
Precautions

The retention period for slow query logs is 72 hours.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Logs > Running Logs**.

6. Perform the following operations based on instance types:
   - Replica set instances: You can select the node role and time range to query the corresponding operation logs.



primary ☑ primary / secondary | Nov 14, 2019 16:43:25 - Nov 15, 2019 16:43:25

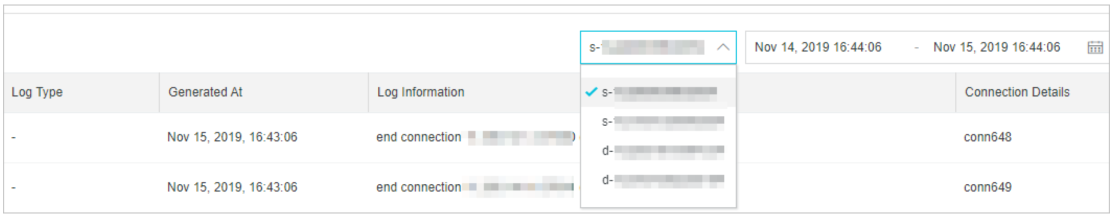
Log Type	Generated At	Log Information	Connection Details
-	Nov 15, 2019, 16:42:51	end connection [redacted] (0 connec	conn2961
NETWORK	Nov 15, 2019, 16:42:51	connection accepted from [redacted] #2961 (11 connections now open)	thread1

   - Sharded cluster instances: You can query operation logs of Mongos or shard nodes.

> **Note** Mongos nodes have IDs prefixed with `s-`. Shard nodes have IDs prefixed with `d-`.

     - Query the operation logs of a Mongos node.

       You can select the Mongos node ID and time range to query the operation logs.

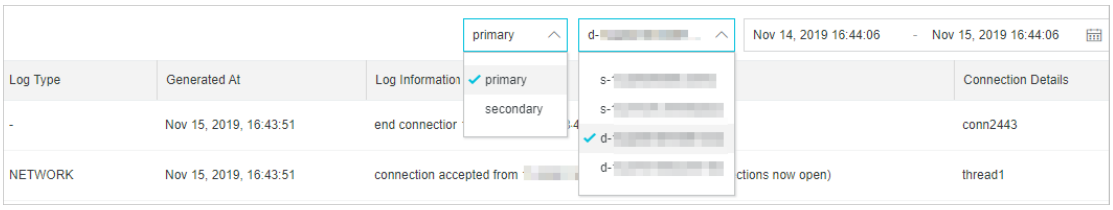


s-[redacted] ☑ s-[redacted] / s-[redacted] / d-[redacted] / d-[redacted] | Nov 14, 2019 16:44:06 - Nov 15, 2019 16:44:06

Log Type	Generated At	Log Information	Connection Details
-	Nov 15, 2019, 16:43:06	end connection [redacted])	conn648
-	Nov 15, 2019, 16:43:06	end connection [redacted]	conn649

     - Query the operation logs of a shard node.

       You can select the shard node ID, role, and time range to query the operation logs.



primary ☑ primary / secondary | d-[redacted] s-[redacted] / s-[redacted] / ☑ d-[redacted] / d-[redacted] | Nov 14, 2019 16:44:06 - Nov 15, 2019 16:44:06

Log Type	Generated At	Log Information	Connection Details
-	Nov 15, 2019, 16:43:51	end connectior [redacted]	conn2443
NETWORK	Nov 15, 2019, 16:43:51	connection accepted from [redacted] ctions now open)	thread1

Related information

- Enable the new audit log feature
- Overview
- Slow query logs

14.Data migration and synchronization

14.1. Overview

This topic provides an overview of the solutions to migrate and synchronize data to or from an ApsaraDB for MongoDB database in different scenarios.

Data migration solutions

By using Data Transmission Service (DTS), you can fully or incrementally migrate the data of a MongoDB database. This can achieve smooth data migration from a MongoDB database to the cloud without affecting businesses.

ApsaraDB for MongoDB supports full data migration by using the official mongodump and mongorestore tools.

In addition, ApsaraDB for MongoDB allows you to migrate data from the Cloud to an on-premises database by using a physical or logical backup file.

Migration scenario	Source database architecture	Documentation
Migrate data from a self-mamanged or on-premises database to Alibaba Cloud	Standalone instance	Migrate self-managed standalone MongoDB databases to Alibaba Cloud by using DTS
		Migrate self-managed MongoDB databases to standalone instances by using tools provided by MongoDB
	Replica set instance	Migrate the replica set of a user-created MongoDB database to ApsaraDB for MongoDB by using DTS
		Migrate user-created MongoDB databases to Alibaba Cloud by using the built-in commands of MongoDB
	Sharded cluster instance	Migrate a user-created sharded MongoDB database to ApsaraDB for MongoDB by using DTS
		Migrate a self-managed MongoDB database to ApsaraDB for MongoDB by using tools provided by MongoDB
Migrate data from a database on a third-party cloud platform to Alibaba Cloud	N/A	Migrate data from Amazon DynamoDB to ApsaraDB for MongoDB by using mongoimport
	Replica set or sharded cluster instance	• Migrate data from MongoDB Atlas to ApsaraDB for MongoDB by using mongodump and mongorestore • Migrate data from a MongoDB Atlas database to Alibaba Cloud

Migration scenario	Source database architecture	Documentation
Migrate data between ApsaraDB for MongoDB instances	Replica set instance	Migrate data from a replica set MongoDB instance to a sharded cluster instance
	Standalone instance	Migrate data from a standalone instance to a replica set or sharded cluster instance
	Standalone or replica set instance	Migrate data between ApsaraDB for MongoDB instances created by different Alibaba Cloud accounts
Migrate data from an ApsaraDB for MongoDB instance to a self-managed or on-premises MongoDB database	Replica set instance	Restore data of an ApsaraDB for MongoDB instance to self-managed MongoDB databases by using logical backup
		将MongoDB物理备份文件恢复至自建数据库

Data synchronization solutions

You can use the MongoShake tool developed by Alibaba Cloud to synchronize data between MongoDB databases.

Synchronization scenario	Tool	Documentation
Synchronize data to an existing instance	MongoShake	Use MongoShake to implement one-way synchronization between ApsaraDB for MongoDB replica set instances

14.2. Migrate an ECS-hosted self-managed MongoDB database that uses the standalone or replica set architecture to ApsaraDB for MongoDB

This topic describes how to use Data Transmission Service (DTS) to migrate a self-managed MongoDB database that hosts on Elastic Compute Service (ECS) and uses the standalone or replica set architecture to ApsaraDB for MongoDB. DTS allows you to migrate historical and incremental data without business interruptions.

Prerequisites

- The version of the self-managed MongoDB database is 3.0, 3.2, 3.4, 3.6, 4.0, or 4.2.
- The storage space of the ApsaraDB for MongoDB instance is larger than the size of the self-managed MongoDB database.

Usage notes

- We recommend that you migrate your data during off-peak hours to avoid business interruptions.
- The config database is an internal database. Do not migrate its data unless otherwise specified.
- If the self-managed MongoDB database and the destination ApsaraDB for MongoDB instance run different MongoDB versions or storage engines, make sure that your applications can run on both databases. For more information about the MongoDB versions and storage engines that ApsaraDB for MongoDB supports, see MongoDB versions and storage engines.

Billing information

Migration Types	Instance configurations	Internet traffic
Full data migration	Free of charge	Free of charge
Incremental data migration	Charged, For more information, see Data Transmission Service (DTS) pricing.	Free of charge

Migration type description

- Full data migration: All data of the migration objects is migrated from the source instance to the destination instance..

  > Note Data migration is supported at the database, collection, and index levels.

- Incremental data migration: Updated data of the migration objects is synchronized from the source instance to the destination instance..

  > Note
  > - The create and delete operations for databases, collections, and indexes can be synchronized.
  > - The create, delete, and update operations on documents can be synchronized.

Permissions required for database accounts

Database	Full data migration	Incremental data migration
Self-managed MongoDB database on ECS	Read permissions on the source database	Read permissions on the source database, admin database, and local database
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination databases	Read/write permissions on the destination databases

For more information about how to create and authorize a database account, see the following topics:

- Manage user permissions on MongoDB databases for an ApsaraDB for MongoDB instance.
- db.createUser() for a self-managed MongoDB database.

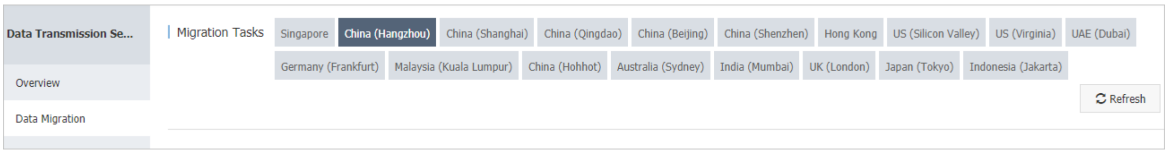
Preparations before data migration

Skip this step if the self-managed MongoDB database on ECS uses the replica set architecture.

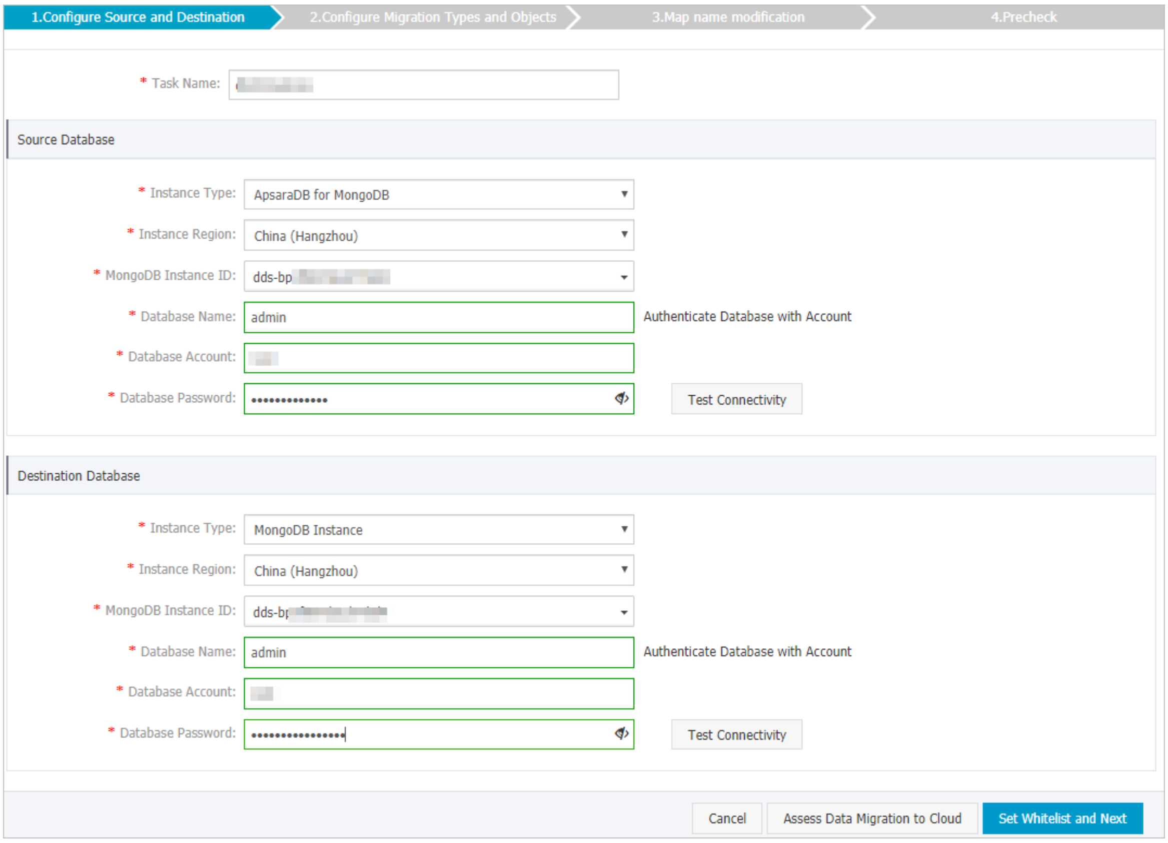
If the self-managed MongoDB database on ECS uses the standalone architecture, enable oplog for the database before you migrate incremental data. For more information, see Preparations for incremental data migration.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. In the **Migration Tasks** section, select the region where the ApsaraDB for MongoDB instance is deployed.



4. In the upper-right corner, click **Create Migration Task**.
5. Configure the source and destination databases.



1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name: [redacted]

Source Database

* Instance Type: ApsaraDB for MongoDB

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin Authenticate Database with Account

* Database Account: [redacted]

* Database Password: •••••••••••••

Destination Database

* Instance Type: MongoDB Instance

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin Authenticate Database with Account

* Database Account: [redacted]

* Database Password: ••••••••••••••••

Section	Parameter	Description

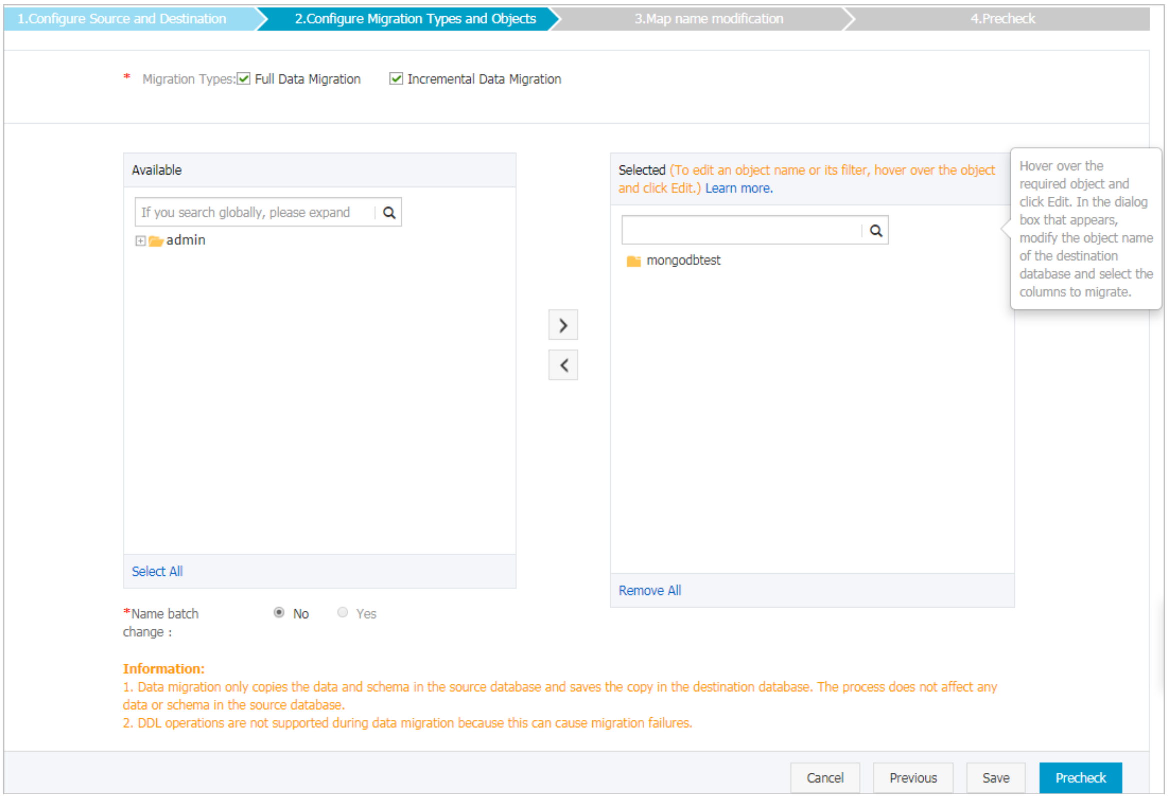
Section	Parameter	Description
Task Name	N/A	○ DTS automatically generates a task name. You do not need to use a unique task name. ○ We recommend that you use an informative name for easy identification.
Source Database	Instance Type	Select **User-Created Database in ECS Instance**.
	Instance Region	The region where the ECS instance is deployed.
	ECS Instance ID	The ID of the ECS instance on which the self-managed MongoDB database is deployed.
	Database Type	The type of the database. In this example, select **MongoDB** from the drop-down list.
	Port Number	The service port of the self-managed MongoDB database.
	Database Name	The name of the destination database to which the database account belongs.
	Database Account	The username of the database account used to connect to the source database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	The password of the database account used to connect to the source database. **Note** After you specify the source database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message appears. If the information is incorrect, the **Failed** message appears. In this case, you must click **Check** next to the **Failed** message to modify the information.
	Instance Type	The type of the instance. In this example, select **MongoDB Instance**.
	Instance Region	The region where the ApsaraDB for MongoDB instance is deployed.
	The ID of the ApsaraDB for MongoDB instance.	The ID of the destination ApsaraDB for MongoDB instance.
	Database Name	The name of the destination database to which the database account belongs.

Section	Parameter	Description
Destination Database	Database Account	The username of the database account used to connect to the destination database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	The password of the database account used to connect to the destination database. ? **Note** After you specify the destination database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message appears. If the information is incorrect, the **Failed** message appears. In this case, you must click **Check** next to the **Failed** message to modify the information.

6. In the lower-right corner of the page, click **Set Whitelist and Next**.

> ? **Note** The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for MongoDB instance. This ensures that DTS servers can connect to the source and destination instances. After data migration is complete, you can remove the CIDR blocks of DTS servers from the whitelists. For more information, seeManage security group rules and Configure a whitelist for an ApsaraDB for MongoDB instance.

7. Select the migration types and the objects to be migrated.



* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

If you search globally, please expand

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

mongodbtest

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description
Migration Types	○ To migrate all data, select **Full Data Migration**. **Note** To ensure data consistency, do not write new data to the source MongoDB database during full data migration. ○ To migrate data with minimal downtime, select both **Full Data Migration** and **Incremental Data Migration**. **Note** Before you migrate incremental data from a standalone self-managed MongoDB database, you must enable oplog for the database. For more information, see Preparations before data migration.

Parameter	Description
Migration objects	○ Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. Note ▪ Data in the admin and local databases cannot be migrated. ▪ The config database is an internal database. Do not migrate its data unless otherwise specified. ○ A migration object can be a database, collection, or function. ○ By default, the names of the objects to be migrated remain unchanged after the migration. You can change the names of the objects in the destination ApsaraDB for MongoDB instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

> Note
>
> - A precheck is performed before the migration task starts. The migration task starts only after the precheck succeeds.
> - If the precheck fails, click the ⓘ icon for each failed check item to view their details.
>
>   Perform a precheck again after the failures are fixed.

9. After the precheck succeeds, click **Next**.

10.

11. Click **Buy and Start** to start the migration task.

    - Full data migration

      Do not manually end a migration task. If you do so, the system may fail to migrate all data of the database. Wait until the migration task is complete.

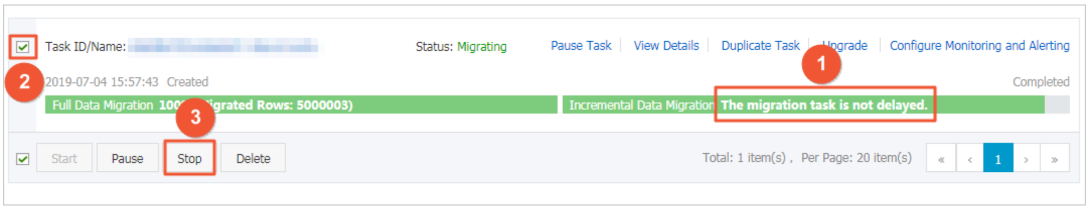
    - Incremental data migration

      An incremental data migration task does not automatically end. You need to manually end the task.

      > Note Select an appropriate point in time to manually end a migration task. For example, you can end the migration task during off-peak hours or before you switch over your business to the destination ApsaraDB for MongoDB instance.

      a. When the task progress bar displays **Incremental Data Migration** and **The migration task is not delayed**, stop writing data to the source database for a few minutes. Wait until the progress bar displays the delay time of the incremental data migration next to **Incremental Data Migration**.

b. After the status of **Incremental Data Migration** changes to **The migration task is not delayed**, manually end the migration task.



Task ID/Name: [redacted] Status: Migrating

2019-07-04 15:57:43 Created

Full Data Migration 100% (Migrated Rows: 5000003)

Incremental Data Migration The migration task is not delayed.

Completed

Total: 1 item(s) , Per Page: 20 item(s)

12. Switch over your business to the destination ApsaraDB for MongoDB instance.

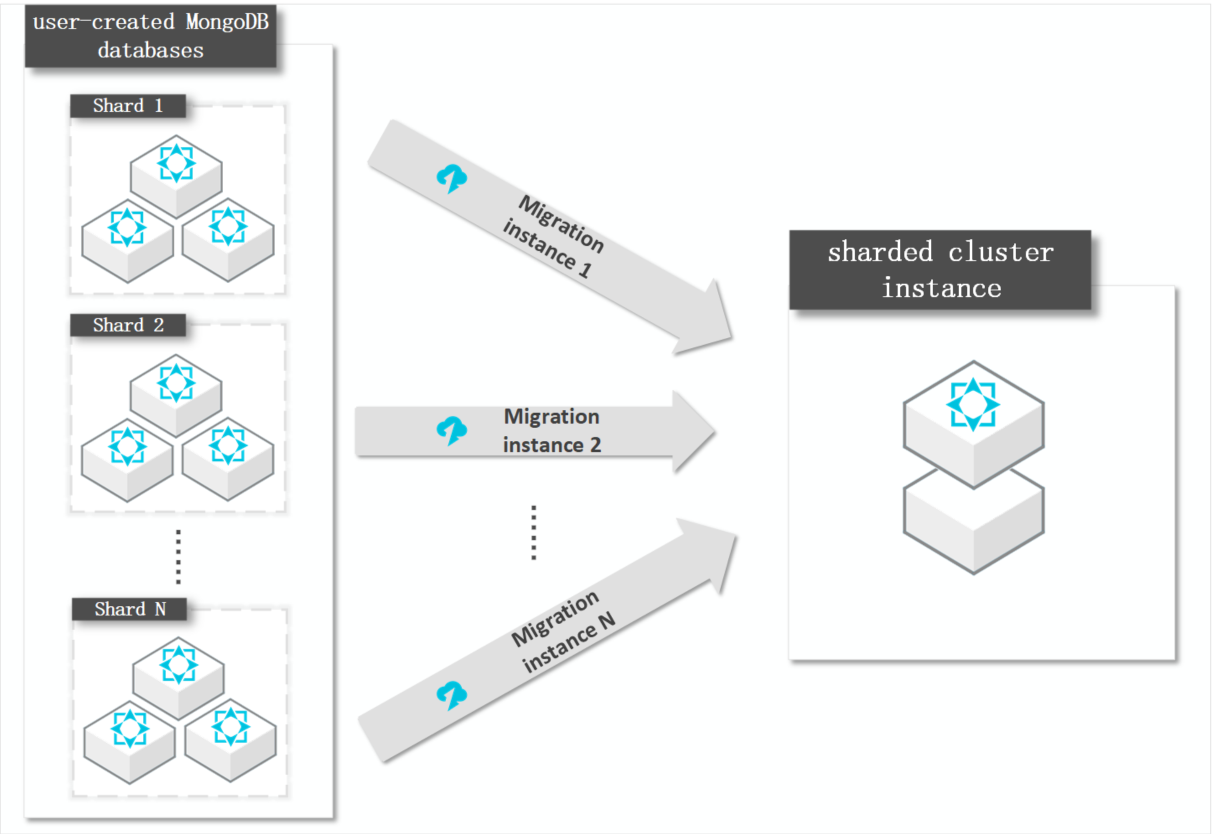
14.3. Migrate the shards of a self-managed MongoDB database that hosts on ECS to ApsaraDB for MongoDB

This topic describes how to use Data Transmission Service (DTS) to migrate the shards of a self-managed MongoDB database that hosts on Elastic Compute Service (ECS) to an ApsaraDB for MongoDB instance. DTS allows you to migrate historical and incremental data without business interruptions.

How it works

DTS migrates a user-created MongoDB database by migrating each shard in the instance. You must create a data migration task for each shard.

> **Note** The distribution of migrated data in the destination instance depends on the shard key that you specify. For more information, see Configure sharding to maximize the performance of shards.



Prerequisites

- The version of the self-managed MongoDB database is 3.0, 3.2, 3.4, 3.6, 4.0, or 4.2.
- Each shard in the ApsaraDB for MongoDB instance has sufficient storage space.

> Note For example, a self-managed MongoDB database has three shards, and one of these shards occupies the most storage space, which is 500 GB. In this case, the storage space of each shard in the ApsaraDB for MongoDB instance must be greater than 500 GB.

Usage notes

- DTS uses resources of the source and destination instances during full data migration. This may increase the load of the database server. If the data volume is large or the specification is low, the database server may become unavailable. We recommend that you migrate user-created MongoDB databases during off-peak hours.
- If the source user-created MongoDB database and the destination ApsaraDB for MongoDB instance run different MongoDB versions or storage engines, ensure that your applications can run on both instances. For more information about MongoDB versions and storage engines that are supported by ApsaraDB for MongoDB, see MongoDB versions and storage engines.

Billing information

Migration Types	Instance configurations	Internet traffic
Full data migration	Free of charge	Free of charge

Migration Types	Instance configurations	Internet traffic
Incremental data migration	Charged, For more information, see Data Transmission Service (DTS) pricing.	Free of charge

Migration types

- Full data migration: All historical data in the source instance is migrated to the destination instance.

> Note Data migration is supported at the database, collection, and index levels.

- Incremental data migration: After full data migration, incremental data is synchronized to the destination instance.

> Note
> - The create and delete operations on databases, collections, and indexes can also be synchronized.
> - The create, delete, and update operations on documents can be synchronized.

Permissions required for database accounts

Database	Full data migration	Incremental data migration
Self-managed MongoDB database on ECS	Read permissions on the source database	Read permissions on the source database, admin database, and local database
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination databases	Read/write permissions on the destination databases

For more information about how to create and authorize a database account, see the following topics:

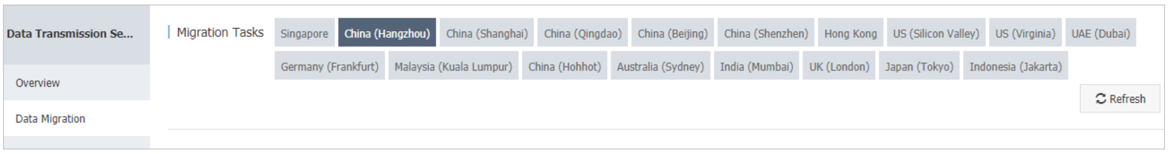
- Manage MongoDB users through DMS for an ApsaraDB for MongoDB instance.
- db.createUser() for a self-managed MongoDB database.

Preparations before data migration

Disable the balancer for the source database and delete orphaned documents. For more information, see Migrate a user-created sharded MongoDB database to ApsaraDB for MongoDB by using DTS.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. In the **Migration Tasks** section, select the region where the ApsaraDB for MongoDB instance is deployed.



4. In the upper-right corner, click **Create Migration Task**.
5. Click Create Migration Task. In the **Configure Source and Destination** step, configure the source and destination databases for the migration task.

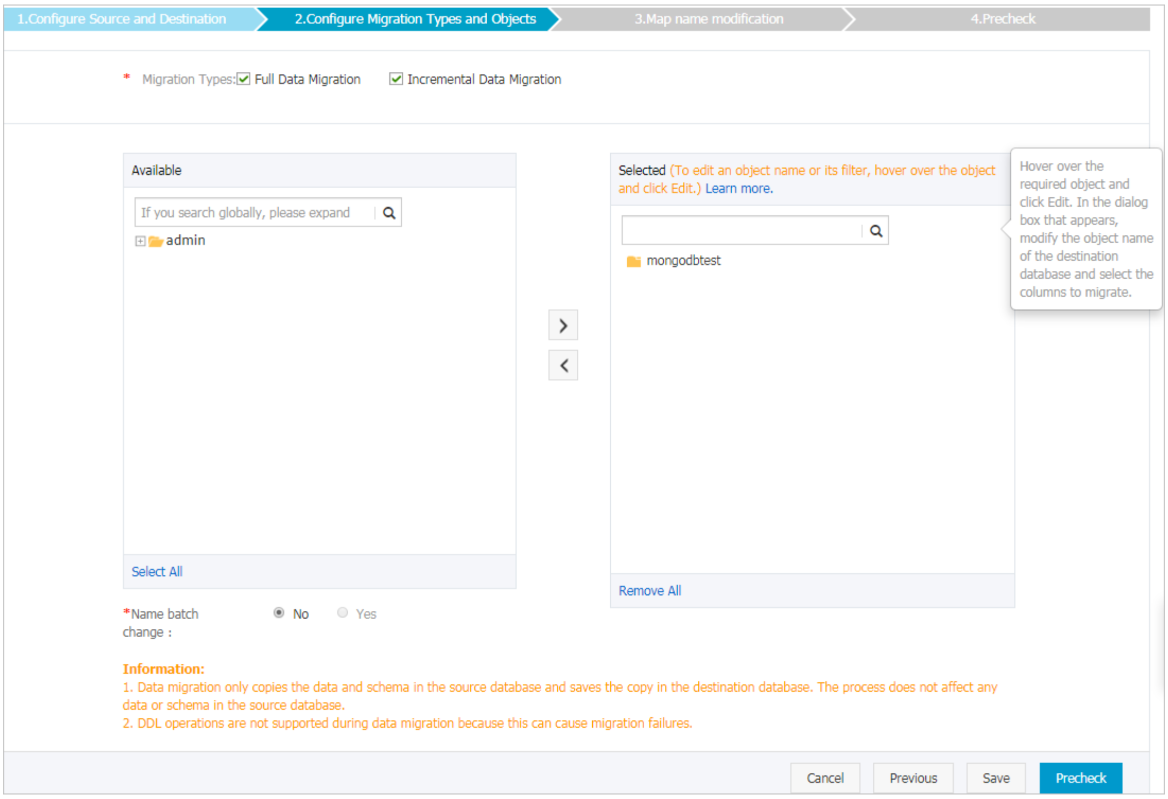
Section	Parameter	Description
N/A	-	○ DTS automatically generates a task name. You do not need to use a unique task name. ○ We recommend that you use an informative name for easy identification.
Source Database	Instance Type	Select **User-Created Database in ECS Instance**.
	Instance Region	The region where the ECS instance is deployed.
	ECS Instance ID	The ID of the ECS instance. DTS migrates each shard of the source database in turn. In this example, enter the ID of the ECS instance on which the first shard is deployed. For the second migration task, enter the ID of the ECS instance on which the second shard is deployed. Repeat this operation until all shards are migrated.
	Database Type	The type of the database. In this example, select **MongoDB** from the drop-down list.
	Port Number	The service port of the shard. In this example, enter the service port of the first shard. For the second migration task, enter the service port of the second shard. Repeat this operation until all shards are migrated.
	Database Name	The name of the destination database to which the database account belongs.
	Database Account	The username of the database account used to connect to the source database. For more information about the permissions that are required for the account, see Permissions required for database accounts.

Section	Parameter	Description
	Database Password	The password of the database account used to connect to the source database. ⓘ **Note** After you specify the source database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
Destination Database	Select the instance type.	The type of the instance. In this example, select **MongoDB Instance**.
	Instance Region	The region where the ApsaraDB for MongoDB instance is deployed.
	MongoDB Instance ID	Select the ID of the destination ApsaraDB for MongoDB instance.
	Database Name	The name of the destination database to which the database account belongs.
	Database Account	The username of the database account used to connect to the destination database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	The password of the database account used to connect to the destination database. ⓘ **Note** After you specify the destination database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message appears. If the information is incorrect, the **Failed** message appears. In this case, you must click **Check** next to the **Failed** message to modify the information.

6. In the lower-right corner of the page, click **Set Whitelist and Next**.

> ⓘ **Note** The Classless CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for MongoDB instance. This ensures that DTS servers can connect to the source and destination instances. After data migration is complete, you can remove the CIDR blocks of DTS servers from the whitelists. For more information, seeManage security group rules and Configure a whitelist for an ApsaraDB for MongoDB instance.

7. Select the migration types and the objects to be migrated.



1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

Available

If you search globally, please expand

⊞ admin

Select All

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

mongodbtest

Remove All

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description
Migration Types	◦ To migrate all data, select **Full Data Migration**. Note To ensure data consistency, do not write new data to the source MongoDB database during full data migration. ◦ To migrate data with minimal downtime, select both **Full Data Migration** and **Incremental Data Migration**.
Migration objects	◦ Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. Note ▪ Data in the admin and local databases cannot be migrated. ▪ The config database is an internal database. Do not migrate its data unless otherwise specified. ◦ A migration object can be a database, collection, or function. ◦ By default, the names of the objects to be migrated remain unchanged after the migration. You can change the names of the objects in the destination ApsaraDB for MongoDB instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

   > Note
   >
   > - A precheck is performed before the migration task starts. The migration task starts only after the precheck succeeds.
   > - If the precheck fails, click the ⓘ icon for each failed check item to view their details.
   >
   >   Perform a precheck again after the failures are fixed.

9. After the precheck succeeds, click **Next**.
10.
11. Click **Buy and Start** to start the migration task.
12. Repeat step 4 to step 11 to create migration tasks for the remaining shards.
13. Stop the data migration task.
    - Full data migration

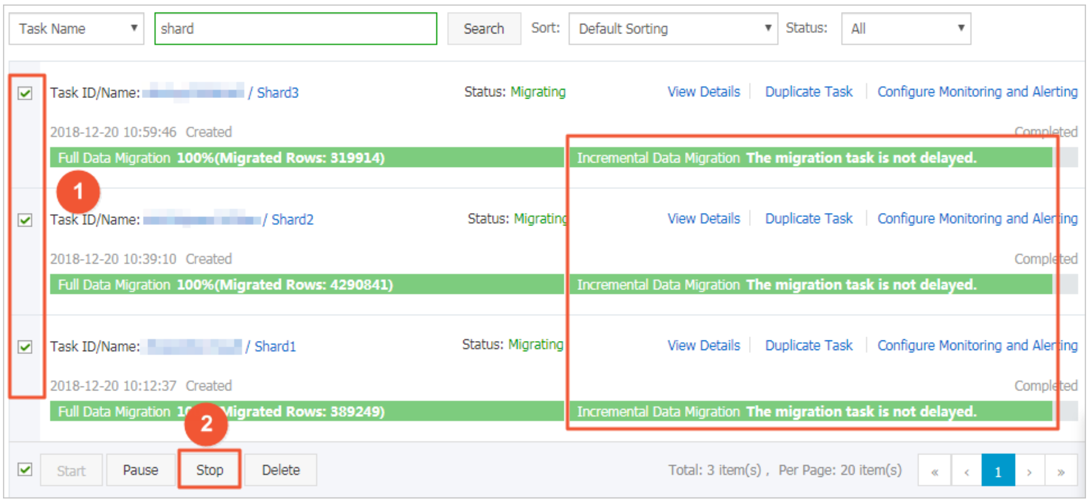
      Do not manually stop a task during full data migration. Otherwise, the system may fail to perform a full data migration. Wait until the data migration task automatically stops.
    - Incremental data migration

      An incremental data migration task does not automatically stop. You must manually stop the migration task.

      > Note Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the destination instance.

      a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

      b. After the status of **Incremental Data Migration** changes to **The migration task is not delayed**, stop the migration task.



Task Name ▾ | shard | Search | Sort: Default Sorting ▾ | Status: All ▾

☑ Task ID/Name: [redacted] / Shard3 — Status: Migrating — View Details | Duplicate Task | Configure Monitoring and Alerting
2018-12-20 10:59:46 Created — Completed
Full Data Migration 100%(Migrated Rows: 319914) — Incremental Data Migration The migration task is not delayed.

☑ Task ID/Name: [redacted] / Shard2 — Status: Migrating — View Details | Duplicate Task | Configure Monitoring and Alerting
2018-12-20 10:39:10 Created — Completed
Full Data Migration 100%(Migrated Rows: 4290841) — Incremental Data Migration The migration task is not delayed.

☑ Task ID/Name: [redacted] / Shard1 — Status: Migrating — View Details | Duplicate Task | Configure Monitoring and Alerting
2018-12-20 10:12:37 Created — Completed
Full Data Migration 1 Migrated Rows: 389249) — Incremental Data Migration The migration task is not delayed.

☑ Start | Pause | Stop | Delete — Total: 3 item(s) , Per Page: 20 item(s) « ‹ 1 › »

14. Switch over your business to the destination ApsaraDB for MongoDB instance.

14.4. Migrate data between ApsaraDB for MongoDB instances

14.4.1. Migrate data from a replica set MongoDB instance to a sharded cluster instance

This topic describes how to migrate data from a replica set MongoDB instance to a sharded cluster instance by using Data Transmission Service (DTS). DTS supports full data migration and incremental data migration. When you migrate data between ApsaraDB for MongoDB instances, you can select both of the supported migration types to ensure service continuity.

Prerequisites

Each shard in the destination sharded cluster instance has sufficient storage space.

Precautions

- During full data migration, DTS uses read and write resources of the source and destination databases. This may increase the load of the database server. If you migrate a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- If the source and destination ApsaraDB for MongoDB instances have different versions or storage engines, make sure that the versions or storage engines are compatible. For more information, see MongoDB versions and storage engines.
- To ensures better performance and stability of the instance, the system will upgrade the minor version to the latest version by default If the minor version of your instance expires or is not included in the maintenance list and the instance is upgraded, migrated, changed, Created from a backup, Created by point-in-time, or performed Restore data to a new ApsaraDB for MongoDB instance.

Billing

Migration type	Instance configuration	Internet traffic
Full data migration	Free of charge.	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see DTS pricing.
Incremental data migration	Charged. For more information, see DTS pricing.	

Migration types

Migration type	Description

Migration type	Description
Full data migration	DTS migrates all historical data of the required objects from the source MongoDB database to the destination MongoDB database. Note The following types of objects are supported: database, collection, and index.
Incremental data migration	After full data migration is complete, DTS synchronizes incremental data from the source MongoDB database to the destination MongoDB database. Note • The create and delete operations that are performed on databases, collections, and indexes can be synchronized. • The create, delete, and update operations that are performed on documents can be synchronized.

Permissions required for database accounts

Instance	Full data migration	Incremental data migration
ApsaraDB for MongoDB replica set instance	The read permission on the source database	The read permission on the source database, admin database, and local database
ApsaraDB for MongoDB sharded cluster instance	The read/write permissions on the destination database	The read/write permissions on the destination database

Note For more information about how to create and authorize a database account, see Use DMS to manage MongoDB users.

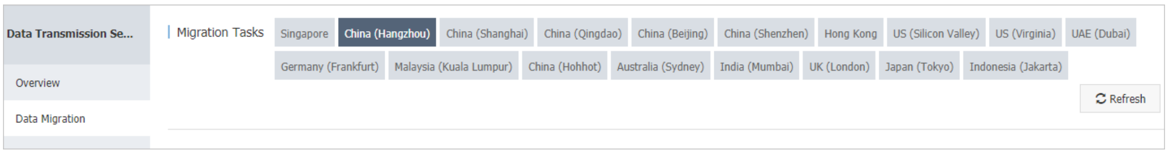
Before you begin

Create databases and collections to be sharded in the destination ApsaraDB for MongoDB instance, and configure data sharding based on your business requirements. For more information, see Configure sharding to maximize the performance of shards.

Note After you configure sharding for a cluster, data will not be migrated to the same shard. This maximizes the performance of the sharded cluster.

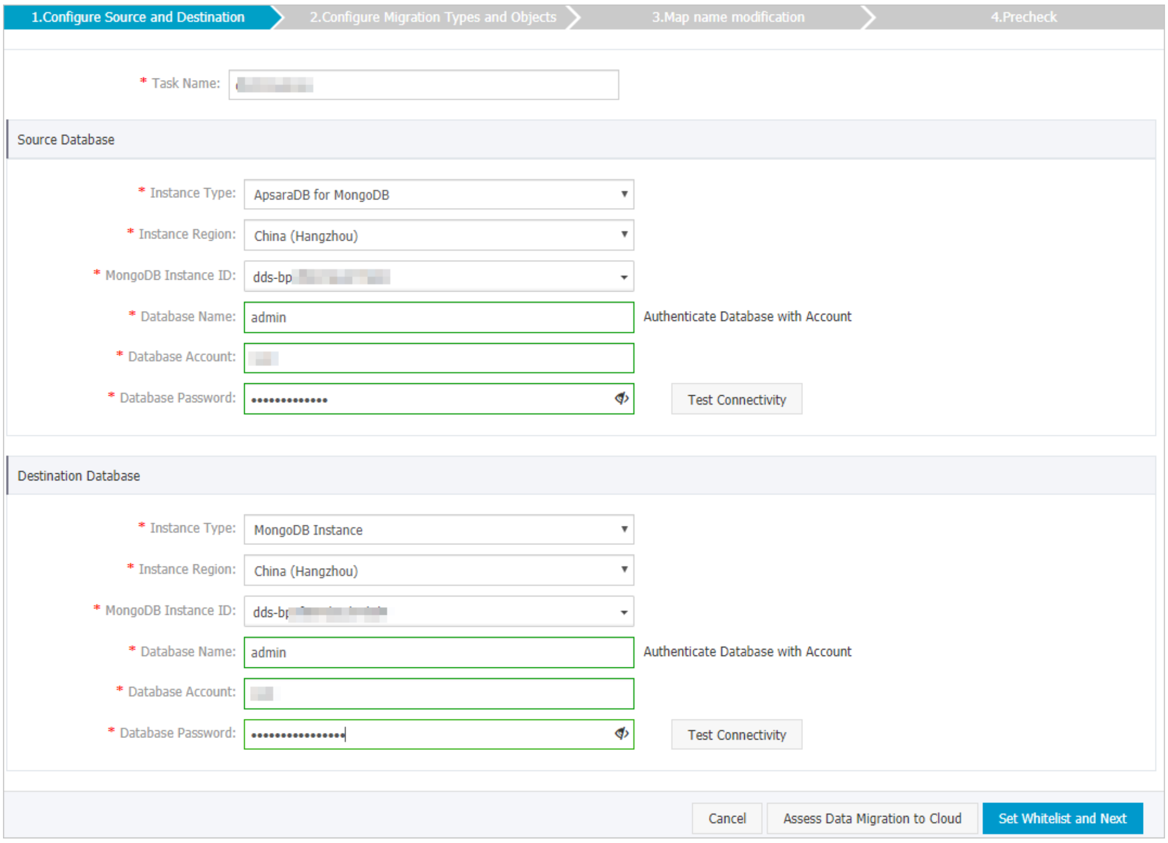
Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination ApsaraDB for MongoDB instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.

5. Configure the source and destination databases for the data migration task.


1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name: [redacted]

Source Database

* Instance Type: ApsaraDB for MongoDB

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin — Authenticate Database with Account

* Database Account: [redacted]

* Database Password: •••••••••••••

Destination Database

* Instance Type: MongoDB Instance

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin — Authenticate Database with Account

* Database Account: [redacted]

* Database Password: ••••••••••••••••

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select **ApsaraDB for MongoDB**.
	Instance Region	Select the region where the source ApsaraDB for MongoDB instance resides.
	MongoDB Instance ID	Select the ID of the source ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. The database account is created in this database. **Note** If the database account is root, enter admin.

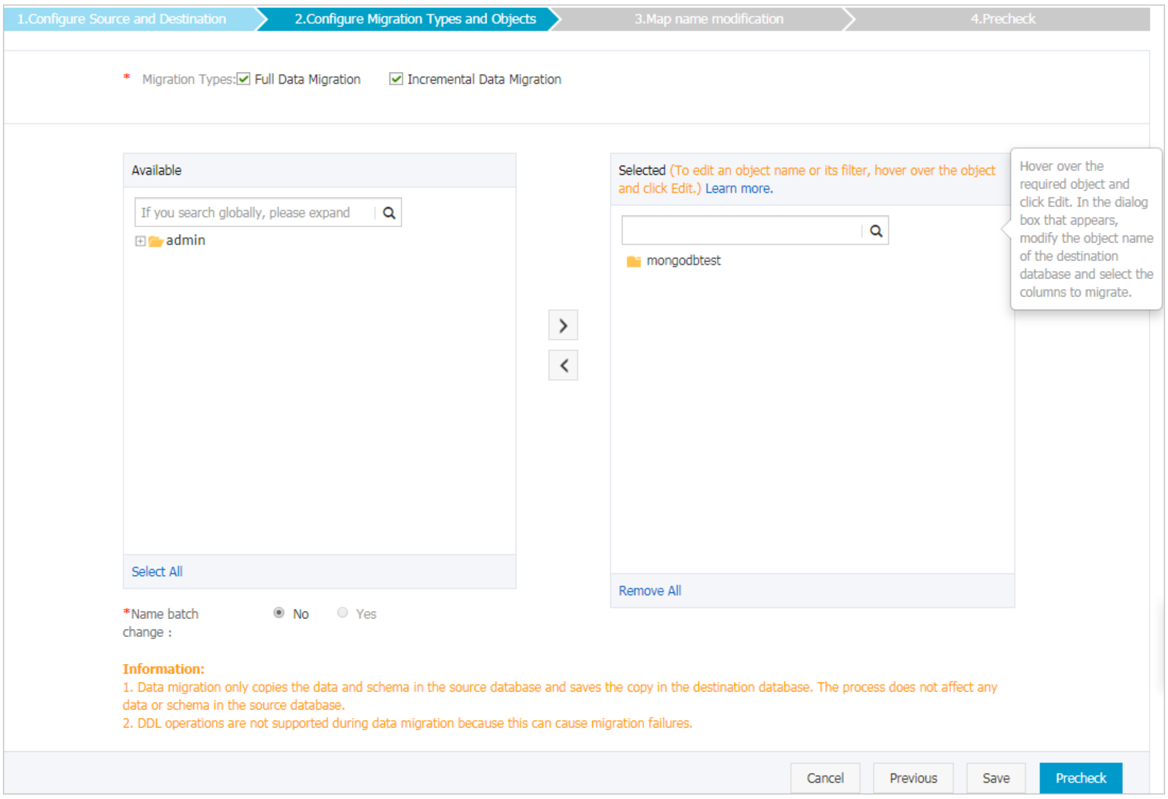
Section	Parameter	Description
Source Database	Database Account	Enter the database account of the source ApsaraDB for MongoDB instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the source database account. Note After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
Destination Database	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region where the destination ApsaraDB for MongoDB instance resides.
	MongoDB Instance ID	Select the ID of the destination ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. The database account is created in this database. Note If the database account is root, enter admin.
	Database Account	Enter the database account of the destination ApsaraDB for MongoDB instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.

Section	Parameter	Description
	Database Password	Enter the password of the destination database account. **Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

6. In the lower-right corner, click **Set Whitelist and Next**.

   > **Note** DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination ApsaraDB for MongoDB instances. This ensures that DTS servers can connect to the source and destination ApsaraDB for MongoDB instances. After data migration is complete, you can remove the CIDR blocks of DTS servers from the whitelists. For more information, see Configure a whitelist for a sharded cluster instance.

7. Select the migration types and objects to be migrated.



* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:

1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck
Available
If you search globally, please expand
admin
Select All
mongodbtest
Remove All
Cancel
Previous
Save
Precheck

Parameter	Description
Migration Types	◦ To perform only full data migration, select only **Full Data Migration**. ◦ To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source ApsaraDB for MongoDB database during full data migration. This ensures data consistency between the source and destination databases.
Objects	◦ Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. **Note** You cannot migrate data from the admin or local database. ◦ You can select databases, collections, or functions as the objects to be migrated. ◦ After an object is migrated to the destination database, the name of the object remains unchanged. You can change the names of the objects that are migrated to the destination database by using the object name mapping feature. For more information about how to use this feature, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

   Note

   - Before you can start the data migration task, a precheck is performed. You can start the data migration task only after the task passes the precheck.
   - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

9. After the task passes the precheck, click **Next**.
10. In the **Confirm Settings** dialog box, specify the **Channel Specification** and select **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the migration task.

    - Full data migration

      We recommend that you do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the migration task automatically stops.

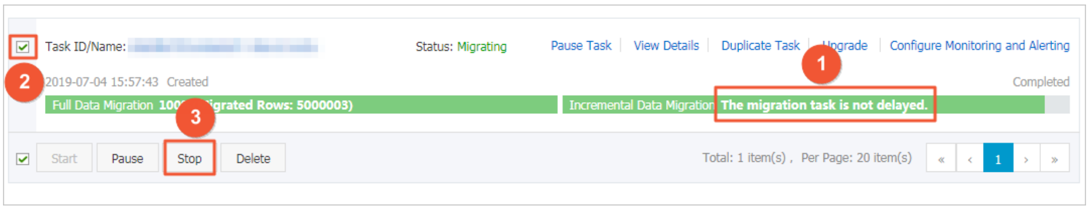
    - Incremental data migration

      An incremental data migration task does not automatically stop. You must manually stop the migration task.

> ⓘ **Note** Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the destination ApsaraDB for MongoDB instance.

a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

b. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



☑ Task ID/Name: [redacted] Status: Migrating

2019-07-04 15:57:43 Created

Full Data Migration 100% (Migrated Rows: 5000003)

Incremental Data Migration The migration task is not delayed.

Completed

Total: 1 item(s), Per Page: 20 item(s)

12. Switch your workloads to the destination ApsaraDB for MongoDB instance.

14.4.2. Migrate data from a standalone instance to a replica set or sharded cluster instance

ApsaraDB for MongoDB provides standalone instances, replica set instances, and sharded cluster instances. Standalone instances are designed to store non-core enterprise data, such as data in development and testing scenarios. Replica set instances and sharded cluster instances are more suitable for production scenarios. This topic describes how to migrate data from a standalone instance to a replica set or sharded cluster instance by using Data Transmission Service (DTS).

Prerequisites

The available storage space of the destination instance is larger than the total size of the data in the source instance.

Precautions

- DTS uses resources of the source and destination instances during full data migration. This may increase the load of the database server. If the data volume is large or the specification is low, the database server may become unavailable. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- DTS does not support incremental data migration from a standalone instance. To ensure data consistency, do not write data to the source instance during full data migration.
- If the source and destination ApsaraDB for MongoDB instances have different database engine versions or storage engines, make sure that the versions or storage engines are compatible. For more information, see MongoDB versions and storage engines.

Billing for data migration

Migration type	Instance configuration fee	Internet traffic fee
Support for full data migration	Free of charge	Free of charge

Migration types

Full data migration: DTS migrates all historical data of the required objects from the source MongoDB database to the destination MongoDB database.

> **Note** The following types of objects are supported: database, collection, and index.

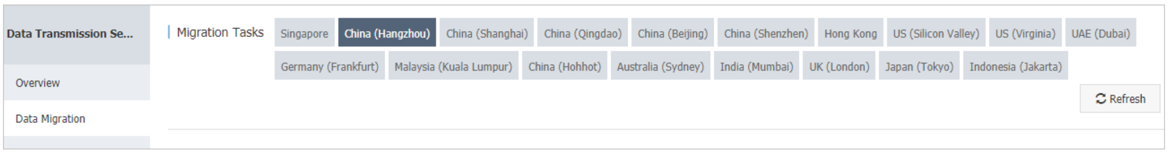
Required database account permissions

Instance	Permissions
Source ApsaraDB for MongoDB instance	Read permissions on the source database
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination databases

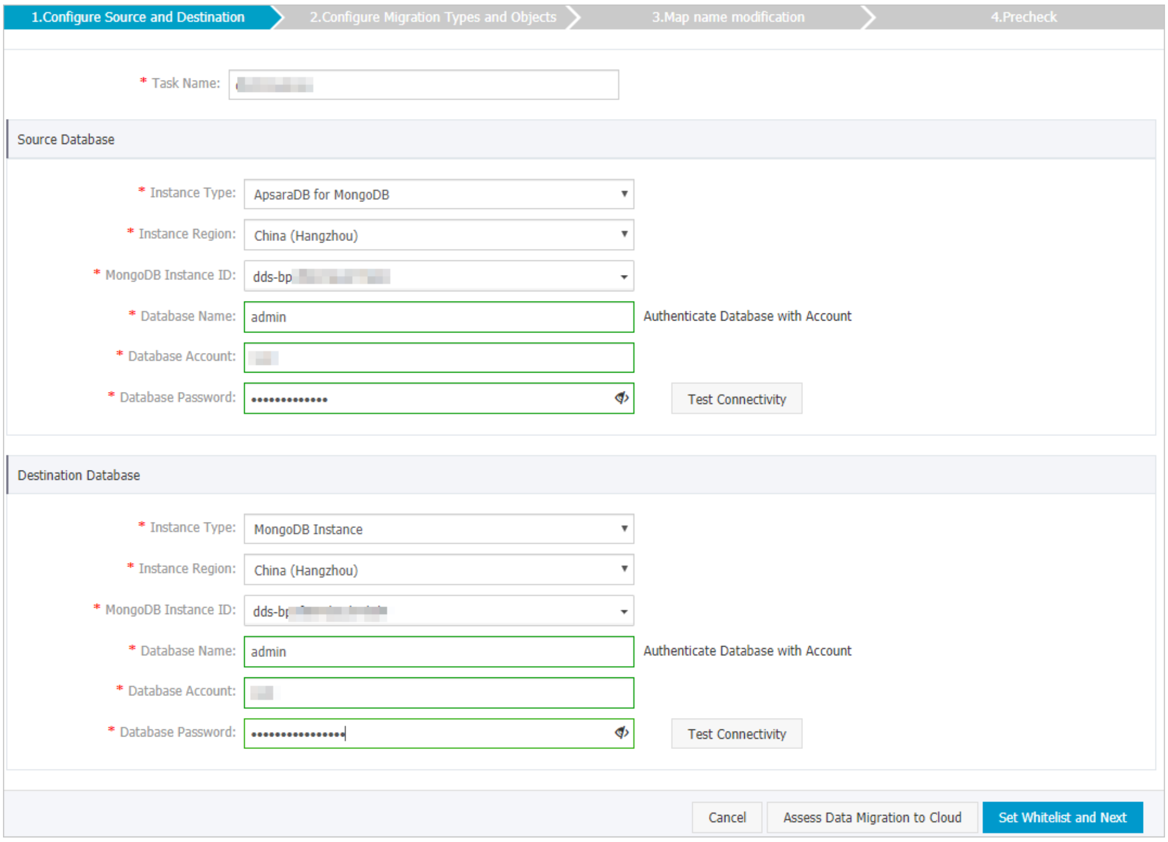
For more information about how to create and authorize a database account, see Manage MongoDB databases by using DMS.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination ApsaraDB for MongoDB instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the source and destination databases for the data migration task.



1.Configure Source and Destination | 2.Configure Migration Types and Objects | 3.Map name modification | 4.Precheck

* Task Name: [redacted]

Source Database

* Instance Type: ApsaraDB for MongoDB

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin — Authenticate Database with Account

* Database Account: [redacted]

* Database Password: ••••••••••••• — Test Connectivity

Destination Database

* Instance Type: MongoDB Instance

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-b[redacted]

* Database Name: admin — Authenticate Database with Account

* Database Account: [redacted]

* Database Password: •••••••••••••••• — Test Connectivity

Cancel | Assess Data Migration to Cloud | Set Whitelist and Next

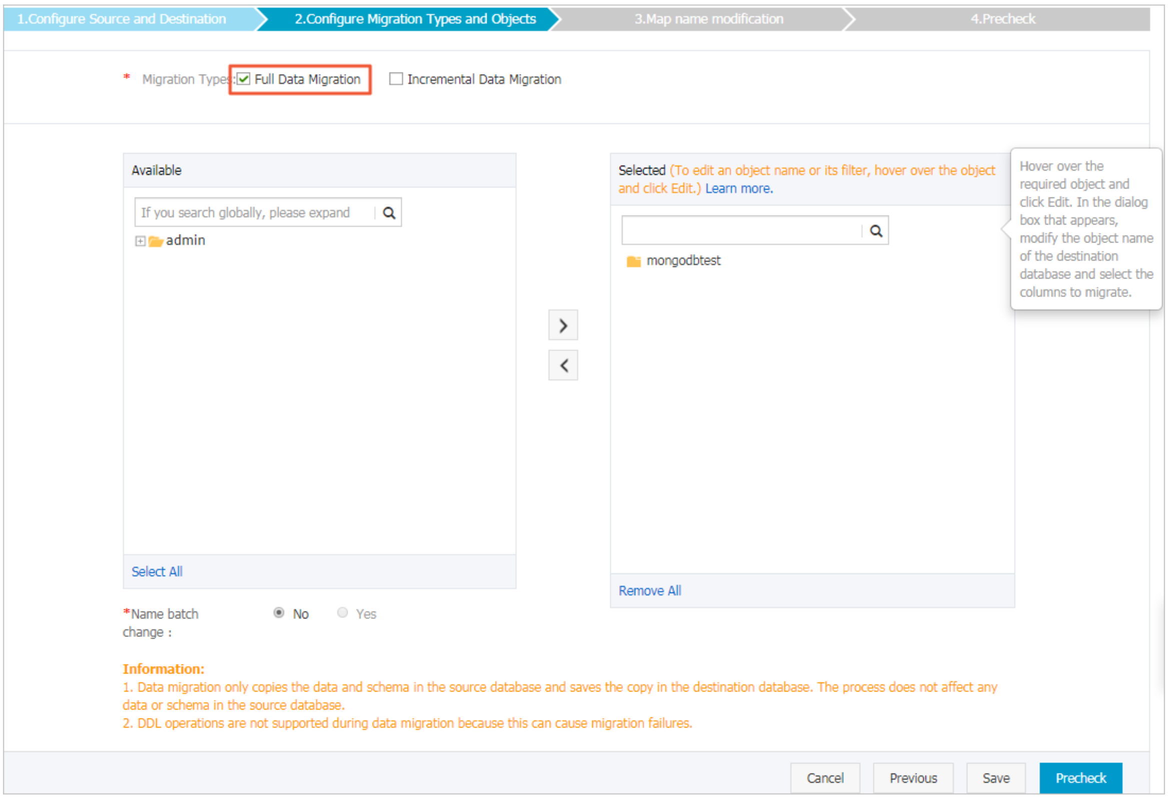
Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **ApsaraDB for MongoDB**.
	Instance Region	Select the region where the source ApsaraDB for MongoDB instance is deployed.
	MongoDB Instance ID	Select the ID of the source ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database to which the database account belongs. **Note** If you want to use the root account, specify admin for the Database Name parameter.
	Database Account	Enter the database account of the source ApsaraDB for MongoDB instance. For more information about the permissions that are required for the account, see Required database account permissions.

Section	Parameter	Description
	Database Password	Enter the password of the source database account. **Note** After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
Destination Database	Instance Type	The type of the instance. In this example, select **MongoDB Instance**.
	Instance Region	The region where the ApsaraDB for MongoDB instance is deployed.
	MongoDB Instance ID	Select the ID of the ApsaraDB for MongoDB database.
	Database Name	Enter the name of the authentication database to which the database account belongs. **Note** If you want to use the root account, specify admin for the Database Name parameter.
	Database Account	Enter the database account of the destination ApsaraDB for MongoDB instance. For more information about the permissions that are required for the account, see Required database account permissions.
	Database Password	Enter the password of the destination database account. **Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

6. In the lower-right corner, click **Set Whitelist and Next**.

> **Note** DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination ApsaraDB for MongoDB instances. This ensures that DTS servers can connect to the source and destination ApsaraDB for MongoDB instances. After data migration is complete, you can remove the CIDR blocks of DTS servers from the whitelists. For more information, see Configure a whitelist for a sharded cluster instance.

7. Select the migration types and the objects to be migrated.


1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck
Available
If you search globally, please expand
admin
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
mongodbtest
Remove All
Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.
>
<
Cancel
Previous
Save
Precheck

* Migration Types: ☑ Full Data Migration ☐ Incremental Data Migration

*Name batch change : ● No ○ Yes

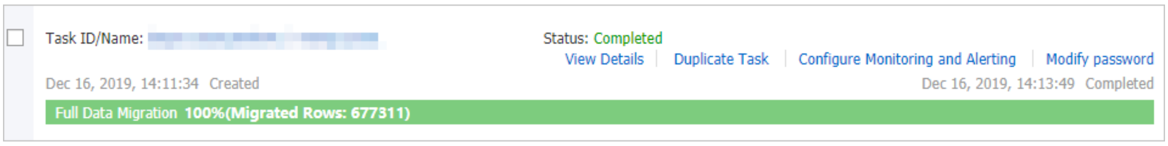
Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description
Migration Types	Select **Full Data Migration**. Note If the data source is a standalone instance, you can select only **Full Data Migration**. To ensure data consistency, do not write data to the source instance during full data migration.
Objects	- Select objects from the **Available** section and click the > icon to move the objects to the **Selected** section. Note Data in the admin and local databases cannot be migrated. - You can select databases, collections, or functions to migrate. - By default, the name of an object remains unchanged after the migration. You can change the names of the objects that are migrated to the destination database by using the object name mapping feature. For more information about how to use this feature, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

> Note
> - Before you can start the data migration task, a precheck is performed. You can start the data migration task only after the task passes the precheck.
> - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

9. After the task passes the precheck, click **Next**.
10. In the **Confirm Settings** dialog box, specify the **Channel Specification** and select **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.



□ Task ID/Name: [redacted]

Status: Completed

View Details | Duplicate Task | Configure Monitoring and Alerting | Modify password

Dec 16, 2019, 14:11:34 Created

Dec 16, 2019, 14:13:49 Completed

Full Data Migration 100%(Migrated Rows: 677311)

> Note Do not manually stop a task during full data migration. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task automatically stops.

12. Switch your workloads to the destination ApsaraDB for MongoDB instance.

References

- Overview of replica set instance connections
- Overview of sharded cluster instance connections
- Configure sharding to maximize the performance of shards

14.4.3. Migrate the data of an ApsaraDB for MongoDB instance across regions

This topic describes how to migrate the data of a standalone instance or a replica set instance across regions by using Data Transmission Service (DTS). DTS supports both full data migration and incremental data migration. You can use these two methods together to migrate the data of an ApsaraDB for MongoDB instance across regions without interruptions to your business.

Prerequisites

- The source instance is either a standalone instance or a replica set instance. If the source instance is a sharded cluster instance, we recommend that you use the built-in commands of MongoDB to migrate data. For more information, see Migrate a self-managed MongoDB database to ApsaraDB for MongoDB by using tools provided by MongoDB.

  > Note You cannot use DTS to incrementally migrate the data of a standalone instance. For more information, see Migration types.

- The destination instance is created in the destination region. For more information, see Create a standalone instance, 创建副本集实例, or 创建分片集群实例.

> Note The storage capacity of the destination instance must be greater than the occupied storage space of the source instance.

Context

You may need to migrate the data of an ApsaraDB for MongoDB instance across regions if:

- You want to restructure your business.
- You want to use the ApsaraDB for MongoDB instance to provide database services for applications deployed on an ECS instance, but the two instances are in different regions.

The following procedure illustrates how to migrate data from an ApsaraDB for MongoDB instance in China (Qingdao) to an instance in China (Hangzhou).

MongoDB instance
China (Qingdao)
MongoDB instance
China (Hangzhou)

> Note The procedure described in this topic only shows how to migrate the data of the source instance. If you no longer need the source instance after the migration is complete, you can release it.

Precautions

- We recommend that you migrate your data during off-peak hours to avoid business interruptions.
- To ensure data consistency, we recommend that you do not write data to the source instance while full data migration of a standalone instance is in progress.
- If the source and destination instances run different database versions or storage engines, ensure there are no compatibility issues between them before you start migration. For more information about the database versions and storage engines supported by ApsaraDB for MongoDB, see MongoDB versions and storage engines.
- To ensures better performance and stability of the instance, the system will upgrade the minor version to the latest version by default If the minor version of your instance expires or is not included in the maintenance list and the instance is upgraded, migrated, changed, Created from a backup, Created by point-in-time, or performed Restore data to a new ApsaraDB for MongoDB instance.

Billing

Migration type	Instance configuration	Internet traffic

Migration type	Instance configuration	Internet traffic
Full data migration	Free of charge.	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see DTS pricing.
Incremental data migration	Charged. For more information, see DTS pricing.	

Migration types

Migration type	Description
Full data migration	DTS migrates all historical data of the required objects from the source MongoDB database to the destination MongoDB database. **Note** The following types of objects are supported: database, collection, and index.
Incremental data migration	After full data migration is complete, DTS synchronizes incremental data from the source MongoDB database to the destination MongoDB database. **Note** • The create and delete operations that are performed on databases, collections, and indexes can be synchronized. • The create, delete, and update operations that are performed on documents can be synchronized.

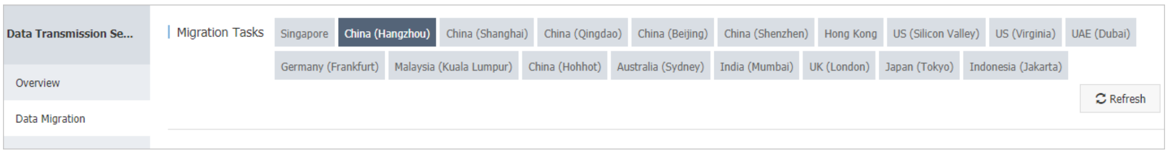
Required database account permissions

Data source	Full data migration	Incremental data migration
Source ApsaraDB for MongoDB instance	Read permissions on the source database	Read permissions on the source database, admin database, and local database
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination database	Read/write permissions on the destination database

Note For more information about how to create and authorize a database account, see Use DMS to manage MongoDB users.

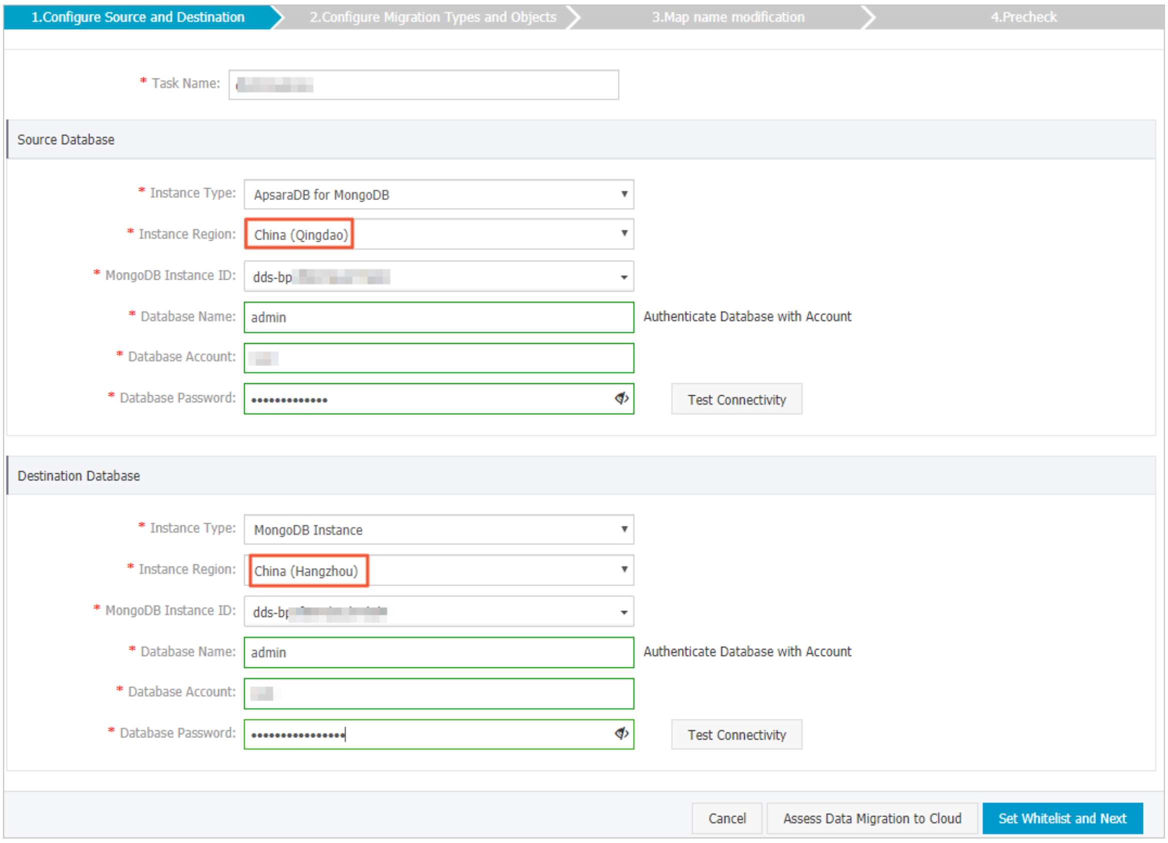
Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination ApsaraDB for MongoDB instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.

5. Configure both the source and destination databases.


1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name: [redacted]

Source Database

* Instance Type: ApsaraDB for MongoDB

* Instance Region: China (Qingdao)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin   Authenticate Database with Account

* Database Account: [redacted]

* Database Password: •••••••••••••

Destination Database

* Instance Type: MongoDB Instance

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin   Authenticate Database with Account

* Database Account: [redacted]

* Database Password: ••••••••••••••••

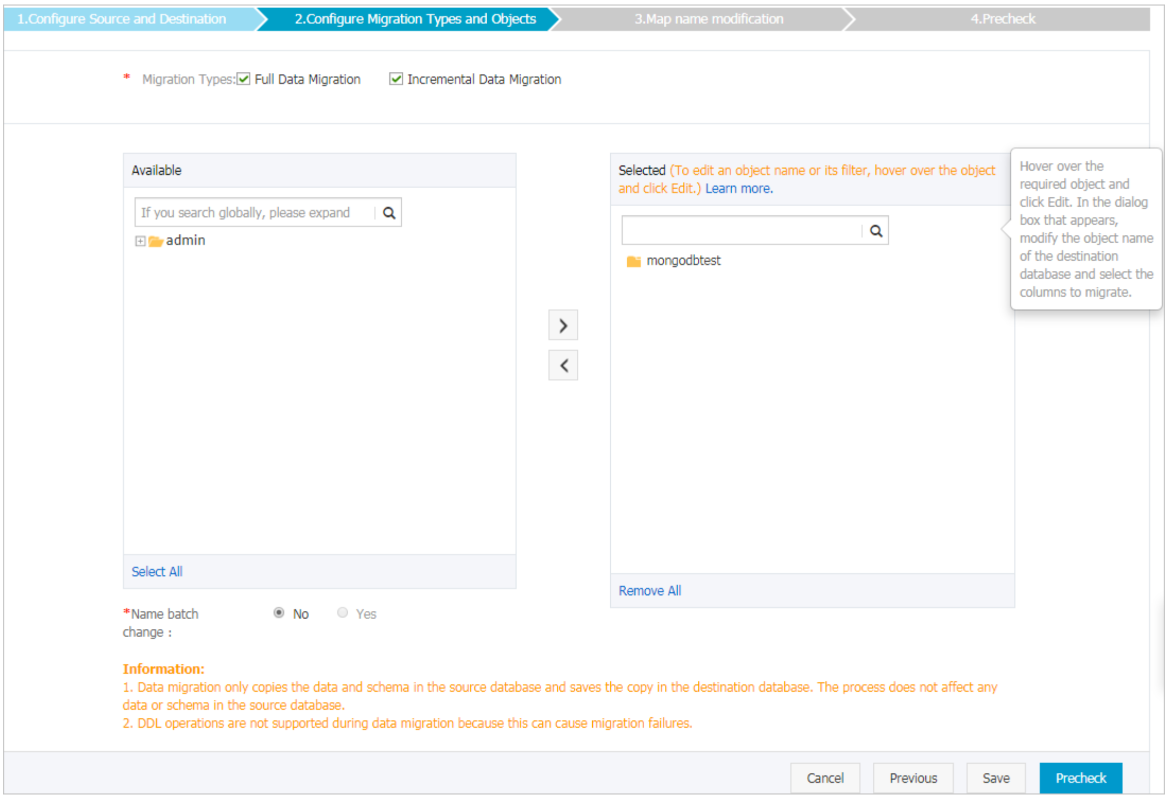
Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify your own task name that helps identify the task. Task names do not need to be unique.
Source	Instance Type	Select **ApsaraDB for MongoDB**.
	Instance Region	Select the region where the source ApsaraDB for MongoDB instance resides. For this example, select **China (Qingdao)**.
	MongoDB Instance ID	Select the ID of the source ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. It is the database where the database account is created. **Note** If the database account is root, enter admin.

Section	Parameter	Description
Source Database	Database Account	Enter the username of the database account you use to manage the source database. For more information about the account permission requirements, see Required database account permissions.
	Database Password	Enter the password of the database account. **Note** After you specify the source database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message is displayed. If the information is incorrect, the **Failed** message is displayed, and you must click **Check** next to the **Failed** message to modify the information as prompted.
Destination Database	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region where the destination ApsaraDB for MongoDB instance resides. For this example, select **China (Hangzhou)**.
	MongoDB Instance ID	Select the ID of the destination ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. It is the database where the database account is created. **Note** If the database account is root, enter admin.
	Database Account	Enter the username of the database account you use to manage the destination database. For more information about the account permission requirements, see Required database account permissions.
	Database Password	Enter the password of the database account. **Note** After you specify the destination database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message is displayed. If the information is incorrect, the **Failed** message is displayed, and you must click **Check** next to the **Failed** message to modify the information as prompted.

6. In the lower-right corner, click **Set Whitelist and Next**.

> **Note** DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination ApsaraDB for MongoDB instances. This ensures that DTS servers can connect to the source and destination ApsaraDB for MongoDB instances. After data migration is complete, you can remove the CIDR blocks of DTS servers from the whitelists. For more information, see Configure a whitelist for a sharded cluster instance.

7. Configure migration types and migration objects.


1.Configure Source and Destination
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

Available
If you search globally, please expand
admin
Select All
Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
mongodbtest
Remove All

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description

Parameter	Description
Migration Types	○ If you want to migrate all data, select **Full Data Migration**. ○ If you want to migrate data without interruptions to your business, select both **Full Data Migration** and **Incremental Data Migration**. ? Note ○ You cannot use DTS to incrementally migrate the data of a standalone instance. ○ If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
Available	○ In the **Available** section, select the objects you want to migrate and then click the > icon to move them to the **Selected** section. ? Note Data in the admin database cannot be migrated even if this database is selected. ○ A migration object can be a database, collection, or function. ○ By default, the name of an object remains unchanged after migration. If you want a different object name after migration, use the object name mapping feature provided by DTS. For more information, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

> ? Note
>
> - Before you can start the data migration task, a precheck is performed. You can start the data migration task only after the task passes the precheck.
> - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

9. After the task passes the precheck, click **Next**.
10. In the **Confirm Settings** dialog box, specify the **Channel Specification** and select **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the migration task.

- Full data migration

  We recommend that you do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the migration task automatically stops.

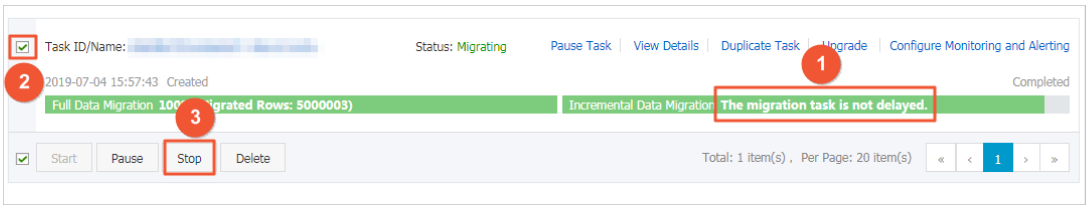
- Incremental data migration

  An incremental data migration task does not automatically stop. You must manually stop the migration task.

  > **Note** Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the destination ApsaraDB for MongoDB instance.

  a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

  b. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.



☑ Task ID/Name: [redacted] Status: Migrating

2019-07-04 15:57:43 Created

Full Data Migration 100% (Migrated Rows: 5000003)

Incremental Data Migration The migration task is not delayed.

Completed

Total: 1 item(s) , Per Page: 20 item(s)

What to do next

Determine whether to release the source instance.

- If the source instance uses pay-as-you-go billing, release it. For more information, see Release an ApsaraDB for MongoDB instance.
- If the source instance uses subscription billing, you cannot release it.

14.4.4. Migrate data between ApsaraDB for MongoDB instances created by different Alibaba Cloud accounts

This topic describes how to migrate data between ApsaraDB for MongoDB instances created by different Alibaba Cloud accounts by using Data Transmission Service (DTS). DTS supports both full data migration and incremental data migration. You can use these two methods together to migrate data between ApsaraDB for MongoDB instances without interruptions to your business.

Prerequisites

- The source instance is either a standalone instance or a replica set instance. If the source instance is a sharded cluster instance, we recommend that you use the built-in commands of MongoDB to migrate data. For more information, see Migrate a self-managed MongoDB database to ApsaraDB for MongoDB by using tools provided by MongoDB.

> Note You cannot use DTS to incrementally migrate the data of a standalone instance. For more information, see Migration types.

- The destination instance is created in the destination region. For more information, see Create a standalone instance, 创建副本集实例, or 创建分片集群实例.

> Note The storage capacity of the destination instance must be greater than the occupied storage space of the source instance.

Precautions

- DTS uses resources of the source and destination instances during full data migration. This may increase the load of the database server. If the data volume is large or the specification is low, the database server may become unavailable. We recommend that you migrate your data during off-peak hours to avoid business interruptions.
- To ensure data consistency, we recommend that you do not write data to the source instance when full data migration of a standalone instance is in progress.
- If the source and destination instances run different database versions or storage engines, make sure that the instances do not have compatibility issues between them before you start the migration. For more information about the database versions and storage engines supported by ApsaraDB for MongoDB, see MongoDB versions and storage engines.

Billing

Migration type	Instance configuration	Internet traffic
Full data migration	Free of charge.	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see DTS pricing.
Incremental data migration	Charged. For more information, see DTS pricing.	

Migration types

Migration type	Description
Full data migration	DTS migrates all historical data of the required objects from the source MongoDB database to the destination MongoDB database. Note The following types of objects are supported: database, collection, and index.

Migration type	Description
Incremental data migration	After full data migration is complete, DTS synchronizes incremental data from the source MongoDB database to the destination MongoDB database. Note • The create and delete operations that are performed on databases, collections, and indexes can be synchronized. • The create, delete, and update operations that are performed on documents can be synchronized.

Required database account permissions

Data source	Full data migration	Incremental data migration
Source ApsaraDB for MongoDB instance	Read permissions on the source database	Read permissions on the source database, admin database, and local database
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination database	Read/write permissions on the destination database

Note For more information about how to create and authorize a database account, see Use DMS to manage MongoDB users.

Preparation

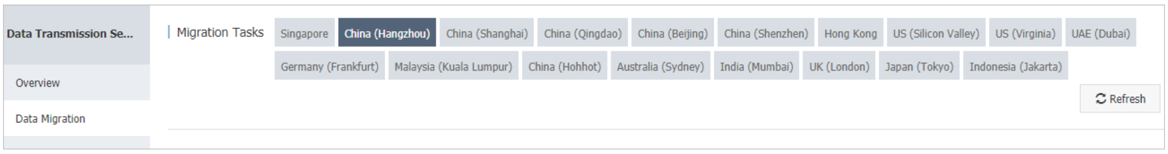
1. Log on to the ApsaraDB for MongoDB console with the Alibaba Cloud account to which the source instance belongs.
2. Apply for a public endpoint for the source instance. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.
3. Add the Classless Inter-Domain Routing (CIDR) blocks of DTS servers to a whitelist of the source instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

   Note You can determine the CIDR blocks you need to add based on the region where the destination instance resides. For more information, see Add the CIDR blocks of DTS servers to the IP whitelist of on-premises databases. For example, if the source instance is in China (Hangzhou) and the destination instance is in China (Shenzhen), add the CIDR blocks of the DTS servers in China (Shenzhen) to a whitelist of the source instance.

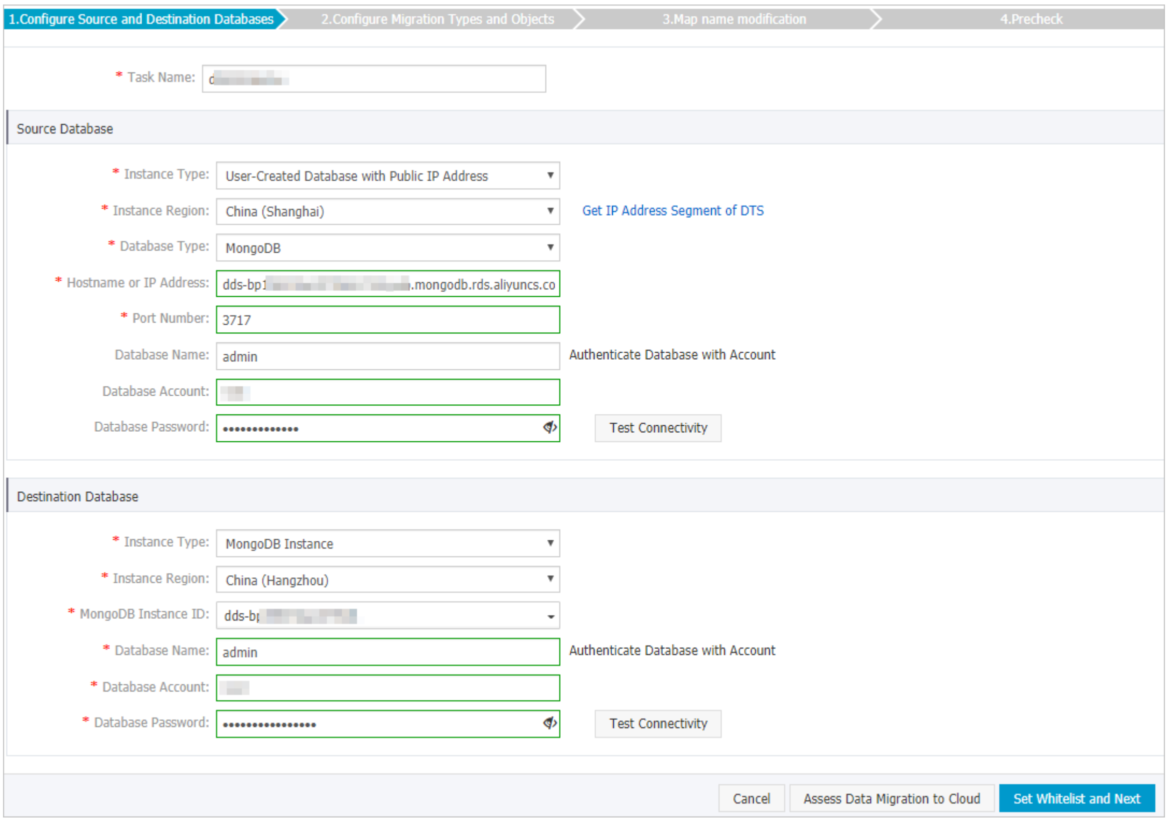
Procedure

1. Log on to the DTS console with the Alibaba Cloud account to which the destination ApsaraDB for MongoDB instance belongs.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination ApsaraDB for

MongoDB instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure both the source and destination databases.


1.Configure Source and Destination Databases
2.Configure Migration Types and Objects
3.Map name modification
4.Precheck

* Task Name: d[redacted]

Source Database

* Instance Type: User-Created Database with Public IP Address

* Instance Region: China (Shanghai)

Get IP Address Segment of DTS

* Database Type: MongoDB

* Hostname or IP Address: dds-bp1[redacted].mongodb.rds.aliyuncs.co

* Port Number: 3717

Database Name: admin

Authenticate Database with Account

Database Account: [redacted]

Database Password: •••••••••••••

Test Connectivity

Destination Database

* Instance Type: MongoDB Instance

* Instance Region: China (Hangzhou)

* MongoDB Instance ID: dds-bp[redacted]

* Database Name: admin

Authenticate Database with Account

* Database Account: [redacted]

* Database Password: ••••••••••••••••

Test Connectivity

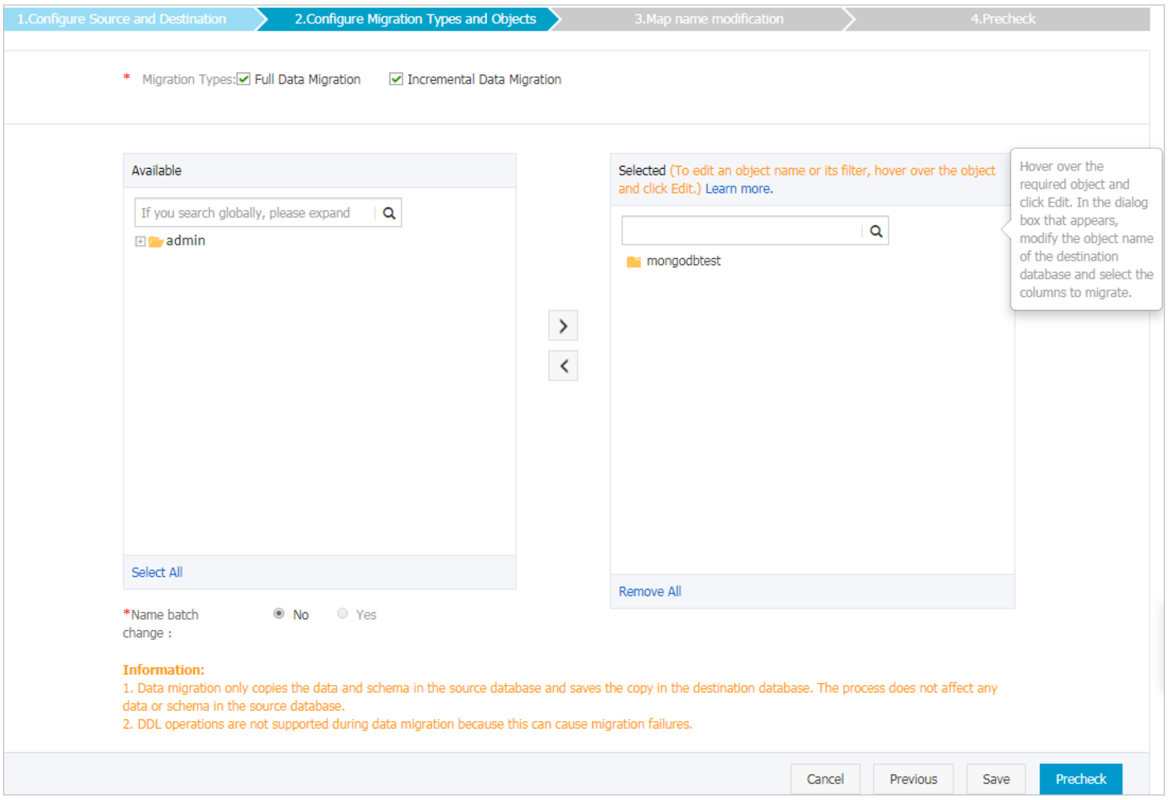
Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify your own task name that helps identify the task. Task names do not need to be unique.
	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If you set the instance type to **User-Created Database with Public IP Address**, the system automatically specifies **Instance Region**.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the domain name obtained from the public endpoint of the source instance. For example, enter dds-1udxxxxxxx-pub.mongodb.rds.aliyuncs.com.
	Port Number	Enter **3717**, which is the service port of the source instance.

Section	Parameter	Description
Source Database	Database Name	Enter the name of the authentication database. It is the database where the database account is created. Note If the database account is root, enter admin.
	Database Account	Enter the username of the database account you use to manage the source database. For more information about the account permission requirements, see Required database account permissions.
	Database Password	Enter the password of the database account. Note After you specify the source database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message is displayed. If the information is incorrect, the **Failed** message is displayed, and you must click **Check** next to the **Failed** message to modify the information as prompted.
Destination Database	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region where the destination ApsaraDB for MongoDB instance resides.
	MongoDB Instance ID	Select the ID of the destination ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. It is the database where the database account is created. Note If the database account is root, enter admin.
	Database Account	Enter the username of the database account you use to manage the destination database. For more information about the account permission requirements, see Required database account permissions.
	Database Password	Enter the password of the database account. Note After you specify the destination database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the information is correct, the **Passed** message is displayed. If the information is incorrect, the **Failed** message is displayed, and you must click **Check** next to the **Failed** message to modify the information as prompted.

6. In the lower-right corner, click **Set Whitelist and Next**.

> **Note** The IP addresses of DTS servers are automatically added to a whitelist of the destination instance. This ensures that the DTS servers can connect to the destination instance. After the migration is complete, you can remove the IP addresses from the whitelist if you no longer need them. For more information, see Configure a whitelist.

7. Configure migration types and migration objects.



* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:

1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

Parameter	Description

Parameter	Description
Migration Types	○ If you want to migrate all data, select **Full Data Migration**. ○ If you want to migrate data without interruptions to your business, select both **Full Data Migration** and **Incremental Data Migration**. Note ○ DTS does not support **incremental data migration** for standalone instances. ○ If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
Available	○ In the **Available** section, select the objects you want to migrate and then click the › icon to move them to the **Selected** section. Note Data in the admin and local databases cannot be migrated. ○ A migration object can be a database, collection, or function. ○ By default, the name of an object remains unchanged after migration. If you want a different object name after migration, use the object name mapping feature provided by DTS. For more information, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

   Note

   - Before you can start the data migration task, a precheck is performed. You can start the data migration task only after the task passes the precheck.
   - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

9. After the task passes the precheck, click **Next**.
10. In the **Confirm Settings** dialog box, specify the **Channel Specification** and select **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the migration task.

   - Full data migration

We recommend that you do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the migration task automatically stops.

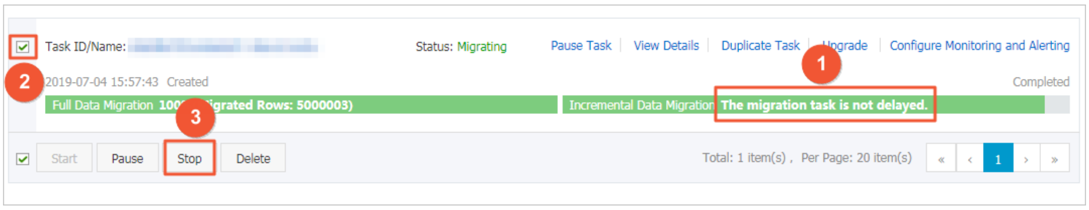
- Incremental data migration

  An incremental data migration task does not automatically stop. You must manually stop the migration task.

  > **Note** Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the destination ApsaraDB for MongoDB instance.

  a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

  b. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.


☑ Task ID/Name: [redacted]
Status: Migrating
Pause Task
View Details
Duplicate Task
Upgrade
Configure Monitoring and Alerting
1
2
2019-07-04 15:57:43 Created
Completed
Full Data Migration 100% (Migrated Rows: 5000003)
3
Incremental Data Migration
The migration task is not delayed.
☑
Start
Pause
Stop
Delete
Total: 1 item(s) , Per Page: 20 item(s)
«
‹
1
›
»

Subsequent operations

Determine whether to release the source instance.

- If the source instance uses pay-as-you-go billing, release it. For more information, see Release an ApsaraDB for MongoDB instance.
- If the source instance uses subscription billing, you cannot release it.

References

If you migrate data to a sharded cluster instance, you can configure data sharding as needed. For more information, see Configure sharding to maximize the performance of shards.

14.5. Migrate data from a third-party cloud service provider to Alibaba Cloud

14.5.1. Migrate data from Amazon DynamoDB to ApsaraDB for MongoDB by using mongoimport

This topic describes how to migrate data from Amazon DynamoDB to ApsaraDB for MongoDB by using mongoimport, which is built in MongoDB for data restoration. You can install a MongoDB database on an on-premises server or in an ECS instance as an intermediary, and then use mongoimport to migrate data from an Amazon DynamoDB instance to an ApsaraDB for MongoDB instance.

ⓘ **Note** An on-premises server is used in the following example.

Prerequisites

An ApsaraDB for MongoDB instance is created. For more information, see 创建副本集实例 or 创建分片集群实例.

Precautions

- This is full data migration. Incremental data migration is not supported. To ensure data consistency, we recommend that you do not write data to the source database before you migrate data.
- Run the mongoimport command on the server. Do not run this command in the mongo shell.

Term mapping

Term in Amazon DynamoDB	Term in ApsaraDB for MongoDB
Table	Collection
Item	Document
Attribute	Field

Preparation

Install a MongoDB database on the on-premises server. Make sure that the version of the database is the same as the database version of the ApsaraDB for MongoDB instance. This server plays an intermediary role during data migration. In this example, MongoDB is installed on Ubuntu.

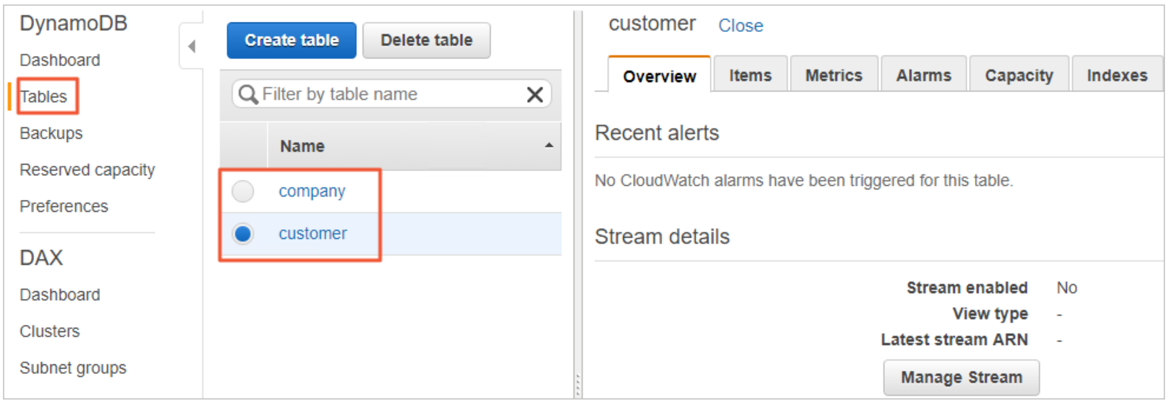
1. Install the MongoDB database to your server. For more information, visit Install MongoDB Community Edition.

   ⓘ **Note** This server plays an intermediary role in both data backup and restoration. The server is no longer required after the migration is complete.

2. Add the public IP address of the server to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

Procedure

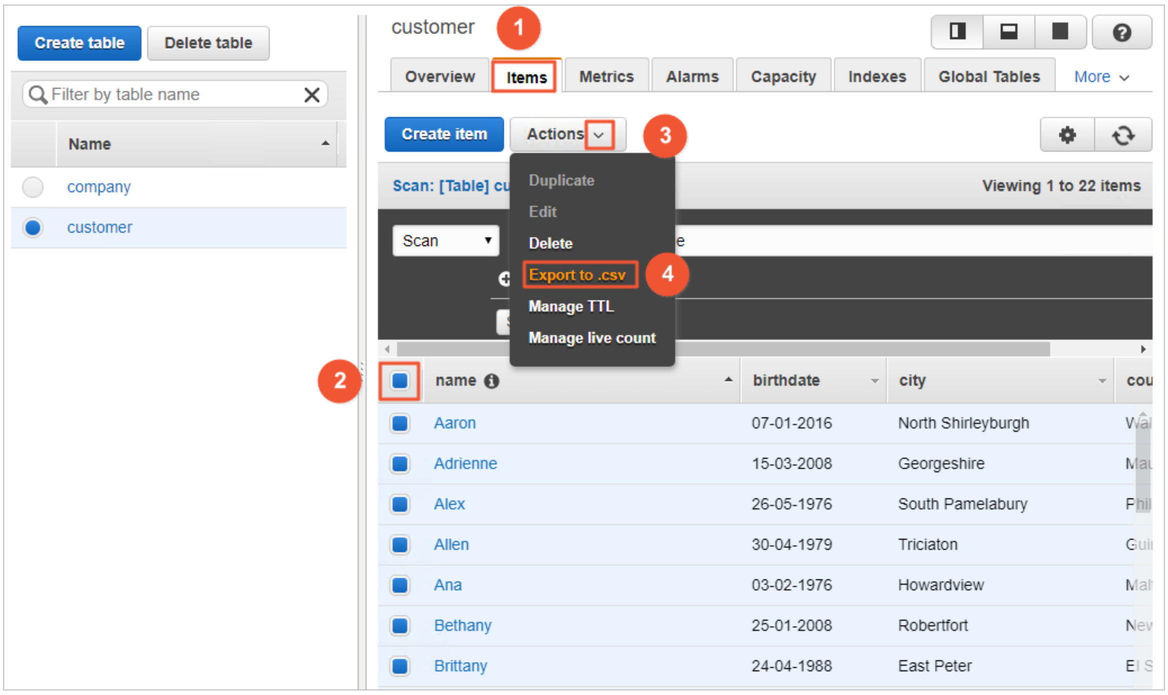
1. Log on to the Amazon DynamoDB console.
2. In the left-side navigation pane, click **Tables**.
3. Select the table you want to migrate. For this example, select **customer**.



No CloudWatch alarms have been triggered for this table.

Stream enabled No
View type -
Latest stream ARN -

4. Select the data you want to migrate and export it to a .csv file.



name	birthdate	city	cou
Aaron	07-01-2016	North Shirleyburgh	Wal
Adrienne	15-03-2008	Georgeshire	Mau
Alex	26-05-1976	South Pamelabury	Phil
Allen	30-04-1979	Triciaton	Gui
Ana	03-02-1976	Howardview	Mal
Bethany	25-01-2008	Robertfort	Nev
Brittany	24-04-1988	East Peter	El S

i. Click the **Items** tab.

ii. Select the items you want to migrate.

iii. Choose **Actions** > **Export to .csv**.

5. Copy the exported .csv file to the server described in Preparation.

6. Log on to the ApsaraDB for MongoDB console to obtain the public endpoint of the ApsaraDB for MongoDB instance.

   - If you migrate data to a replica set instance of ApsaraDB for MongoDB, obtain the public endpoint of the primary node. For more information, see Overview of replica set instance connections.
   - If you migrate data to a sharded cluster instance of ApsaraDB for MongoDB, obtain the public endpoint of a mongos. For more information, see Overview of sharded cluster instance connections.

> **Note** You must manually apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.

7. On the server, run the following command to import the backup files to an ApsaraDB for MongoDB database:

   ```
   mongoimport --host <mongodb_host:port> --authenticationDatabase admin -u <username> --db <database> --collection <collection> --file <filename>  --type csv --headerline
   ```

   Note

   - <mongodb_host:port>: the public endpoint of the primary node in the replica set instance of ApsaraDB for MongoDB, or the public endpoint of a mongos in the sharded cluster instance of ApsaraDB for MongoDB.
   - <username>: the username of the destination database in the ApsaraDB for MongoDB instance. The initial username is **root**. This user must have read and write permissions on the destination database.
   - <database>: the name of the authentication database. It is the database where the database user is created. If the username is root, enter admin.
   - <collection>: the name of the collection to which you want to import data.
   - <filename>: the path and name of the .csv file.

   The following command is used to import data from the customer.csv file to the customer collection in the mongodbtest database.

   ```
   mongoimport --host dds-bpxxxxxxxx-pub.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u root --db mongodbtest --collection customer --file ~/download/customer.csv  --type csv --headerline
   ```

8. When `Enter password:` is displayed, enter the password of the ApsaraDB for MongoDB database user and press Enter.

   Note If you want to migrate data of other tables, repeat Steps 3 to 8.

After data import is complete, the data of the source Amazon DynamoDB instance is migrated to the destination ApsaraDB for MongoDB instance.

What to do next

To improve the performance of data operations, we recommend that you create indexes after you import data. For more information, visit db.collection.createIndex.

14.5.2. Migrate data from MongoDB Atlas to ApsaraDB for MongoDB by using mongodump and mongorestore

This topic describes how to migrate data from MongoDB Atlas to ApsaraDB for MongoDB by using mongodump and mongorestore, which are built in open source MongoDB for backup and restoration. You can directly install mongodump and mongorestore (or create a MongoDB Atlas instance) on an on-premises server or an ECS instance to migrate data from the MongoDB Atlas instance to an ApsaraDB for MongoDB instance.

ⓘ **Note** In the following procedure, an on-premises server is used as an example.

Precautions

- mongodump and mongorestore are installed, and their version is consistent with the database version of the MongoDB Atlas instance. For more information about the installation procedure, see Install MongoDB.
- If the source and destination instances run different database versions or storage engines, ensure there are no compatibility issues between them before you start migration. For more information about the database versions and storage engines supported by ApsaraDB for MongoDB, see MongoDB versions and storage engines.
- This process is a full data migration. To ensure data consistency, we recommend that you do not write data to the source databases before you migrate data.
- If you run the `mongodump` command, the historical backup files in the dump folder are overwritten. If you have used the mongodump command to back up a MongoDB Atlas database, move the backup files in the *dump* folder to another directory and make sure that the *dump* folder is empty.
- Run the mongodump and mongorestore commands on the on-premises server. Do not run these commands in the mongo shell.

Required database account permissions

Data source	Account permission
Source MongoDB Atlas instance	Read permissions on the source databases
Destination ApsaraDB for MongoDB instance	Read/write permissions on the destination databases

Preparations

In the ApsaraDB for MongoDB console:

1. Create an ApsaraDB for MongoDB instance. For more information, see Create a replica set instance or Create a sharded cluster instance.

   ⓘ **Note**

   - The storage capacity of the ApsaraDB for MongoDB instance must be greater than that of the MongoDB Atlas instance.
   - If you migrate data to a sharded cluster instance of ApsaraDB for MongoDB, we recommend that you configure sharding to store data. For more information, see Configure sharding to maximize the performance of shards.

2. Set the password of the root user for the ApsaraDB for MongoDB instance. For more information,

see Reset the password for an ApsaraDB for MongoDB instance.

> Note If you have set the password when you create the instance, skip this step.

On the on-premises server:

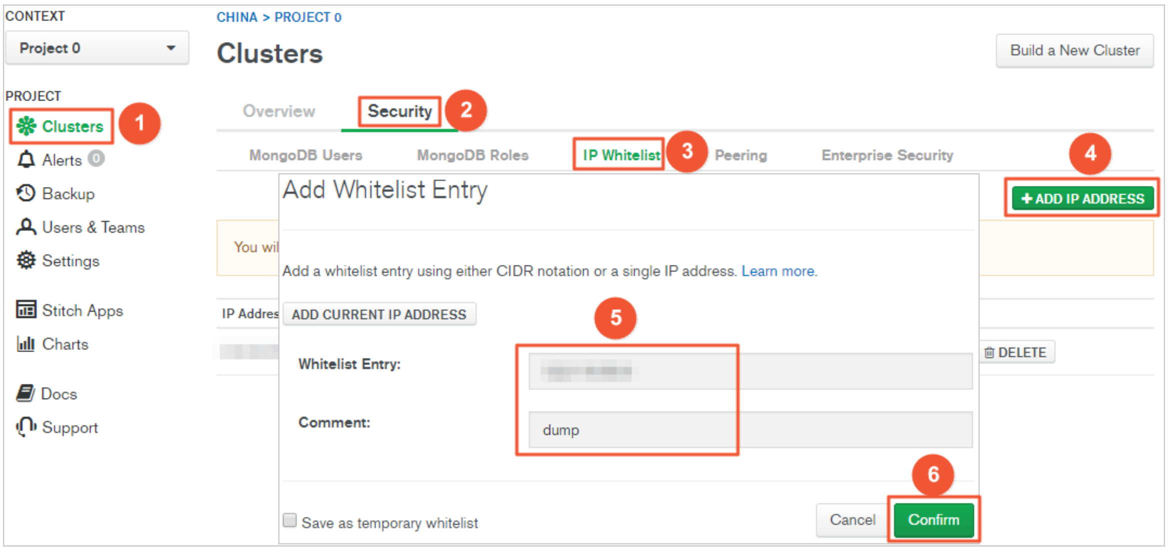
1. Install MongoDB databases. For more information, see Install MongoDB.

   > Note
   >
   > - This server functions as an intermediary platform for data backup and restoration. It is no longer required after the migration is complete.
   > - The available storage space of the partition where the backup directory is located must be greater than the occupied storage space of the source MongoDB Atlas databases.

2. Add the public IP address of the on-premises server to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

Procedure

1. Log on to the MongoDB Atlas console.
2. Add the public IP address of the on-premises server to a whitelist of the MongoDB Atlas instance.



Add Whitelist Entry

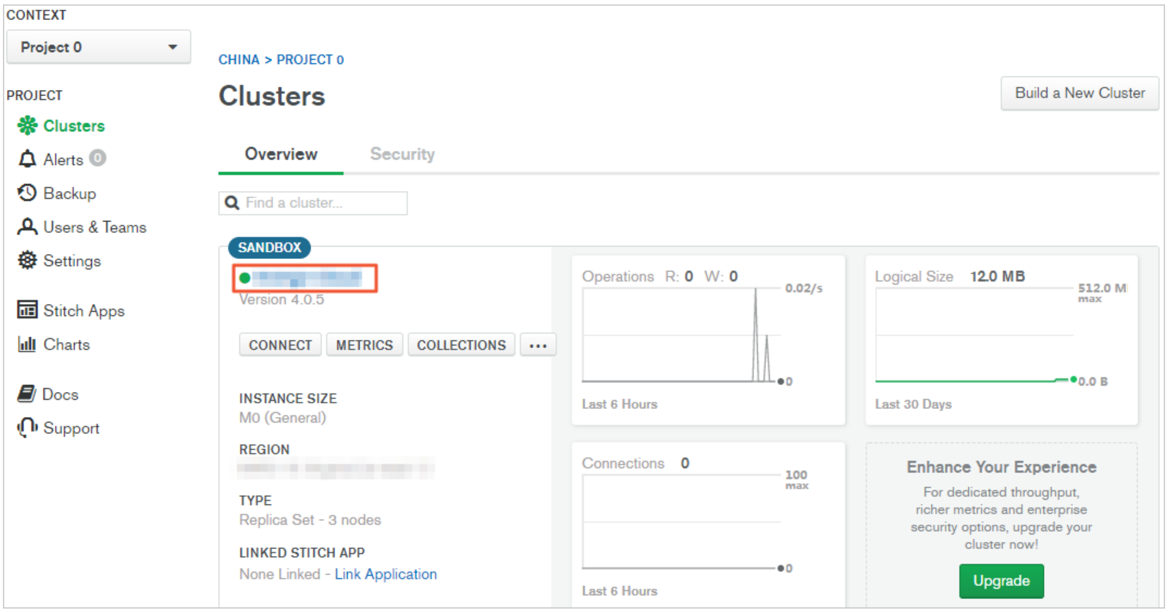
Add a whitelist entry using either CIDR notation or a single IP address. Learn more.

Whitelist Entry: [redacted]

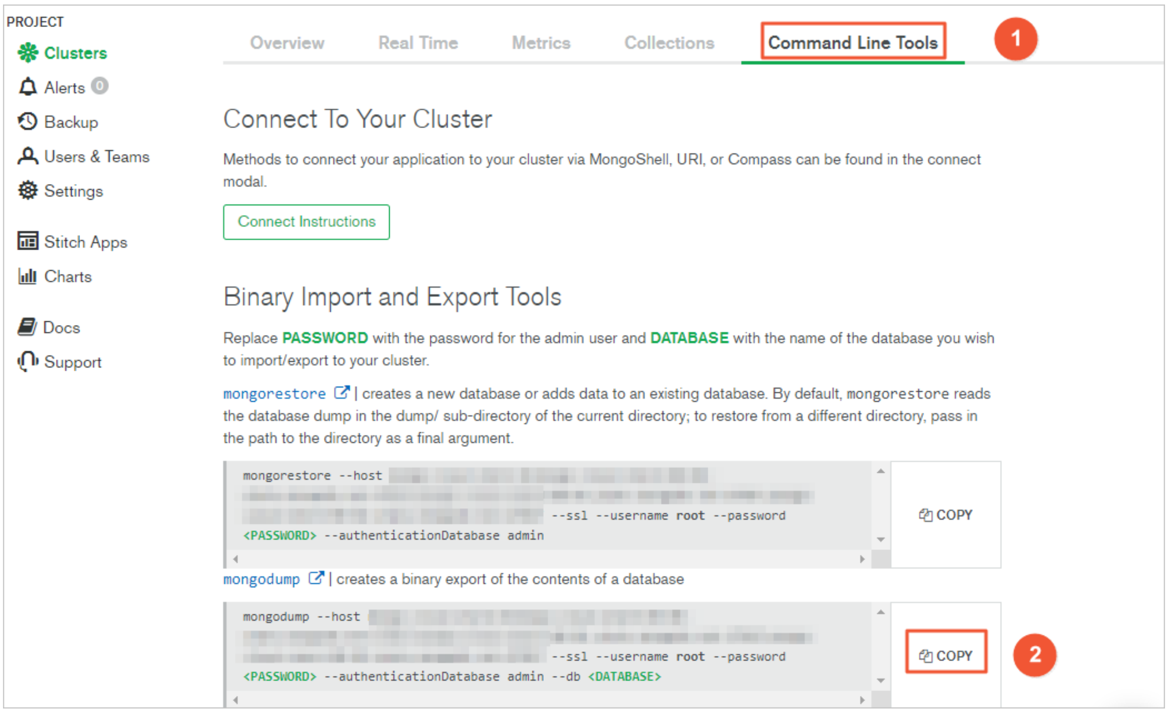
Comment: dump

□ Save as temporary whitelist

3. On the **Clusters** page, find the target cluster and click its name.



4. On the **Command Line Tools** tab, click **COPY** next to the mongodump command to copy this command with the connection information of the source MongoDB Atlas databases.



Connect To Your Cluster

Methods to connect your application to your cluster via MongoShell, URI, or Compass can be found in the connect modal.

Binary Import and Export Tools

Replace **PASSWORD** with the password for the admin user and **DATABASE** with the name of the database you wish to import/export to your cluster.

mongorestore | creates a new database or adds data to an existing database. By default, mongorestore reads the database dump in the dump/ sub-directory of the current directory; to restore from a different directory, pass in the path to the directory as a final argument.

```
mongorestore --host [redacted]
 --ssl --username root --password
<PASSWORD> --authenticationDatabase admin
```

mongodump | creates a binary export of the contents of a database

```
mongodump --host [redacted]
 --ssl --username root --password
<PASSWORD> --authenticationDatabase admin --db <DATABASE>
```

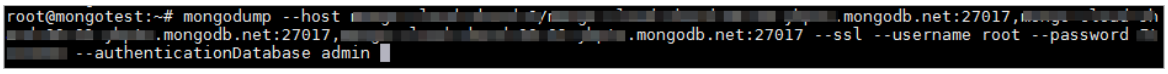
5. On the on-premises server, back up the source MongoDB Atlas databases.

   i. On the on-premises server, paste the mongodump command that contains the connection information of the source MongoDB Atlas databases.

   ii. Replace <PASSWORD> with the password of the root user and replace <DATABASE> with the names of the source MongoDB Atlas databases.

   iii. Run this command and wait until data backup is complete.

   The following figure shows an example.



```
root@mongotest:~# mongodump --host [redacted].mongodb.net:27017,[redacted]
[redacted].mongodb.net:27017,[redacted].mongodb.net:27017 --ssl --username root --password [redacted]
[redacted] --authenticationDatabase admin
```

6. Log on to the ApsaraDB for MongoDB console to obtain the public endpoint of the ApsaraDB for MongoDB instance.
   - If you migrate data to a replica set instance of ApsaraDB for MongoDB, obtain the public endpoint of the primary node. For more information, see Overview of replica set instance connections.
   - If you migrate data to a sharded cluster instance of ApsaraDB for MongoDB, obtain the public endpoint of a mongos. For more information, see Overview of sharded cluster instance connections.

   > **Note** You must manually apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance.

7. On the on-premises server, run the following command to import the backup files to the ApsaraDB for MongoDB database:

   ```
   mongorestore --host <mongodb_host>:3717 --authenticationDatabase admin -u <username> -d <database> <database_backupfile_directory>
   ```

   > **Note**
   > - <mongodb_host>: the public endpoint of the primary node in the replica set instance of ApsaraDB for MongoDB, or the public endpoint of a mongos in the sharded cluster instance of ApsaraDB for MongoDB.
   > - <username>: the username of the destination database in the ApsaraDB for MongoDB instance.
   > - <database>: the source database that you want to restore. If the backup files contain data from more than one database, repeat this step to restore all the databases.
   > - <database_backupfile_directory>: the directory of the backup files.

   Examples:

   Restore the mongodbtest database.

   ```
   mongorestore --host dds-bp**********-pub.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u root -d mongodbtest /dump/mongodbtest
   ```

   Restore the test123 database.

   ```
   mongorestore --host dds-bp**********-pub.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u root -d test123 /dump/test123
   ```

8. When `Enter password:` is displayed, enter the password of the ApsaraDB for MongoDB database user and press Enter.

After data restoration is complete, the data of the source MongoDB Atlas instance is migrated to the destination ApsaraDB for MongoDB instance.

14.5.3. Migrate data from a MongoDB Atlas database to Alibaba Cloud

This topic describes how to migrate incremental data from a MongoDB Atlas database to Alibaba Cloud by using Data Transmission Service (DTS). DTS supports full data migration and incremental data migration. When you migrate data from a MongoDB Atlas database, you can select both migration types to ensure service continuity.

Prerequisites

The available storage space of the destination ApsaraDB for MongoDB instance is larger than the total size of the data in the MongoDB Atlas database.

Precautions

- DTS uses resources of the source and destination instances during full data migration. This may increase the loads of the database servers. If you migrate a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.
- You cannot migrate data from the admin or local database.
- The config database is an internal database. We recommend that you do not migrate this database.
- If the source MongoDB Atlas database and the destination ApsaraDB for MongoDB instance have different versions or storage engines, make sure that the versions or storage engines are compatible. For more information, see MongoDB versions and storage engines.

Billing

Migration type	Instance configuration fee	Internet traffic fee
Full data migration	Free of charge	Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see Pricing.
Incremental data migration	Charged. For more information, see Pricing.	

Migration types

Migration type	Description
Full data migration	DTS migrates all historical data of the required objects from the source MongoDB database to the destination MongoDB database. Note The following types of objects are supported: database, collection, and index.

Migration type	Description
Incremental data migration	After full data migration is complete, DTS synchronizes incremental data from the source MongoDB database to the destination MongoDB database. Note • The create and delete operations that are performed on databases, collections, and indexes can be synchronized. • The create, delete, and update operations that are performed on documents can be synchronized.

Permissions required for database accounts

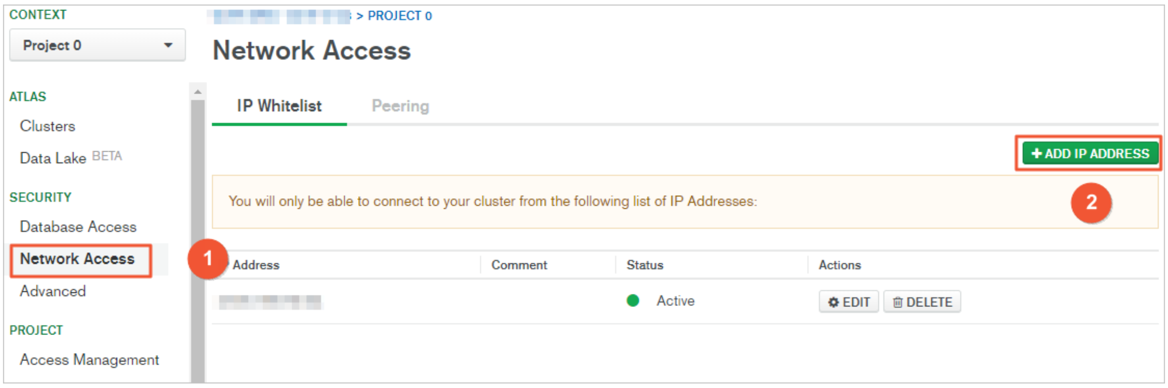
Database	Full data migration	Incremental data migration
MongoDB Atlas database	The read permission on the source database and the permission to perform the listDatabases operation	• The read permission on the source database, admin database, and local database • The permission to perform the listDatabases operation
ApsaraDB for MongoDB instance	The read and write permissions on the destination database	The read and write permissions on the destination database.

For more information about how to create and authorize a database account, see the following topics:

- MongoDB Atlas database: Create User in MongoDB
- ApsaraDB for MongoDB instance: Manage MongoDB users through DMS

Before you begin

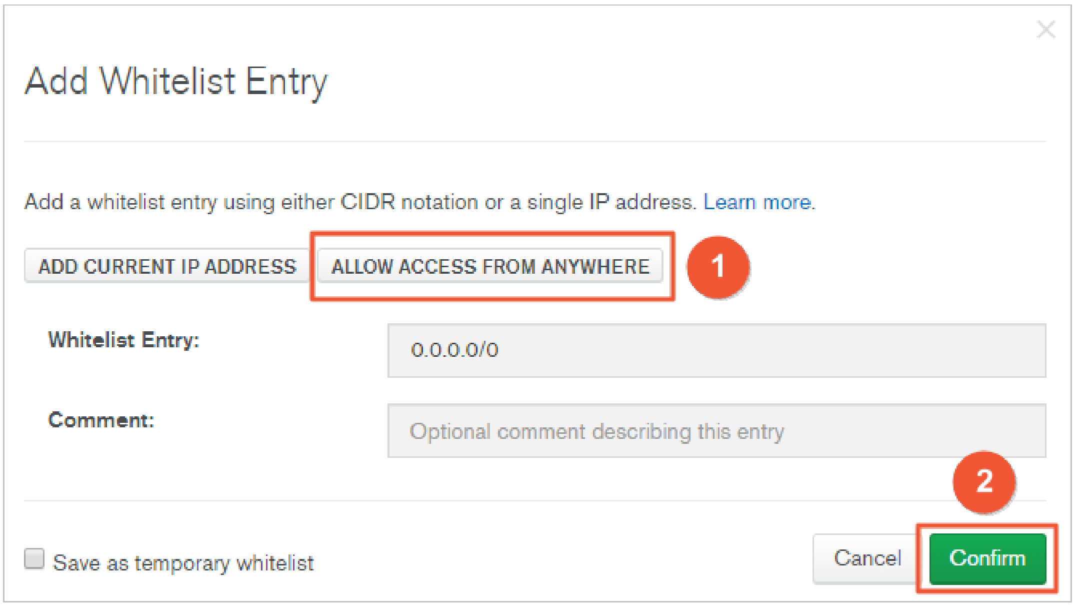
1. Log on to the MongoDB Atlas console.
2. In the left-side navigation pane, click **Network Access**. On the page that appears, click **ADD IP ADDRESS**.



You will only be able to connect to your cluster from the following list of IP Addresses:

Address	Comment	Status	Actions
[redacted]		● Active	⚙ EDIT 🗑 DELETE

3. In the dialog box that appears, click **ALLOW ACCESS FROM ANYWHERE**, and click **Confirm**.



Add Whitelist Entry

Add a whitelist entry using either CIDR notation or a single IP address. Learn more.

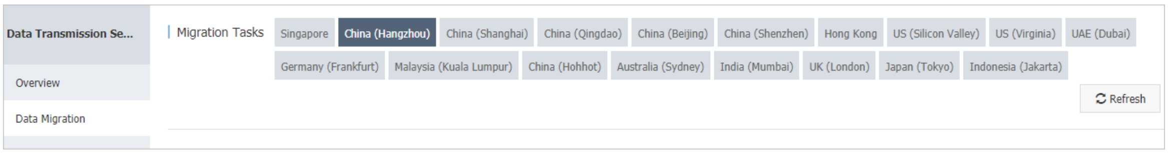
Whitelist Entry: 0.0.0.0/0

Comment: Optional comment describing this entry

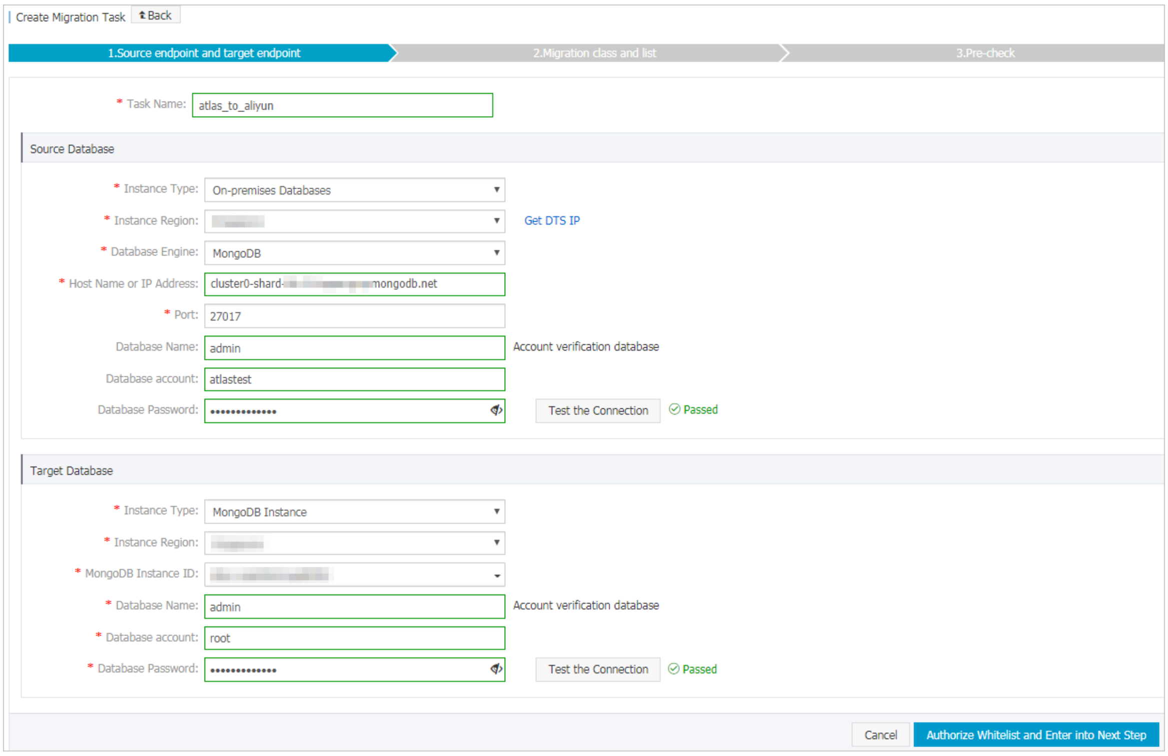
> **Note** This step allows all IP addresses to access the MongoDB Atlas database. Delete this rule after data migration is complete.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the ApsaraDB for MongoDB instance resides.



4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the source and destination databases.



Create Migration Task Back

1.Source endpoint and target endpoint | 2.Migration class and list | 3.Pre-check

* Task Name: atlas_to_aliyun

Source Database

* Instance Type: On-premises Databases

* Instance Region: [redacted] Get DTS IP

* Database Engine: MongoDB

* Host Name or IP Address: cluster0-shard-[redacted]mongodb.net

* Port: 27017

Database Name: admin Account verification database

Database account: atlastest

Database Password: •••••••••••••• Test the Connection ✓ Passed

Target Database

* Instance Type: MongoDB Instance

* Instance Region: [redacted]

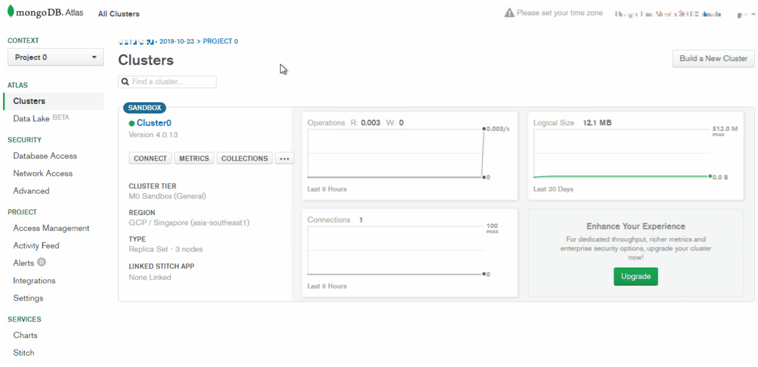
* MongoDB Instance ID: [redacted]

* Database Name: admin Account verification database

* Database account: root

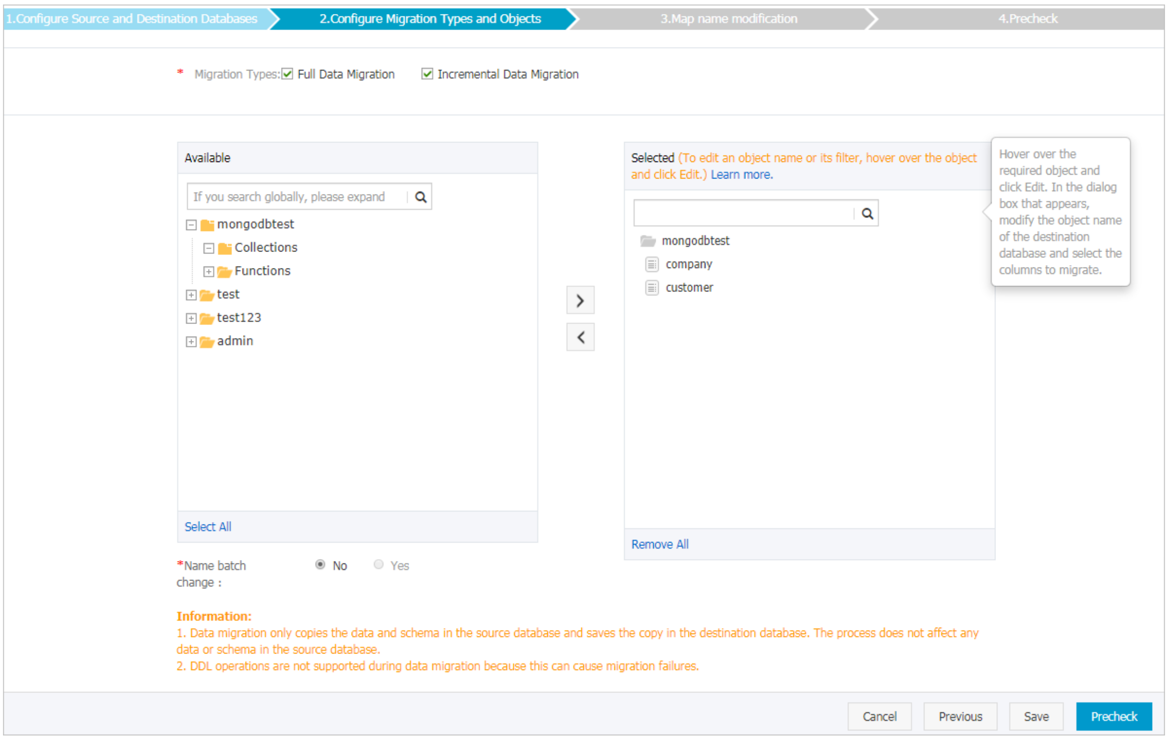
* Database Password: •••••••••••••• Test the Connection ✓ Passed

Cancel Authorize Whitelist and Enter into Next Step

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Database	Instance Type	Select **User-Created Database with Public IP Address**.
	Instance Region	If the instance type is set to **User-Created Database with Public IP Address**, you do not need to specify the **instance region**.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the endpoint of the PRIMARY node in the MongoDB Atlas database. You can obtain the endpoint in the MongoDB Atlas console, as shown in the following figure.  mongoDB. Atlas All Clusters Please set your time zone CONTEXT Project 0 ATLAS Clusters Data Lake BETA SECURITY Database Access Network Access Advanced PROJECT Access Management Activity Feed Alerts 0 Integrations Settings SERVICES Charts Stitch 2019-10-23 > PROJECT 0 Clusters Build a New Cluster Find a cluster... SANDBOX Cluster0 Version 4.0.13 CONNECT METRICS COLLECTIONS ... CLUSTER TIER M0 Sandbox (General) REGION GCP / Singapore (asia-southeast1) TYPE Replica Set - 3 nodes LINKED STITCH APP None Linked Operations R: 0.003 W: 0 0.003/s 0 Last 6 Hours Logical Size 12.1 MB 512.0 M max 0.0 B Last 30 Days Connections 1 100 max 0 Last 6 Hours Enhance Your Experience For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now! Upgrade

Section	Parameter	Description
	Port Number	Enter the service port number of the MongoDB Atlas database. The default port number is **27017**.
	Database Name	Enter the name of the authentication database. The database account is created in this database.
	Database Account	Enter the account of the MongoDB Atlas database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the database account. **Note** After you specify the source database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the source database parameters based on the check results.
	Encryption	Select **SSL-encrypted**.
Destination Database	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region where the ApsaraDB for MongoDB instance resides.
	MongoDB Instance ID	Select the ID of the ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database. The database account is created in this database. **Note** If the database account is root, enter admin.
	Database Account	Enter the database account of the ApsaraDB for MongoDB instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the destination database account. **Note** After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the specified parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

6. In the lower-right corner of the page, click **Set Whitelist and Next**.



* Migration Types: ☑ Full Data Migration ☑ Incremental Data Migration

Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.

Hover over the required object and click Edit. In the dialog box that appears, modify the object name of the destination database and select the columns to migrate.

*Name batch change : ● No ○ Yes

Information:
1. Data migration only copies the data and schema in the source database and saves the copy in the destination database. The process does not affect any data or schema in the source database.
2. DDL operations are not supported during data migration because this can cause migration failures.

> **Note** DTS adds the CIDR blocks of DTS servers to the whitelist of the ApsaraDB for MongoDB instance. This ensures that DTS servers can connect to the ApsaraDB for MongoDB instance.

7. Select the migration types and the objects to be migrated.

Setting	Description
Select the migration types	- To perform only full data migration, select only **Full Data Migration**. - To ensure service continuity during data migration, select both **Full Data Migration** and **Incremental Data Migration**. **Note** If **Incremental Data Migration** is not selected, do not write data to the source database during full data migration. This ensures data consistency between the source and destination databases.

Setting	Description
Select the objects to be migrated	○ Select objects from the **Available** section and click the › icon to move the objects to the **Selected** section. **Note** You cannot migrate data from the admin or local database. ○ You can select databases, collections, or functions as the objects to be migrated. ○ After an object is migrated to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are migrated to the ApsaraDB for MongoDB instance. For more information, see Object name mapping.

8. In the lower-right corner of the page, click **Precheck**.

> **Note**
>
> - Before you can start the data migration task, a precheck is performed. You can start the data migration task only after the task passes the precheck.
> - If the task fails to pass the precheck, click the ⓘ icon next to each failed item to view
>
>   details. Troubleshoot the issues based on the causes and run the precheck again.

9. After the task passes the precheck, click **Next** .
10. In the **Confirm Settings** dialog box, specify the **Channel Specification** parameter and select **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the migration task.

   - Full data migration

     We recommend that you do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task automatically stops.

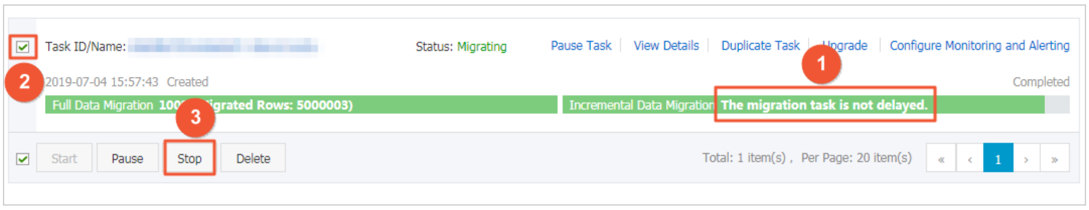
   - Incremental data migration

     An incremental data migration task does not automatically stop. You must manually stop the migration task.

     > **Note** Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the ApsaraDB for MongoDB instance.

     a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

b. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.


☑ Task ID/Name: [redacted]
Status: Migrating
Pause Task
View Details
Duplicate Task
Upgrade
Configure Monitoring and Alerting
2019-07-04 15:57:43 Created
Completed
Full Data Migration 100% (Migrated Rows: 5000003)
Incremental Data Migration
The migration task is not delayed.
1
2
3
☑
Start
Pause
Stop
Delete
Total: 1 item(s) , Per Page: 20 item(s)
« ‹ 1 › »

12. Switch your workloads to the ApsaraDB for MongoDB instance.

14.5.4. Migrate an Amazon DynamoDB database to ApsaraDB for MongoDB by using NimoShake

NimoShake, also known as DynamoShake, is a data synchronization tool developed by Alibaba Cloud. You can use this tool to migrate an Amazon DynamoDB database to ApsaraDB for MongoDB.

Prerequisites

An ApsaraDB for MongoDB instance is created. For more information, see 创建副本集实例 or 创建分片集群实例.

Context

This topic describes how to use NimoShake.

NimoShake is used to migrate data from an Amazon DynamoDB database. The destination must be an ApsaraDB for MongoDB instance.For more information, see NimoShake overview.

Precautions

Resources of the source and destination databases are occupied during full data migration. This may increase the load of the database servers. If you attempt to migrate a large volume of data or if the server specifications are insufficient, the databases may be overloaded or become unavailable. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours.

Terms

- Resumable upload: In a resumable upload task, data is split into multiple chunks. When transmission is interrupted due to network failures or other causes, the task can be resumed from where it was left off rather than restarting from the beginning.

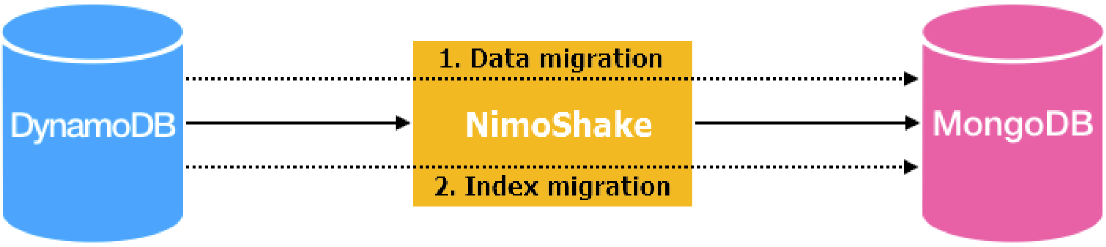
  > **Note** Resumable upload is supported in incremental migration, but not in full migration. If an incremental migration task is interrupted due to disconnection and the connection is recovered within a short time range, the task can be resumed. In some cases such as a long-period disconnection or the loss of a previous checkpoint, a full migration may be triggered.

- Checkpoint: Resumable upload is based on checkpoints. Default checkpoints are written to the ApsaraDB for MongoDB database named dynamo-shake-checkpoint. Each collection records a checkpoint list and the status_table collection records whether the current task is a full or incremental migration.

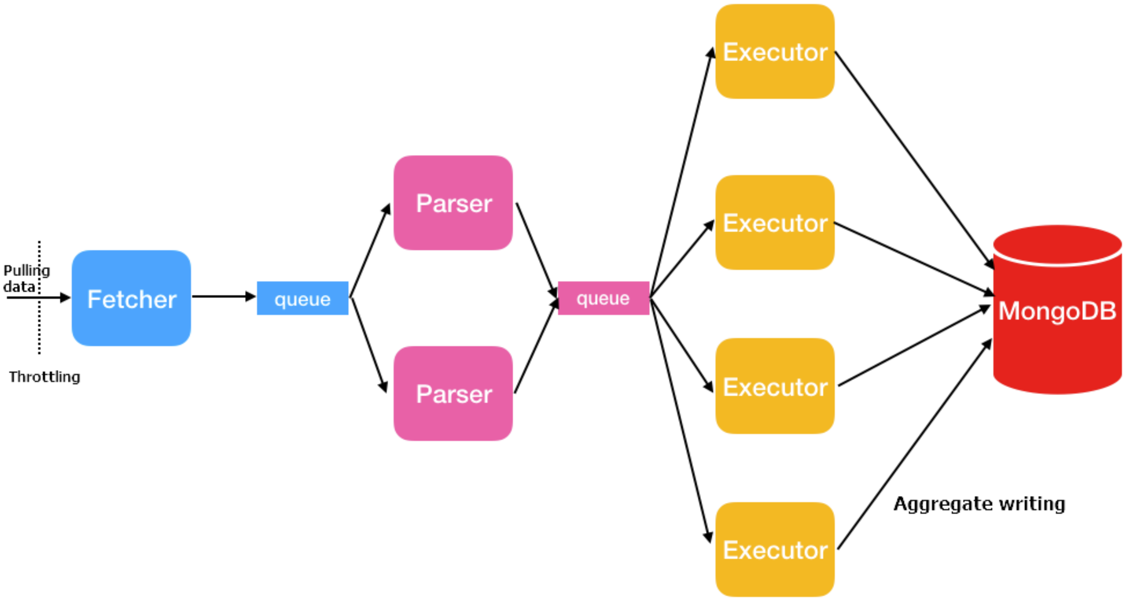
NimoShake features

NimoShake performs full migration the first time and then incremental migration.

- Full migration: contains data migration and index migration. The following figure shows the basic architecture of full migration.



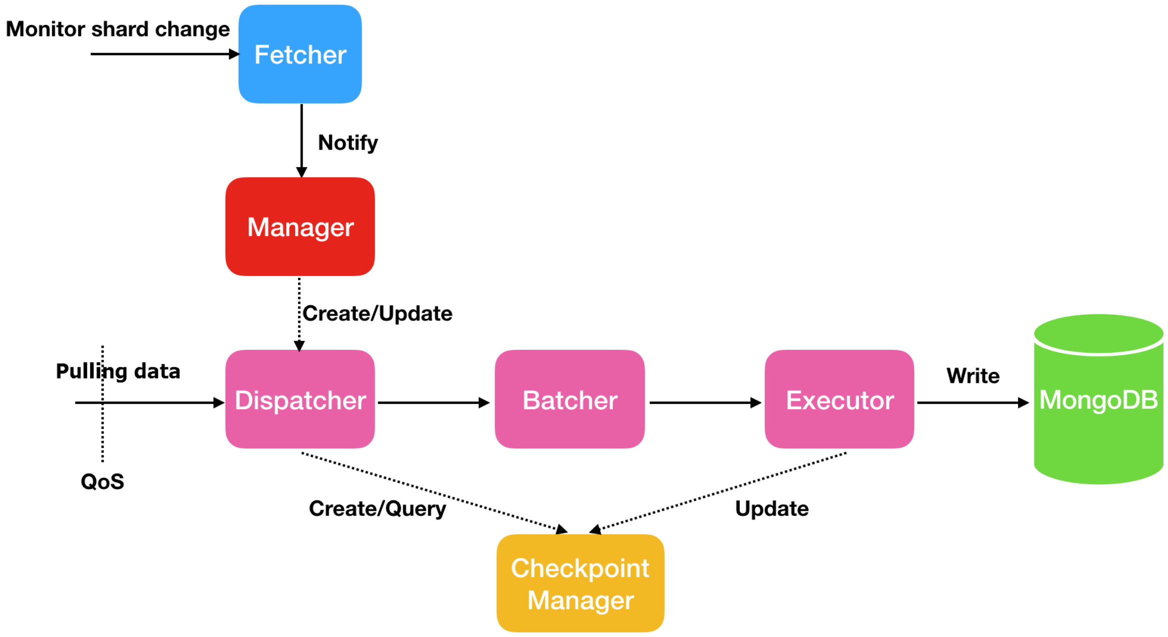
- Data migration: NimoShake uses multiple concurrent threads to pull source data, as shown in the following figure.



Thread	Description
Fetcher	Calls the protocol conversion driver provided by Amazon to capture data in the source collection in batches and then place the batches into queues until all source data is captured. Note Only one fetcher thread is provided.
Parser	Reads data from queues and parses data into the BSON structure. After data is parsed, the parser thread writes data to queues of the executor thread as entries. Multiple parser threads can be started and the default value is 2. You can specify the number of parser threads through the `FullDocumentParser` parameter.
Executor	Pulls data from queues and then aggregates and writes data to the ApsaraDB for MongoDB database. Up to 16 MB data in 1,024 entries can be aggregated. Multiple executor threads can be started and the default value is 4. You can specify the number of executor threads through the `FullDocumentConcurrency` parameter.

  - Index migration: NimoShake writes indexes after data migration is complete. Indexes include self-contained indexes and user-created indexes.
    - Self-contained indexes: If you have both a partition key and a sort key, NimoShake creates a unique composite index and writes the index to the ApsaraDB for MongoDB database. NimoShake also creates a hash index for the partition key and writes the index to the ApsaraDB for MongoDB database. If you have only a partition key, NimoShake writes a hash index and a unique index to the ApsaraDB for MongoDB database.
    - User-created indexes: If you have a user-created index, NimoShake creates a hash index based on the primary key and writes the index to the ApsaraDB for MongoDB database.

- Incremental migration: NimoShake migrates data but not the generated indexes. The following figure shows the basic architecture of incremental migration.



Thread	Description
Fetcher	Senses the changes of shards in a stream.
Manager	Sends messages or creates a dispatcher to process messages. One shard corresponds to one dispatcher.
Dispatcher	Pulls incremental data from the source. In resumable upload, data is pulled from the last checkpoint instead of the very beginning.
Batcher	Parses and encapsulates the incremental data pulled from the dispatcher thread.
Executor	Writes the encapsulated data to the ApsaraDB for MongoDB database and updates the checkpoint.

Migrate an Amazon DynamoDB database to ApsaraDB for MongoDB

This section uses Ubuntu to describe how to use NimoShake to migrate an Amazon DynamoDB database to ApsaraDB for MongoDB.

1. Run the following command to download the NimoShake package and wait until the package is downloaded:

   ```
   wget https://github.com/alibaba/NimoShake/releases/download/release-v1.0.0-20191015/nimo.tar.gz
   ```

   > **Note** We recommend that you download the latest NimoShake package. For more information about the download address, visit NimoShake.

2. Run the following command to decompress the NimoShake package:

   ```
   tar zxvf nimo.tar.gz
   ```

3. After decompression, run the `cd nimo` command to access the nimo folder.
4. Run the `vi nimo-shake.conf` command to open the NimoShake configuration file.
5. Configure the following parameters for NimoShake.

Parameter	Description	Example
id	The ID of the migration task, which is custom. The ID is used to display pid files and other information, such as the log name of the migration task, the name of the database that stores checkpoint information, and the name of the destination database.	`id = nimo-shake`
log.file	The path of the log file. If you do not configure this parameter, logs are displayed in stdout.	`log.file = nimo-shake.log`
log.level	The severity of logs. Valid values: ○ none: No logs are collected. ○ error: logs that contain error information. ○ warn: logs that contain warning information. ○ info: logs that contain system status information. ○ debug: logs that contain debugging information. Default value: info.	`log.level = info`

Parameter	Description	Example
log.buffer	Specifies whether to enable log buffering. Valid values: ○ true. Log buffering ensures high performance. However, several latest log entries may be lost when you disable log buffering. ○ false. If log buffering is disabled, performance is low, but all log entries are displayed. Default value: true.	`log.buffer = true`
system_profile	The pprof port. It is used for debugging and displaying stackful coroutine information.	`system_profile = 9330`
http_profile	The HTTP port. After this port is enabled, you can view the current status of NimoShake over the Internet.	`http_profile = 9340`
sync_mode	The type of data migration. Valid values: ○ all: both full and incremental migration. ○ full: only full migration. ○ incr: only incremental migration. Default value: all. **Note** Both full and incremental migration is performed by default. If you only want to perform either full migration or incremental migration, change the value to full or incr.	`sync_mode = all`
source.access_key_id	The AccessKey ID used to access Amazon DynamoDB.	`source.access_key_id = xxxxxxxxxxx`
source.secret_access_key	The AccessKey secret used to access Amazon DynamoDB.	`source.secret_access_key = xxxxxxxxxx`
source.session_token	The temporary key used to access Amazon DynamoDB. If no temporary key is available, you can skip this parameter.	`source.session_token = xxxxxxxxxx`

Parameter	Description	Example
source.region	The region where the Amazon DynamoDB database is located. If no region is available, you can skip this parameter.	`source.region = us-east-2`
source.session.max_retries	The maximum number of retries after a session failure.	`source.session.max_retries = 3`
source.session.timeout	The session timeout. 0 indicates that the session timeout is disabled. Unit: milliseconds.	`source.session.timeout = 3000`
filter.collection.white	The names of collections to be migrated. For example, `filter.collection.white = c1;c2` indicates that the c1 and c2 collections will be migrated and the rest collections will be filtered out. ⓘ **Note** You cannot specify both the filter.collection.white and filter.collection.black parameters. Otherwise, all collections will be migrated.	`filter.collection.white = c1;c2`
filter.collection.black	The names of collections to be filtered out. For example, `filter.collection.black = c1;c2` indicates that the c1 and c2 collections will be filtered out and the rest collections will be migrated. ⓘ **Note** You cannot specify both the filter.collection.white and filter.collection.black parameters. Otherwise, all collections will be migrated.	`filter.collection.black = c1;c2`
qps.full	The maximum number of the `scan` commands to be called per second in full migration. It is used to limit the frequency of the `scan` command. Default value: 1000.	`qps.full = 1000`
qps.full.batch_num	The number of data entries pulled per second in full migration. Default value: 128.	`qps.full.batch_num = 128`

Parameter	Description	Example
qps.incr	The maximum number of the `GetRecords` commands to be called per second in incremental migration. It is used to limit the frequency of the `GetRecords` command. Default value: 1000.	`qps.incr = 1000`
qps.incr.batch_num	The number of data entries pulled per second in incremental migration. Default value: 128.	`qps.incr.batch_num = 128`
target.type	The category of the destination database. Valid values: ○ mongodb: an ApsaraDB for MongoDB instance. ○ aliyun_dynamo_proxy: an ApsaraDB for MongoDB instance that is compatible with the DynamoDB protocol. **Note** The aliyun_dynamo_proxy option is only available at the China site (aliyun.com).	`target.type = mongodb`
target.mongodb.type	The category of the destination ApsaraDB for MongoDB instance. Valid values: ○ replica: the replica set instance. ○ sharding: the sharded cluster instance.	`target.mongodb.type = sharding`

Parameter	Description	Example
target.address	The connection string of the destination ApsaraDB for MongoDB instance. The destination must be an ApsaraDB for MongoDB instance or an ApsaraDB for MongoDB instance that is compatible with the DynamoDB protocol. For more information about ApsaraDB for MongoDB connection string URIs, see Overview of replica set instance connections or Overview of sharded cluster instance connections. **Note** ApsaraDB for MongoDB instances that are compatible with the DynamoDB protocol are only available at the China site (aliyun.com).	`target.address = mongodb://username:password@s-*****-pub.mongodb.rds.aliyuncs.com:3717`
target.db.exist	Specifies how to handle a second collection with the same name on the destination. Valid values: ○ rename: NimoShake will rename a second collection with the same name and add a timestamp suffix to the name. For example, NimoShake changes c1 to c1.2019-07-01Z12:10:11. **Warning** This operation modifies the names of destination collections and may affect the business. You must make preparations before data migration. ○ drop: NimoShake will delete a second collection with the same name. If this parameter is not specified and the destination already contains a collection with the same name, the migration task is terminated and an error message is returned.	`target.mongodb.exist = drop`
full.concurrency	The maximum number of collections that can be migrated concurrently in full migration. Default value: 4.	`full.concurrency = 4`

Parameter	Description	Example
full.document.concurrency	A parameter for full migration. It specifies the maximum number of threads used concurrently to write documents in a collection to the destination. Default value: 4.	`full.document.concurrency = 4`
full.document.parser	A parameter for full migration. It specifies the maximum number of parser threads used concurrently to convert the Dynamo protocol to the corresponding protocol on the destination. Default value: 2.	`full.document.parser = 2`
full.enable_index.user	A parameter for full migration. It specifies whether to migrate user-created indexes. Valid values: ○ true ○ false	`full.enable_index.user = true`
full.executor.insert_on_dup_update	A parameter for full migration. Specifies whether to change the `INSERT` statement to the `UPDATE` statement if the same keys occur at the destination. Valid values: ○ true ○ false	`full.executor.insert_on_dup_update = true`
increase.executor.insert_on_dup_update	A parameter for incremental migration. Specifies whether to change the `INSERT` statement to the `UPDATE` statement if the same keys occur at the destination. Valid values: ○ true ○ false	`increase.executor.insert_on_dup_update = true`

Parameter	Description	Example
increase.executor.upsert	A parameter for incremental migration. Specifies whether to change the `UPDATE` statement to the `UPSERT` statement if no key is provided at the destination. Valid values: ○ true ○ false Note The `UPSERT` statement checks whether the target key exists. If yes, the `UPDATE` statement is executed. If not, the `INSERT` statement is executed.	`increase.executor.upsert = true`
convert.type	A parameter for incremental migration. It specifies whether to convert the Dynamo protocol. Valid values: ○ raw: writes data directly without conversion of the Dynamo protocol. ○ change: converts the Dynamo protocol. For example, NimoShake converts `{"hello":"1"}` to `{"hello":1}` .	`convert.type = change`
increase.concurrency	A parameter for incremental migration. It specifies the maximum number of shards that can be captured concurrently. Default value: 16.	`increase.concurrency = 16`

Parameter	Description	Example
checkpoint.type	The type of the storage that stores checkpoint information. Valid values: - mongodb: Checkpoint information is store in the ApsaraDB for MongoDB database. Only when the `target.type` parameter is set to `mongodb`, the checkpoint.address and checkpoint.db parameters are valid. - file: Checkpoint information is store in the local computer.	`checkpoint.type = mongodb`
checkpoint.address	The address used to store checkpoint information. - If the `checkpoint.type` parameter is set to `mongodb`, enter the connection string URI of the ApsaraDB for MongoDB database. If you do not specify this parameter, checkpoint information is stored in the destination ApsaraDB for MongoDB database. For more information about ApsaraDB for MongoDB connection string URIs, see Overview of replica set instance connections or Overview of sharded cluster instance connections. - If the `checkpoint.type` parameter is set to `file`, enter a relative path based on the path of the NimoShake file. Example: *checkpoint*. If you do not specify this parameter, checkpoint information is stored in checkpoint folder.	`checkpoint.address = mongodb://username:password@s-*****-pub.mongodb.rds.aliyuncs.com:3717`
checkpoint.db	The name of the ApsaraDB for MongoDB database that stores checkpoint information. If you do not specify this parameter, the database name is in the default format of `<Task ID>-checkpoint`. Example: nimoshake-checkpoint.	`checkpoint.db = nimoshake-checkpoint`

6. Run the following command to start data migration by using the configured nimo-shake.conf file:

```
./nimo-shake.linux -conf=nimo-shake.conf
```

Note After the full migration is complete, `full sync done!` is displayed on the screen. If the migration is terminated due to an error, NimoShake automatically stops and the corresponding error message is displayed on the screen for you to troubleshoot the error.

14.5.5. Use NimoFullCheck to check data consistency after migration

NimoFullCheck is a tool developed by Alibaba Cloud to check data consistency between an Amazon DynamoDB database and an ApsaraDB for MongoDB database. This topic describes how to use NimoFullCheck to check data consistency after you migrate data from an Amazon DynamoDB database to an ApsaraDB for MongoDB database.

Prerequisites

Data is migrated from an Amazon DynamoDB database to an ApsaraDB for MongoDB database by using NimoShake. For more information, see Migrate an Amazon DynamoDB database to ApsaraDB for MongoDB by using NimoShake.

Context

After you migrate data from an Amazon DynamoDB database to an ApsaraDB for MongoDB database, you can use NimoFullCheck to check data consistency between the two databases.

The check consists of the following two steps:

- Brief check: checks whether the number of items in a table in the Amazon Dynamo database is equal to the number of documents in the corresponding collection in the ApsaraDB for MongoDB database. If the numbers are different, the check terminates and an error message is returned. You can locate issues based on the returned error message.
- Precise check: precisely compares data in the two databases after the brief check is passed. NimoFullCheck fetches data from the Amazon Dynamo database and parses the data. If the data contains unique indexes, NimoFullCheck compares the data with that in the destination ApsaraDB for MongoDB database based on the unique indexes. If the data does not contain unique indexes, NimoFullCheck compares all data entries in two databases one by one, which is slow.

Usage notes

- NimoFullCheck only supports consistency check for full data migration. If you check data consistency after an incremental data synchronization, the result is inconsistent.
- NimoFullCheck uses data in the ApsaraDB for MongoDB database as the baseline for check. In other words, NimoFullCheck checks whether data in the Amazon DynamoDB database is consistent with that in the ApsaraDB for MongoDB database.

Procedure

The following procedure assumes that you run NimoFullCheck in the Ubuntu operating system.

1. Run the following command to download the NimoShake package:

```
wget https://github.com/alibaba/NimoShake/releases/download/release-v1.0.0-20191015/nimo.tar.gz
```

> Note We recommend that you download the latest NimoShake package. For more information, see NimoShake.

2. Run the following command to decompress the NimoShake package:

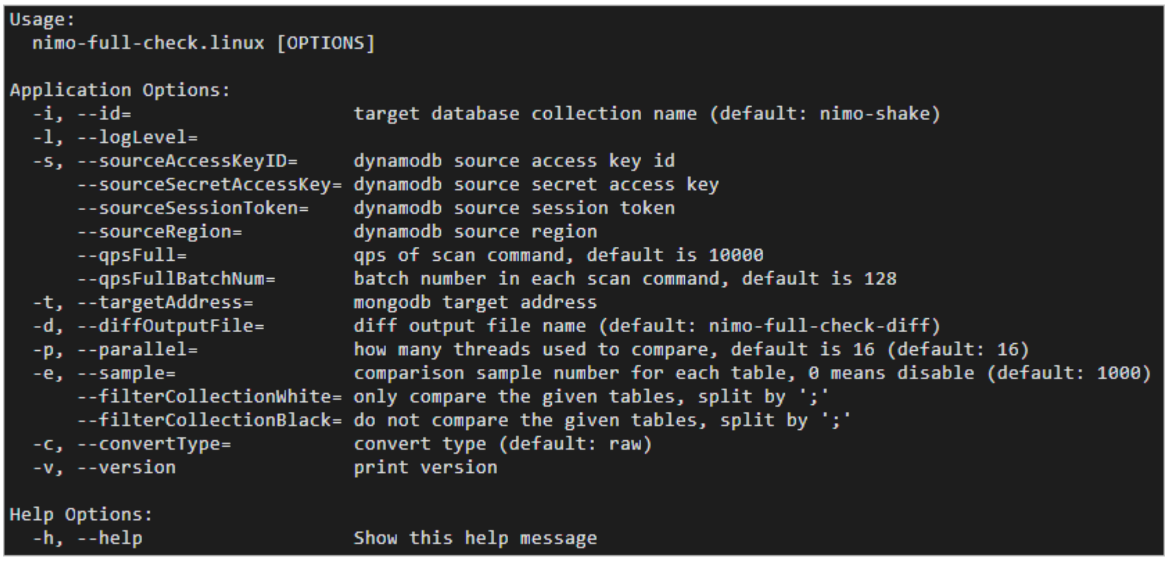
```
tar zxvf nimo.tar.gz
```

3. After you decompress the package, run the `cd nimo` command to go to the nimo directory.

4. Run the following command to start NimoFullCheck with required parameters:

```
./nimo-full-check.linux --<Parameter 1>=<Value 1> --<Parameter 2>=<Value 2>
```

The following table describes the parameters of NimoFullCheck.



```
Usage:
  nimo-full-check.linux [OPTIONS]

Application Options:
  -i, --id=                      target database collection name (default: nimo-shake)
  -l, --logLevel=
  -s, --sourceAccessKeyID=       dynamodb source access key id
      --sourceSecretAccessKey=   dynamodb source secret access key
      --sourceSessionToken=      dynamodb source session token
      --sourceRegion=            dynamodb source region
      --qpsFull=                 qps of scan command, default is 10000
      --qpsFullBatchNum=         batch number in each scan command, default is 128
  -t, --targetAddress=           mongodb target address
  -d, --diffOutputFile=          diff output file name (default: nimo-full-check-diff)
  -p, --parallel=                how many threads used to compare, default is 16 (default: 16)
  -e, --sample=                  comparison sample number for each table, 0 means disable (default: 1000)
      --filterCollectionWhite=   only compare the given tables, split by ';'
      --filterCollectionBlack=   do not compare the given tables, split by ';'
  -c, --convertType=             convert type (default: raw)
  -v, --version                  print version

Help Options:
  -h, --help                     Show this help message
```

Parameter	Description	Example
id	The ID of the migration task. Set the value to the ID of the migration task that is specified when you used NimoShake to migrate data. For more information, see Migrate an Amazon DynamoDB database to ApsaraDB for MongoDB by using NimoShake.	`--id = nimo-shake`

Parameter	Description	Example
logLevel	The level of the logs to be generated. Valid values: ∘ none: does not generate logs. ∘ error: generates logs that contain error messages. ∘ warn: generates logs that contain warnings. ∘ info: generates logs that indicate system status. ∘ debug: generates logs that contain debugging information. Default value: info.	`--logLevel = info`
sourceAccessKeyID	The access key ID used to connect to the source Amazon DynamoDB database.	`--sourceAccessKeyID = xxxxxxxxxx`
sourceSecretAccessKey	The secret access key used to connect to the source Amazon DynamoDB database.	`--sourceSecretAccessKey = xxxxxxxxxx`
sourceSessionToken	Optional. The session token used to access the source Amazon DynamoDB database.	`--sourceSessionToken = xxxxxxxxxx`
sourceRegion	Optional. The region where the source Amazon DynamoDB database resides.	`--sourceRegion = us-east-2`
qpsFull	The number of times that the `Scan` command is run on tables per second. Default value: 10000.	`--qpsFull = 10000`
qpsFullBatchNum	The number of data entries to fetch per second. Default value: 128.	`--qpsFullBatchNum = 128`
targetAddress	The endpoint of the destination ApsaraDB for MongoDB database. For more information about how to view the endpoint, see Overview of replica set instance connections or Overview of sharded cluster instance connections. Example: mongodb://username:password@s-*****-pub.mongodb.rds.aliyuncs.com:3717.	`--targetAddress = mongodb://username:password@s-*****-pub.mongodb.rds.aliyuncs.com:3717`

Parameter	Description	Example
diffOutputFile	The name of the file that stores the information about inconsistent data. If you do not specify this parameter, the default file name nimo-full-check-diff is used.	--diffOutputFile = nimo-full-check-diff
parallel	The number of threads to be used for consistency check. Default value: 16.	--parallel = 16
sample	The maximum number of documents to be checked in each collection. A value of 0 indicates that all documents in a collection are checked. Default value: 1000. A value of 1000 indicates that a maximum of 1,000 documents can be checked in a collection.	--sample = 1000
filterCollectionWhite	The collection whitelist for consistency check. Set the value to the names of the collections that must be checked. Example: --filterCollectionWhite = c1;c2 , indicating that only collections c1 and c2 are checked.	--filterCollectionWhite = ci;c2
filterCollectionBlack	The collection blacklist for consistency check. Set the value to the names of the collections that do not need to be checked. Example: --filterCollectionBlack = c1;c2 , indicating that all collections other than c1 and c2 are checked.	--filterCollectionBlack = ci;c2

Parameter	Description	Example
convertType	Specifies whether the source data, which uses the Dynamo protocol, was converted during the migration. Valid values: - raw: Data was written to the destination database without conversion. - change: Data was converted before it is written to the destination database. For example, `{"hello":"1"}` was converted to `{"hello": 1}` . ⓘ **Note** The value of the parameter must be the same as that specified for migration. If the values are different, the check fails.	`--convertType=change`
version	Displays the version number of NimoFullCheck. ⓘ **Note** You do not need to specify a value for the parameter. If you need to display the version number, add the `--version` parameter to the command.	`--version`
help	Displays all the parameters that are supported by NimoFullCheck.	`--help`

ⓘ **Note** If the check is successful, the message `full check done!` is returned. If the check terminates due to an error, NimoFullCheck exits and an error message is returned. You can locate issues based the returned error message.

14.6. Data synchronization

14.6.1. Use MongoShake to implement one-way synchronization between ApsaraDB for MongoDB replica set instances

You can use the open source MongoShake tool developed by Alibaba Cloud to synchronize data between MongoDB databases. This tool can be used in scenarios such as data analysis, disaster recovery, and active-active replication. This topic describes how to configure MongoShake to synchronize data between ApsaraDB for MongoDB replica set instances in real time.

MongoShake overview

MongoShake is a general-purpose Platform as a Service (PaaS) tool, which is written in the Go language by Alibaba Cloud. MongoShake reads the oplogs of a MongoDB database and replicates data based on the oplogs to meet specific requirements.

MongoShake also allows you to subscribe to and consume MongoDB logs. You can connect to MongoShake by using multiple methods such as SDKs, Kafka, and MetaQ. MongoShake is suitable for scenarios such as log subscription, data synchronization across data centers, and asynchronous cache eviction.

Note

- For more information about MongoShake, visit MongoShake homepage on GitHub.
- If you want to implement two-way data synchronization between replica set instances, you can submit a ticket.

Supported databases

Source database	Destination database
Self-managed MongoDB database hosted on Elastic Compute Service (ECS)	Self-managed MongoDB database hosted on ECS
Self-managed MongoDB database hosted on an on-premises machine	Self-managed MongoDB database hosted on an on-premises machine
ApsaraDB for MongoDB instance	ApsaraDB for MongoDB instance
MongoDB database on a third-party cloud	MongoDB database on a third-party cloud

Precautions

- Do not perform data definition language (DDL) operations on the source database before full data synchronization is complete. Otherwise, data inconsistency may occur.
- You cannot use MongoShake to synchronize data in the admin and local databases.

Required permissions on databases

Database	Required permission
Source ApsaraDB for MongoDB instance	readAnyDatabase permissions, read permissions on the local database, and read/write permissions on the mongoshake database Note The mongoshake database is created by MongoShake at the source when the incremental synchronization task starts.

Database	Required permission
Destination ApsaraDB for MongoDB instance	readWriteAnyDatabase or read/write permissions on the destination database

> **Note** For more information about how to create and authorize MongoDB users, see Manage user permissions on MongoDB databases or visit db.createUser().

Preparations

1. For best synchronization performance, make sure that the source ApsaraDB for MongoDB replica set instance resides in a VPC. If the source instance resides in the classic network, switch the network type to VPC. For more information, see Switch the network type of an ApsaraDB for MongoDB instance.
2. Create an ApsaraDB for MongoDB replica set instance as the synchronization destination. Select the same VPC as the one used by the source ApsaraDB for MongoDB replica set instance to minimize network latency. For more information, see 创建副本集实例.
3. Create an ECS instance to run MongoShake. Select the same VPC as the one used by the source ApsaraDB for MongoDB instance to minimize network latency. For more information, see Create an ECS instance.
4. Add the private IP address of the ECS instance to the whitelists of the source and destination ApsaraDB for MongoDB instances. Make sure that the ECS instance can connect to the source and destination ApsaraDB for MongoDB instances. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

> **Note** If the network type does not meet the preceding requirements, you can apply for public endpoints for the source and destination ApsaraDB for MongoDB instances. Then, add the public IP address of the ECS instance to the whitelists of the source and destination ApsaraDB for MongoDB instances. This way, you can synchronize data by using the Internet. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance and Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

Procedure

By default, the */root/mongoshake* directory is used as the installation directory for MongoShake in this example.

1. Log on to the ECS instance. For more information, see Connect to a Linux instance by using a username and password.
2. Run the following command to download the MongoShake package and rename the package as `mongoshake.tar.gz` :

```
wget "http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/196977/jp_ja/1608863913991/
mongo-shake-v2.4.16.tar.gz" -O mongoshake.tar.gz
```

Note The download URL for MongoShake V2.4.16 is used in this example. To download the latest version of MongoShake, visit Releases.

3. Run the following command to decompress the MongoShake package to the */root/mongoshake* directory:

```
tar zxvf mongoshake.tar.gz && mv mongo-shake-v2.4.16 /root/mongoshake && cd /root/mongoshake
```

4. Run the `vi collector.conf` command to modify the *collector.conf* configuration file of MongoShake. The following table describes the parameters that you must configure to synchronize data between ApsaraDB for MongoDB instances.

Parameter	Description	Example
mongo_urls	The connection string URI of the source ApsaraDB for MongoDB instance. Note - We recommend that you use a VPC endpoint to minimize network latency. - For more information about the format of a connection string URI, see Overview of replica set instance connections.	`mongo_urls = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717` Note The password cannot contain at signs (@). Otherwise, the connection may fail.
tunnel.address	The connection string URI of the destination ApsaraDB for MongoDB instance. Note - We recommend that you use a VPC endpoint to minimize network latency. - For more information about the format of a connection string URI, see Overview of replica set instance connections.	`tunnel.address = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717` Note The password cannot contain at signs (@). Otherwise, the connection may fail.

Parameter	Description	Example
sync_mode	The data synchronization method. Valid values: ○ all: performs both full data synchronization and incremental data synchronization. ○ full: performs only full data synchronization. ○ incr: performs only incremental data synchronization. ⓘ Note The default value is incr.	sync_mode = all

ⓘ Note For more information about all parameters in the *collector.conf* file, see the Appendix section of this topic.

5. Run the following command to start the data synchronization task and generate the log information:

```
./collector.linux -conf=collector.conf -verbose
```

6. Check the log information. If the following log is displayed, it indicates that the full data synchronization is complete and the incremental data synchronization starts.

```
[09:38:57 CST 2019/06/20] [INFO] (mongoshake/collector.( *ReplicationCoordinator).Run:80) finish full s
ync, start incr sync with timestamp: fullBeginTs[1560994443], fullFinishTs[1560994737]
```

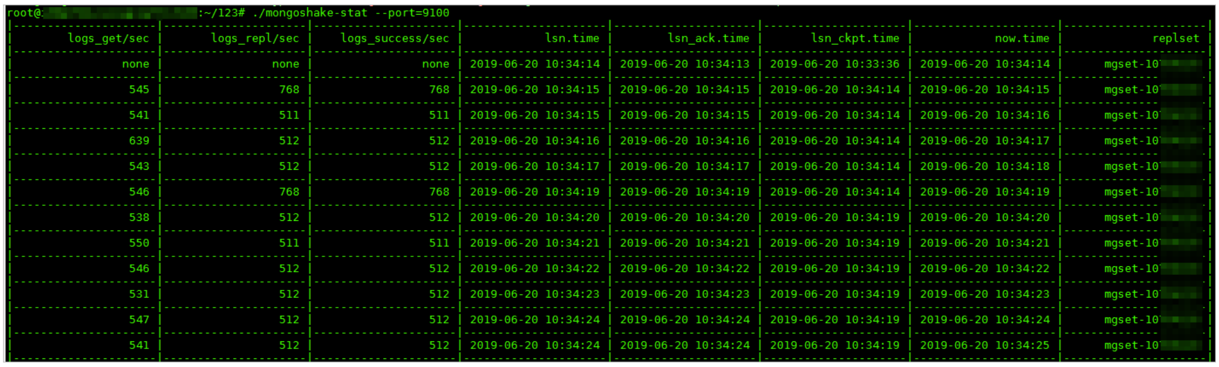
Monitor the MongoShake status

When the incremental data synchronization starts, you can open a command line window to monitor MongoShake.

```
cd /root/mongoshake && ./mongoshake-stat --port=9100
```

ⓘ Note `mongoshake-stat` is a Python script. Before you run the script, make sure that Python 2.7 is installed. For more information, visit Python official website.

The following figure shows sample monitoring information about MongoShake.



```
root@[redacted]:~/123# ./mongoshake-stat --port=9100
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|        logs_get/sec |       logs_repl/sec |    logs_success/sec |            lsn.time |        lsn_ack.time |       lsn_ckpt.time |            now.time |             replset |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                none |                none |                none | 2019-06-20 10:34:14 | 2019-06-20 10:34:13 | 2019-06-20 10:33:36 | 2019-06-20 10:34:14 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 545 |                 768 |                 768 | 2019-06-20 10:34:15 | 2019-06-20 10:34:15 | 2019-06-20 10:34:14 | 2019-06-20 10:34:15 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 541 |                 511 |                 511 | 2019-06-20 10:34:15 | 2019-06-20 10:34:15 | 2019-06-20 10:34:14 | 2019-06-20 10:34:16 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 639 |                 512 |                 512 | 2019-06-20 10:34:16 | 2019-06-20 10:34:16 | 2019-06-20 10:34:14 | 2019-06-20 10:34:17 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 543 |                 512 |                 512 | 2019-06-20 10:34:17 | 2019-06-20 10:34:17 | 2019-06-20 10:34:14 | 2019-06-20 10:34:18 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 546 |                 768 |                 768 | 2019-06-20 10:34:19 | 2019-06-20 10:34:19 | 2019-06-20 10:34:14 | 2019-06-20 10:34:19 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 538 |                 512 |                 512 | 2019-06-20 10:34:20 | 2019-06-20 10:34:20 | 2019-06-20 10:34:19 | 2019-06-20 10:34:20 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 550 |                 511 |                 511 | 2019-06-20 10:34:21 | 2019-06-20 10:34:21 | 2019-06-20 10:34:19 | 2019-06-20 10:34:21 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 546 |                 512 |                 512 | 2019-06-20 10:34:22 | 2019-06-20 10:34:22 | 2019-06-20 10:34:19 | 2019-06-20 10:34:22 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 531 |                 512 |                 512 | 2019-06-20 10:34:23 | 2019-06-20 10:34:23 | 2019-06-20 10:34:19 | 2019-06-20 10:34:23 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 547 |                 512 |                 512 | 2019-06-20 10:34:24 | 2019-06-20 10:34:24 | 2019-06-20 10:34:19 | 2019-06-20 10:34:24 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
|                 541 |                 512 |                 512 | 2019-06-20 10:34:24 | 2019-06-20 10:34:24 | 2019-06-20 10:34:19 | 2019-06-20 10:34:25 | mgset-10[redacted] |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
```

Parameter	Description
logs_get/sec	The number of oplogs obtained per second.
logs_repl/sec	The number of oplogs for replay operations performed per second.
logs_success/sec	The number of oplogs for successful replay operations per second.
lsn.time	The time when the last oplog was sent.
lsn_ack.time	The time when the destination database acknowledges the write operation.
lsn_ckpt.time	The time when the last checkpoint was generated.
now.time	The current time.
replset	The name of the replica set instance where the source database resides.

Appendix

All parameters in the collector.conf file

Section	Parameter	Description	Example
N/A	conf.version	The version number of the configuration file. Do not change the value.	`conf.version = 4`
	id	The ID of the synchronization task. This value is customizable. The global configuration includes the log file name, the name of the database that stores the checkpoint information, and the name of the destination database.	`id = mongoshake`

Section	Parameter	Description	Example
	master_quorum	Specifies whether the MongoShake node is the active node in high availability scenarios. If you use the active MongoShake node and standby MongoShake node to synchronize data from the same database, you must set this parameter to `true` for the active MongoShake node. Valid values: • true • false Note The default value is false.	`master_quorum = false`
	full_sync.http_port	The HTTP port used to view the status of full data synchronization in MongoShake over the Internet. Note The default value is 9101.	`full_sync.http_port = 9101`
	incr_sync.http_port	The HTTP port used to view the status of incremental data synchronization in MongoShake over the Internet. Note The default value is 9100.	`incr_sync.http_port = 9100`
	system_profile_port	The profiling port used to view internal stack information.	`system_profile_port = 9200`

Section	Parameter	Description	Example
	log.level	The level of the logs to be generated. Valid values: - error: generates logs that contain error messages. - warning: generates logs that contain warnings. - info: generates logs that indicate system status. - debug: generates logs that contain debugging information. The default value is info.	`log.level = info`
	log.dir	The directory where the log file and PID file are stored. If you do not specify a value, the log file and PID file are stored in the logs directory in the working directory. **Note** This parameter must be set to an absolute path.	`log.dir = ./logs/`
	log.file	The name of the log file. This value is customizable. **Note** The default value is collector.log.	`log.file = collector.log`
	log.flush	Specifies whether to display every log entry on the screen. Valid values: - true: Every log entry is displayed on the screen. This ensures that no log entry is missing on the screen but compromises the performance. - false: Not every log entry is displayed on the screen. This ensures the performance but some log entries may be missing on the screen. **Note** The default value is false.	`log.flush = false`

Section	Parameter	Description	Example
	sync_mode	The data synchronization method. Valid values: - all: performs both full data synchronization and incremental data synchronization. - full: performs only full data synchronization. - incr: performs only incremental data synchronization. **Note** The default value is incr.	`sync_mode = all`
	mongo_urls	The connection string URI of the source ApsaraDB for MongoDB instance. **Note** - We recommend that you use a VPC endpoint to minimize network latency. - For more information about the format of a connection string URI, see Overview of replica set instance connections or Overview of sharded cluster instance connections.	`mongo_urls = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717`
	mongo_cs_url	The endpoint of the Configserver node. If the source ApsaraDB for MongoDB instance is a sharded cluster instance, you must specify this parameter. For more information about how to apply for an endpoint for a Configserver node, see Apply for a connection string of a shard or Configserver node.	`mongo_cs_url = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx-csxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx-csxxx.mongodb.rds.aliyuncs.com:3717/admin`

Section	Parameter	Description	Example
Global configuration options	mongo_s_url	The endpoint of the Mongos node. If the source ApsaraDB for MongoDB instance is a sharded cluster instance, you must specify this parameter. You must specify the endpoint of at least one Mongos node. Separate the endpoints of multiple Mongos nodes with commas (,). For more information about how to apply for an endpoint for a Mongos node, see Apply for a connection string of a shard or Configserver node.	`mongos_s_url = mongodb://root:Ftxxxxxx@s-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,s-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717/admin`
	tunnel	The type of the tunnel used for synchronization. Valid values: • direct: directly synchronizes data to the destination ApsaraDB for MongoDB instance. • rpc: synchronizes data by using NET/RPC. • tcp: synchronizes data by using TCP. • file: synchronizes data by transferring files. • kafka: synchronizes data by using Kafka. • mock: only used for testing without writing data to the tunnel. **Note** Th default value is direct.	`tunnel = direct`

Section	Parameter	Description	Example
	tunnel.address	The address used to connect to the destination ApsaraDB for MongoDB instance through the tunnel. - If the tunnel parameter is set to `direct` , set the value to the connection string URI of the destination ApsaraDB for MongoDB instance. - If the tunnel parameter is set to `rpc` , set the value to the receiver socket address used in the RPC connection to the destination ApsaraDB for MongoDB instance. - If the tunnel parameter is set to `tcp` , set the value to the receiver socket address used in the TCP connection to the destination ApsaraDB for MongoDB instance. - If the tunnel parameter is set to `file` , set the value to the file path in the destination ApsaraDB for MongoDB instance. - If the tunnel parameter is set to `kafka` , set the value to the broker server addresses of Kafka. Example: `topic@brokers1,brokers2` . - If the tunnel parameter is set to `mock` , you do not need to set this parameter.	`tunnel.address = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717`

Section	Parameter	Description	Example
	tunnel.message	The type of the data to be written to the tunnel. This parameter takes effect only when the tunnel parameter is set to kafka or file. Valid values: - raw: writes data in the original format. The data is aggregated in batches to be written or read at a time. - json: writes data to Kafka in the JSON format so that the data can be directly read. - bson: writes data to Kafka in the Binary JSON (BSON) format. Note The default value is raw.	tunnel.message = raw
	mongo_connect_mode	The type of the node from which MongoShake pulls data. This parameter takes effect only when the tunnel parameter is set to direct. Valid values: - primary: pulls data from the primary node. - secondaryPreferred: pulls data from a secondary node. - standalone: pulls data from the single node that is specified. Note The default value is secondaryPreferred.	mongo_connect_mode = secondaryPreferred

Section	Parameter	Description	Example
	filter.namespace.black	The namespace blacklist for data synchronization. The specified namespaces are not synchronized to the destination database. Separate multiple namespaces with semicolons (;). Note A namespace is the standard name of a collection or index in ApsaraDB for MongoDB. It is the combination of a database name and a collection or index name. Example: `mongodbtest.customer` .	`filter.namespace.black = mongodbtest.customer;testdata.test123`
	filter.namespace.white	The whitelist for data synchronization. Only the specified namespaces are synchronized to the destination database. Separate multiple namespaces with semicolons (;).	`filter.namespace.white = mongodbtest.customer;test123`
	filter.pass.special.db	The special database from which you want to synchronize data to the destination database. You can specify multiple special databases. By default, the data in special databases such as admin, local, mongoshake, config, and system.views is not synchronized. You can set this parameter to synchronize data from special databases. Separate multiple database names with semicolons (;).	`filter.pass.special.db = admin;mongoshake`
	filter.ddl_enable	Specifies whether to synchronize DDL operations. Valid values: • true • false Note If the source ApsaraDB for MongoDB instance is a sharded cluster instance, you cannot set this parameter to true.	`filter.ddl_enable = false`

Section	Parameter	Description	Example
	checkpoint.storage.url	The storage location of checkpoints, which are used for resumable upload. If you do not set this parameter, MongoShake writes checkpoints to the following databases based on the type of the source ApsaraDB for MongoDB instance: - Replica set instance: MongoShake writes checkpoints to the mongoshake database. - Sharded cluster instance: MongoShake writes checkpoints to the admin database on the Configserver node.	`checkpoint.storage.url = mongodb://root:Ftxxxxxx@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717`
	checkpoint.storage.db	The name of the database that stores checkpoints. **Note** The default value is mongoshake.	`checkpoint.storage.db = mongoshake`
	checkpoint.storage.collection	The name of the collection that stores checkpoints. If you use the active MongoShake node and standby MongoShake node to synchronize data from the same database, you can change this collection name to avoid the conflict caused by duplicate collection names. **Note** The default value is ckpt_default.	`checkpoint.storage.collection = ckpt_default`
	checkpoint.start_position	The start position for resumable upload. If a checkpoint exists, this parameter is invalid. Set the value in the following format: `YYYY-MM-DDTHH:MM:SSZ`. **Note** The default value is 1970-01-01T00:00:00Z.	`checkpoint.start_position = 1970-01-01T00:00:00Z`

Section	Parameter	Description	Example
	transform.namespace	The rule for renaming the source database or collection in the destination database. For example, you change the database name and collection name from `Database A.Collection B` to `Database C.Collection D` in the destination database.	`transform.namespace = fromA.fromB:toC.toD`
Full data synchronization options	full_sync.reader.collection_parallel	The maximum number of collections that can be concurrently pulled by MongoShake at a time.	`full_sync.reader.collection_parallel = 6`
	full_sync.reader.write_document_parallel	The number of concurrent threads used by MongoShake to write a collection.	`full_sync.reader.write_document_parallel = 8`
	full_sync.reader.document_batch_size	The number of documents to be written to the destination ApsaraDB for MongoDB instance at a time. For example, a value of 128 indicates that 128 documents are written to the destination ApsaraDB for MongoDB instance at a time.	`full_sync.reader.document_batch_size = 128`
	full_sync.collection_exist_drop	Specifies whether to delete the collections in the destination database that have the same names as the source collections before synchronization. Valid values: - true: deletes the collections in the destination database that have the same names as the source collections before synchronization. **Warning** This option deletes collections in the destination database. Therefore, back up data in the destination database in advance. - false: returns an error message and exits if a collection in the destination database has the same name as a source collection.	`full_sync.collection_exist_drop = true`

Section	Parameter	Description	Example
	full_sync.create_index	Specifies whether to create indexes after the synchronization is complete. Valid values: - foreground: Indexes are created in the foreground. - background: Indexes are created in the background. - none: No indexes are created.	`full_sync.create_index = none`
	full_sync.executor.insert_on_dup_update	Specifies whether to change an `INSERT` statement to an `UPDATE` statement if a document in the destination database has the same `_id` value as the source document. Valid values: - true - false	`full_sync.executor.insert_on_dup_update = false`
	full_sync.executor.filter.orphan_document	Specifies whether to filter out orphaned documents if the source ApsaraDB for MongoDB instance is a sharded cluster instance. Valid values: - true - false	`full_sync.executor.filter.orphan_document = false`
	full_sync.executor.majority_enable	Specifies whether to enable the majority write feature in the destination ApsaraDB for MongoDB instance. Valid values: - true - false	`full_sync.executor.majority_enable = false`
	incr_sync.mongo_fetch_method	The method used to pull incremental data. Valid values: - oplog: pulls oplogs from the source database. - change_stream: pulls change events from the source database. Only MongoDB 4.0 or later supports this method. The default value is oplog.	`incr_sync.mongo_fetch_method = oplog`

Section	Parameter	Description	Example
Incremental data synchronization options	incr_sync.oplog.gids	The global ID used to implement two-way replication for ApsaraDB for MongoDB clusters. You can apply for global IDs by submitting a ticket.	`incr_sync.oplog.gids = xxxxxxxxxxxx`
	incr_sync.shard_key	The method used to distribute concurrent requests to internal worker threads. Do not modify this parameter value.	`incr_sync.shard_key = collection`
	incr_sync.worker	The number of concurrent threads that transmit oplogs. If the performance of your ECS instance is sufficient, you can increase the number of concurrent threads. **Note** If the source ApsaraDB for MongoDB instance is a sharded cluster instance, the number of concurrent threads must be equal to the number of shard nodes.	`incr_sync.worker = 8`
	incr_sync.worker.oplog_compressor	Specifies whether to decompress data to reduce network bandwidth usage. Valid values: - none: No data is compressed. - gzip: Data is compressed in the GZIP format. - zlib: Data is compressed in the ZLIB format. - deflate: Data is compressed in the DEFLATE format. **Note** This parameter takes effect only when the tunnel parameter is not set to `direct`. If the tunnel parameter is set to `direct`, set the value to `none`.	`incr_sync.worker.oplog_compressor = none`

Section	Parameter	Description	Example
	incr_sync.target_delay	The time delayed for synchronizing data between the source and destination ApsaraDB for MongoDB instances. By default, changes in the source database are synchronized to the destination database in real time. To avoid invalid operations, you can set this parameter to delay the synchronization. For example, if you set `incr_sync.target_delay` to 1800, the synchronization is delayed for 30 minutes. Unit: seconds. ⓘ **Note** A value of 0 indicates that data is synchronized in real time.	`incr_sync.target_delay = 1800`
	incr_sync.worker.batch_queue_size	The parameters for configuring internal queues in MongoShake. Do not modify the settings of these parameters unless otherwise required.	`incr_sync.worker.batch_queue_size = 64`
	incr_sync.adaptive.batching_max_size		`incr_sync.adaptive.batching_max_size = 1024`
	incr_sync.fetcher.buffer_capacity		`incr_sync.fetcher.buffer_capacity = 256`
	incr_sync.executor.upsert	Specifies whether to change an `UPDATE` statement to an `INSERT` statement if no document in the destination database has the same `_id` value or unique index as the source document. Valid values: - true - false	`incr_sync.executor.upsert = false`

Section	Parameter	Description	Example
Direct tunnel options (This section takes effect only when the tunnel parameter is set to `direct`.)	incr_sync.executor.insert_on_dup_update	Specifies whether to change an `INSERT` statement to an `UPDATE` statement if a document in the destination database has the same `_id` value or unique index as the source document. Valid values: - true - false	`incr_sync.executor.insert_on_dup_update = false`
	incr_sync.conflict_write_to	Specifies whether to record conflicting documents if data write conflicts occur during the synchronization. Valid values: - none: Conflict documents are not recorded. - db: Conflict logs are written to the mongoshake_conflict database. - sdk: Conflict logs are written to an SDK.	`incr_sync.conflict_write_to = none`
	incr_sync.executor.majority_enable	Specifies whether to enable the majority write feature in the destination ApsaraDB for MongoDB instance. Valid values: - true - false **Note** The majority write feature may compromise the performance.	`incr_sync.executor.majority_enable = false`

FAQ

For frequently asked questions about MongoShake, visit FAQ.

14.6.2. Use MongoShake to implement delayed synchronization among ApsaraDB for MongoDB instances

This topic describes how to use MongoShake to synchronize data among ApsaraDB for MongoDB instances after a buffer period.

Prerequisites

The MongoShake version must be 2.4.6 or later. For more information, visit the MongoShake releases page.

Context

If you use MongoShake to synchronize data among multiple instances in real time, MongoShake synchronizes misoperations on the primary instance to the secondary instances. In this case, you must restore data of the secondary instances to undo the misoperations. To resolve this issue, a parameter is added to MongoShake version 2.4.6 to allow delayed synchronization. This feature provides a buffer period for data synchronization among ApsaraDB for MongoDB instances. If a misoperation is performed on the primary instance, you can disable synchronization during the buffer period and migrate your workloads to a secondary instance to which the incorrect data is not synchronized.

> **Note** This topic describes how to use the `incr_sync.target_delay` parameter. For more information about how to use MongoShake, see Use MongoShake to implement one-way synchronization between ApsaraDB for MongoDB replica set instances.

Preparations

1. For best synchronization performance, make sure that the source ApsaraDB for MongoDB replica set instance resides in a VPC. If the source instance resides in the classic network, switch the network type to VPC. For more information, see Switch the network type of an ApsaraDB for MongoDB instance.
2. Create an ApsaraDB for MongoDB replica set instance as the synchronization destination. Select the same VPC as the one used by the source ApsaraDB for MongoDB replica set instance to minimize network latency. For more information, see 创建副本集实例.
3. Create an ECS instance to run MongoShake. Select the same VPC as the one used by the source ApsaraDB for MongoDB instance to minimize network latency. For more information, see Create an ECS instance.
4. Add the private IP address of the ECS instance to the whitelists of the source and destination ApsaraDB for MongoDB instances. Make sure that the ECS instance can connect to the source and destination ApsaraDB for MongoDB instances. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

> **Note** If the network type does not meet the preceding requirements, you can apply for public endpoints for the source and destination ApsaraDB for MongoDB instances. Then, add the public IP address of the ECS instance to the whitelists of the source and destination ApsaraDB for MongoDB instances. This way, you can synchronize data by using the Internet. For more information, see Apply for a public endpoint for an ApsaraDB for MongoDB instance and Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

Configure delayed synchronization among ApsaraDB for MongoDB instances

The following example uses an Ubuntu Elastic Compute Service (ECS) instance to describe how to configure delayed synchronization among ApsaraDB for MongoDB instances. For more information, see .

1. Log on to the ECS instance.
2. Run the following command to download the MongoShake program:

```
wget <Download address of the latest MongoShake package>
```

Example:

```
wget https://github.com/alibaba/MongoShake/releases/download/release-v2.0.7-20190817/mongo-shake-2.0.7.tar.gz
```

> Note The download address of the latest MongoShake package is listed on the MongoShake releases page.

3. Run the following command to decompress the MongoShake package:

```
tar xvf <Name of the MongoShake package>
```

Example:

```
tar xvf mongoshake-2.0.tar.gz
```

4. Run the `vi collector.conf` command to configure MongoShake. For information about the configuration parameters, see MongoShake parameters. Find the `incr_sync.target_delay` parameter in the collector.conf file and set this parameter based on your needs. The unit is seconds. In this example, the buffer period is set to 1,800 seconds.

```
incr_sync.target_delay = 1800
```

5. Save and close the collector.conf file to complete the configuration.
6. Run the following command to start synchronization based on the collector.conf file and print the logs.

```
./collector.linux -conf=collector.conf -verbose
```

> Note MongoShake synchronizes changes in the primary instance to secondary instances 30 minutes after the changes.

Perform a switchover between the primary instance and a secondary instance after a misoperation

When you perform create, read, update, and delete (CURD) operations on the primary instance, a misoperation may occur. For example, a statement is executed to write unwanted data to the primary instance. To respond to a misoperation, you can follow these steps to migrate your workloads to a secondary instance to which the incorrect data is not synchronized.

1. Query the operation logs of the primary ApsaraDB for MongoDB instance to identify the time when the misoperation occurred. For example, you can run the following command to query all operation logs that were generated from June 1, 2020 to June 2, 2020. For more information about how to query operation logs, see MongoDB official documentation.

```
use local#Switch to the local database.
db.oplog.rs.find({"o.createTime": {$gte:new Date(2020,6,1),$lte:new Date(2020,6,2)}}) #Query operation logs based on the specified conditions.
```

2. Run the following Restful API command to inject the ExitPoint parameter to MongoShake to terminate the MongoShake program at a specified time.

   ```
   curl -X POST --data '{"ExitPoint": <UNIX timestamp>}' <MongoShake server ID>:<Port number>/sentinel/options
   ```

   Example:

   ```
   curl -X POST --data '{"ExitPoint": 1593534600}' 127.0.0.1:9100/sentinel/options
   ```

   > Note The string `1593534600` is a UNIX timestamp that indicates 16:30:00 June 30, 2020. MongoShake automatically exits at the specified time.

3. Run the `vi collector.conf` command to open the configuration file and exchange the IP addresses of the primary instance and the secondary instance. For more information, see Use MongoShake to implement one-way synchronization between ApsaraDB for MongoDB replica set instances.
4. Run the following command to restart synchronization based on the collector.conf file and print the logs.

   ```
   ./collector.linux -conf=collector.conf -verbose
   ```

5. Migrate your workloads to the new primary instance to complete the switchover operation.

Monitor the MongoShake status

For more information, see Monitor the MongoShake status.

15.Data backup

15.1. 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 you specify.

Usage notes

- If the database version of an instance is 3.2 or 3.4, the number of collections and indexes in the instance cannot exceed 10,000. Otherwise, physical backup may fail. If you want to increase this limit, we recommend that you upgrade MongoDB versions to 4.0 or later. Alternatively, you can select the database version 4.0 or 4.2 when you create the instance. For more information, see Upgrade MongoDB versions.
- After the database version is upgraded, the backup files of the original version cannot be used to restore data of the new version.

Automatic backup

- ApsaraDB for MongoDB stores backup files in Object Storage Service (OSS) to reduce the storage usage of ApsaraDB for MongoDB instances.
- Standalone instances can use only snapshot backup, which affects their I/O performance in the backup process.

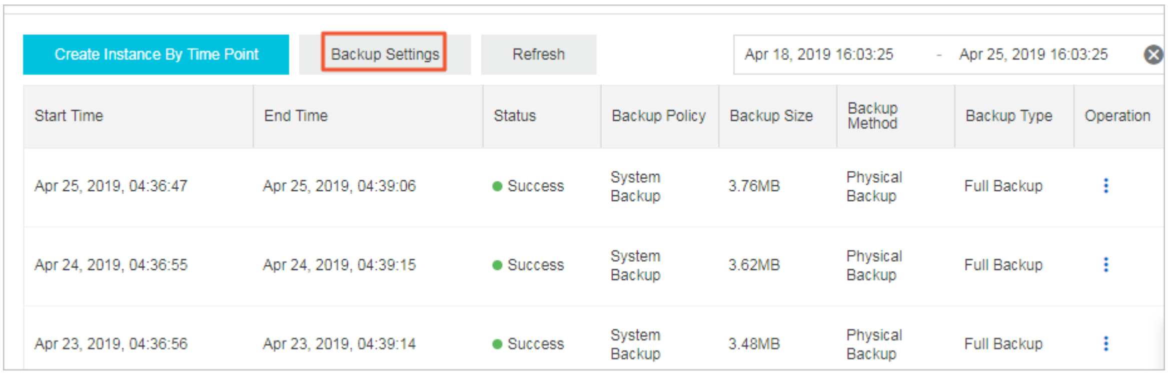
  > ? **Note** Snapshot backup retains the status of disk data at a specific point in time.

- Replica set and sharded cluster instances support physical backup.

  > ? **Note** With physical backup, all physical database files in an ApsaraDB for MongoDB instance are backed up. Physical backup runs on the hidden node of an ApsaraDB for MongoDB instance, and does not affect the I/O performance of the primary and secondary nodes. If the data volume is large, backing up your ApsaraDB for MongoDB instance may require a long time.

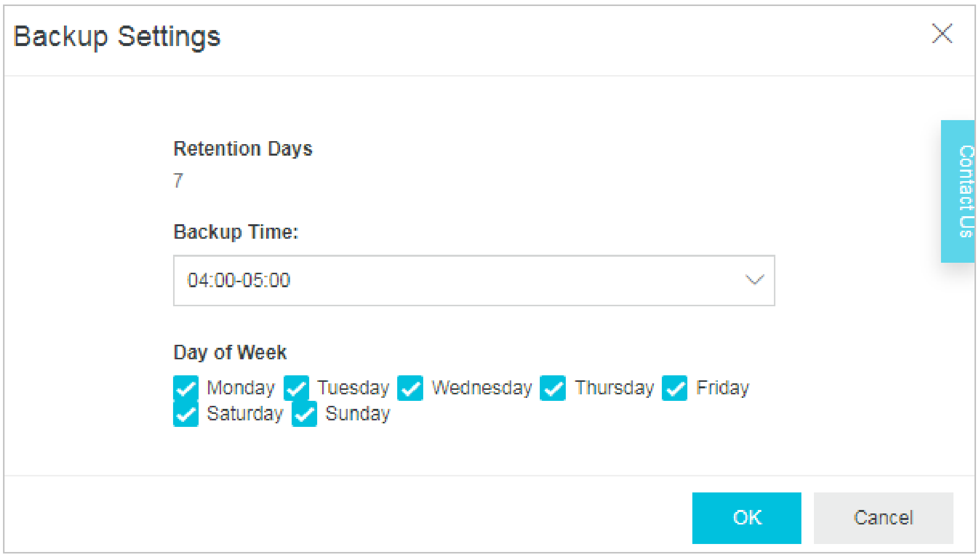
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. Click **Backup Settings**.


Create Instance By Time Point
Backup Settings
Refresh
Apr 18, 2019 16:03:25 - Apr 25, 2019 16:03:25

Start Time	End Time	Status	Backup Policy	Backup Size	Backup Method	Backup Type	Operation
Apr 25, 2019, 04:36:47	Apr 25, 2019, 04:39:06	● Success	System Backup	3.76MB	Physical Backup	Full Backup	⋮
Apr 24, 2019, 04:36:55	Apr 24, 2019, 04:39:15	● Success	System Backup	3.62MB	Physical Backup	Full Backup	⋮
Apr 23, 2019, 04:36:56	Apr 23, 2019, 04:39:14	● Success	System Backup	3.48MB	Physical Backup	Full Backup	⋮

7. In the dialog box that appears, configure the following parameters.



Retention Days
7

Backup Time:
04:00-05:00

Day of Week
☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday
☑ Saturday ☑ Sunday

Parameter	Description
Retention Days	The number of days for which you want to retain backup data. It can only be seven days.
Backup Time	The hour at which you want to perform the backup task. We recommend that you select an off-peak hour.
Day of Week	The backup cycle. You can select one or more days in a week.

8. Click **OK**.

References

Restoration solution overview

15.2. 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.

Impacts

- ApsaraDB for MongoDB stores its backup files in Object Storage Service (OSS) to reduce the storage usage of ApsaraDB for MongoDB instances.
- Standalone instances support only snapshot backup, which decreases their I/O performance.
- Physical backup and logical backup run on the hidden nodes of replica set instances, and do not affect the performance of the primary and secondary nodes. If the data volume is large, backing up your ApsaraDB for MongoDB instance may require a long time.
- After the database version is upgraded, the backup files of the original version cannot be used to restore data of the new version.

Backup methods

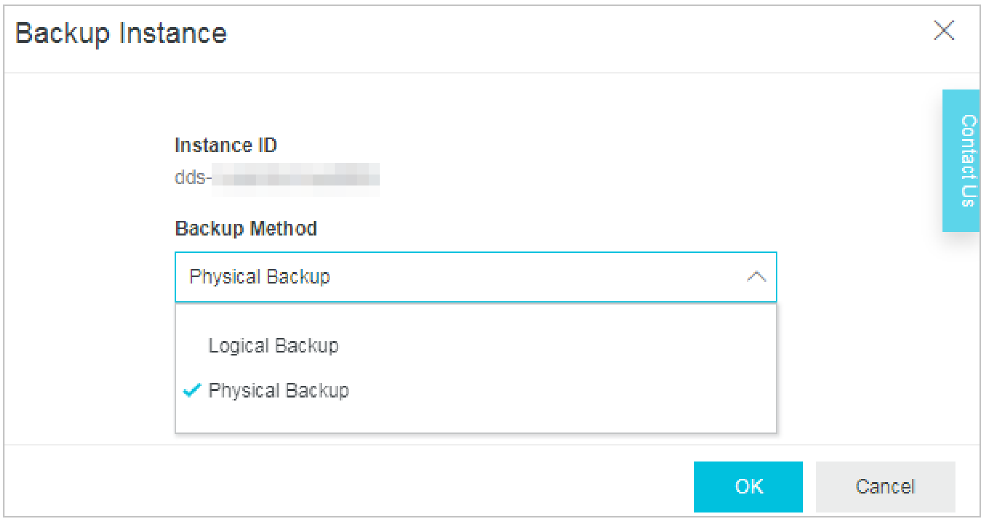
- Snapshot backup: The status of disk data at a specific point in time is retained.
- Physical backup: Physical database files of an ApsaraDB for MongoDB instance are backed up. This method provides faster backup and restoration compared with logical backup.
- Logical backup: mongodump is used to logically back up each database. Logical backup restores data in the form of playback commands during restoration.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the upper-right corner of the page, click **Backup Instance**.
6. In the dialog box that appears, specify **Backup Method**.

   Note

   - If the instance is a standalone instance, you can select only **Snapshot Backup**.
   - If the instance is a replica set or sharded cluster instance, you can select **Logical Backup** or **Physical Backup**.



Backup Instance

Instance ID

dds-[redacted]

Backup Method

Physical Backup

7. Click **OK**.

References

Restoration solution overview

16.Data recovery

16.1. Restoration solution overview

The data restoration feature of ApsaraDB for MongoDB can minimize any losses caused by incorrect operations on databases. ApsaraDB for MongoDB provides many data restoration solutions to meet requirements in different scenarios.

Restore data to ApsaraDB for MongoDB instances

Method	Instance type	Scenario	Remarks
Create an instance from a backup	Standalone or replica set instance	Applicable to the scenarios where the entire instance is restored and data timeliness is not a key requirement.	A new instance is created based on the backup data and data is restored to the new instance. Note - The created instance will be billed. For more information, see Billing items and pricing. - To ensures better performance and stability of the instance, the system will upgrade the minor version to the latest version by default If the minor version of your instance expires or is not included in the maintenance list and the instance is upgraded, migrated, changed, Created from a backup, Created by point-in-time, or performed Restore data to a new ApsaraDB for MongoDB instance.
Create an instance based on a point in time	Replica set or sharded cluster instance	Applicable to the scenarios where multiple databases or the entire instance is restored. The data at a specified point in time will be restored.	
Restore data to a new ApsaraDB for MongoDB instance	Replica set instance	Applicable to the scenarios where one or more databases are quickly restored. For example, a data set or document is deleted by mistake.	
Restore backup data to the current instance	Replica set instance (three-node)	N/A	Restoring data directly to the current instance poses high risks. We recommend that you restore the data by using the following feature: Restore data to a new ApsaraDB for MongoDB instance by point in time or Create an instance from a backup. After the data is validated, migrate the data back to the original instance through DTS.

Restore data to user-created databases

You can download backup files of ApsaraDB for MongoDB to your server and recover the data to a user-created database. This feature is applicable to scenarios such as business testing or data analysis.

Method	Instance type
Restore logical backup files of ApsaraDB for MongoDB to user-created databases	Replica set instance
Restore physical backup files of ApsaraDB for MongoDB to user-created databases	Replica set instance

16.2. Restore data to a new ApsaraDB for MongoDB instance

This topic describes how to restore one or more databases of an ApsaraDB for MongoDB instance to a new ApsaraDB for MongoDB instance by using a backup created at a specific point in time. This method is ideal for quick data restoration.

Background information

Instances created after March 26, 2019 support the restoration of one or more databases. For information about when this feature will be available to instances created before March 26, 2019, follow the official website.

Prerequisites

- The instance is created after March 26, 2019.
- The instance is located in the China (Qingdao), China (Beijing), China (Zhangjiakou), China (Hohhot), China (Hangzhou), China (Shanghai), China (Shenzhen), or Singapore (Singapore) region.
- The instance is a replica set instance.
- The MongoDB version of the instance is 3.4, 4.0, or 4.2.

> Note
>
> - If the MongoDB version of the instance is earlier than required versions, you must upgrade the database version. For more information, see Upgrade MongoDB versions.
> - After the MongoDB version is upgraded, the backup files of the original version cannot be used to restore data of the new version.

- The storage engine of the instance is WiredTiger.
- The backup file list of the instance contains the backup files of the databases you want to restore.

Precautions

- You can restore databases only from physical backups.
- The time required varies based on factors such as the data volume, task queue status, and network conditions. When the status of the new instance changes to **Running**, the restoration is complete.

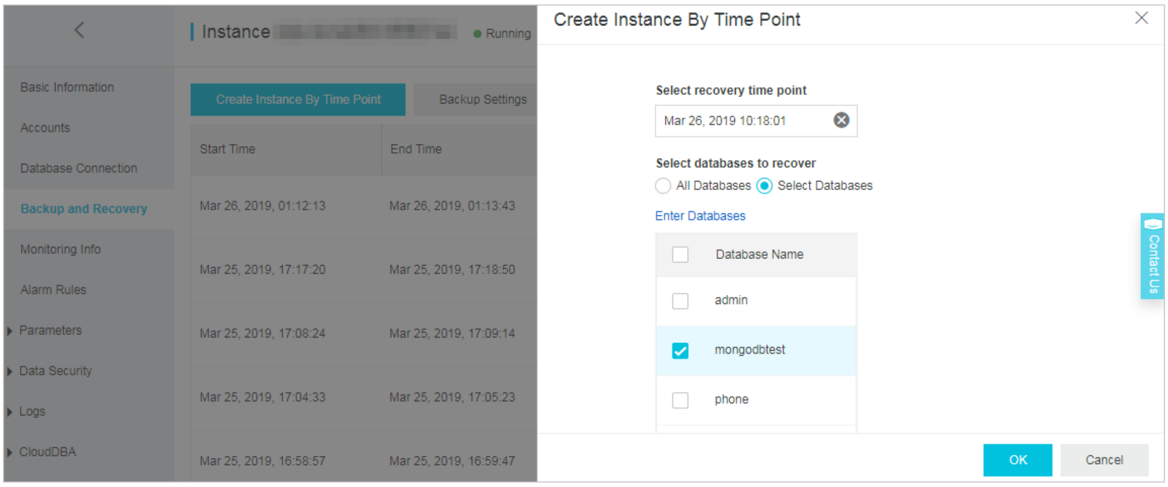
Billing

When you restore one or more databases, the system creates an instance and you are charged for the new instance. For more information, see Billing items and pricing.

> ? **Note** If you want to restore the databases to a pay-as-you-go instance, make sure that your account has sufficient balance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. On the **Backup and Recovery** page, click **Create Instance By Time Point**.
7. In the Create Instance By Time Point panel, configure the following parameters.



Start Time	End Time
Mar 26, 2019, 01:12:13	Mar 26, 2019, 01:13:43
Mar 25, 2019, 17:17:20	Mar 25, 2019, 17:18:50
Mar 25, 2019, 17:08:24	Mar 25, 2019, 17:09:14
Mar 25, 2019, 17:04:33	Mar 25, 2019, 17:05:23
Mar 25, 2019, 16:58:57	Mar 25, 2019, 16:59:47

Create Instance By Time Point

Select recovery time point: Mar 26, 2019 10:18:01

Select databases to recover: ○ All Databases ● Select Databases

Enter Databases

	Database Name
□	admin
☑	mongodbtest
□	phone

Parameter	Description
Select recovery time point	Select a point in time from which you want to restore data. You can select a time point from the last seven days. ? **Note** The time you select must be earlier than the current time and later than the time when the source instance was created.

Parameter	Description
Select databases to recover	○ **All Databases**: If you select this option, all databases in the source instance are restored. ○ **Select Databases**: If you select this option, only selected databases are restored. You can directly select the databases you want to restore, or click **Enter Databases** to enter the names of the databases. ⓘ Note If you want to restore more than one database, separate the database names with commas (,) when you enter them.

8. Click **OK**.
9. On the instance buy page, select a billing method for the new instance.

   ⓘ Note

   - Subscription: You must pay for the subscription when you create an instance. This method is more cost-effective than the pay-as-you-go method. We recommend that you select this method for long-term use. A longer subscription period enables a larger discount.
   - Pay-as-you-go: You are billed on an hourly basis based on the used resources. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance if it is no longer needed.

10. Configure the new instance. For more information, see 创建副本集实例.

   ⓘ Note

   - You cannot change **Region**, **Database Version**, **Storage Engine**, or **Replication Factor** for the new instance.
   - To make sure that the new instance has sufficient space for restoration, we recommend that you set the storage capacity larger than or equal to that of the source instance.

11. Click **Buy Now**.
12. On the **Confirm Order** page, read and select **ApsaraDB for MongoDB Agreement of Service**, and complete the payment.

16.3. Create an instance from a backup

This topic describes how to create a new ApsaraDB for MongoDB instance from a backup. You can use this method to restore data or verify data in instance backups.

Prerequisites

- The source instance is a standalone instance or a replica set instance.
- The selected backup was created in the last seven days.

Precaution

To ensures better performance and stability of the instance, the system will upgrade the minor version to the latest version by default If the minor version of your instance expires or is not included in the maintenance list and the instance is upgraded, migrated, changed, Created from a backup, Created by point-in-time, or performed Restore data to a new ApsaraDB for MongoDB instance.

Billing

If you create an instance from a backup, you are charged for the new instance. For more information, see Billing items and pricing.

> **Note** To create a pay-as-you-go instance, make sure that your account has sufficient balance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. Find the target backup file. On the right side of the backup file, choose

   ⋮

   > Create Instance from Backup Point.

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

7. In the dialog box that appears, configure the following parameters:
   - **All Databases**: restores data of all databases in the source instance to the target instance.
   - **Select Databases**: restores data of some databases in the source instance to the target instance.

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

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

8. On the Instance Purchase page, select a billing method for the new instance.

> ? Note
>
> - Subscription: You must pay for an instance when you create it. This method is more cost-effective than the pay-as-you-go method. We recommend that you select this method for long-term use. A longer subscription period enables a larger discount.
> - Pay-as-you-go: You are billed on an hourly basis based on the used resources. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance if it is no longer needed.

9. Configure the new instance. For more information, see 创建副本集实例.

> ? Note
>
> - You cannot modify **Region**, **Database Version**, **Storage Engine**, and **Replication Factor** for the new instance.
> - We recommend that you select a storage capacity that is no smaller than the storage capacity of the source instance. This ensures that the new instance has sufficient space to store the data restored from the source instance.

10. Click **Buy Now**.
11. Read and select **ApsaraDB for MongoDB Agreement of Service**, and pay for the order.

> ? Note The time required to restore data to a new instance from a backup varies depending on factors such as the data volume, the task priority in the task queue, and network conditions. When the new instance enters the **Running** state, the restoration is complete.

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

This topic describes how to restore data to a new ApsaraDB for MongoDB instance by point in time. This method is ideal for data restoration and verification.

Prerequisites

- The source instance is a replica set or sharded cluster instance.
- You can select a point in time only from the last seven days.

Billing

This method creates an instance and you are charged for the new instance. For more information, see Billing items and pricing.

> ? Note If you want to restore data to a pay-as-you-go instance, make sure that your account has sufficient balance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**, or **Sharded Cluster Instances** based on the instance type.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. On the **Backup and Recovery** page, click **Create Instance By Time Point**.
7. In the **Create Instance By Time Point** panel, select a time point for restoration and select one of the following options:
   - **All Databases**: restores data of all databases in the source instance to the destination instance.
   - **Select Databases**: restores data of some databases in the source instance to the destination instance.

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

     > Note
     >
     > - If you enter the names of the databases, separate the database names with commas (,).
     > - To ensure data integrity and accuracy, do not select the latest point in time (usually the latest hour) if the instance is a sharded cluster instance. Otherwise, restoration fails.
     > - After the database version is upgraded, the backup files of the original version cannot be used to restore data of the new version.

8. Click **OK**.
9. On the instance buy page, select a billing method for the new instance.

   > Note
   >
   > - Subscription: You must pay for the subscription when you create an instance. This method is more cost-effective than the pay-as-you-go method. We recommend that you select this method for long-term use. A longer subscription period enables a larger discount.
   > - Pay-as-you-go: You are billed on an hourly basis based on the used resources. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance after you no longer need it.

10. Configure the new instance. For more information, see 创建副本集实例 or 创建分片集群实例.

Note

- Replica set instance: The storage capacity of the new instance must be larger than or equal to that of the source instance.
- Sharded cluster instance:
  - The number of shard nodes in the new instance must be larger than or equal to that in the source instance.
  - The storage capacity of each shard node in the new instance must be larger than or equal to those in the source instance.

11. Click **Buy Now**.
12. Read and select **ApsaraDB for MongoDB Agreement of Service** and complete the payment.

Note The time required to restore data to a new instance by backup set varies based on factors such as the data volume, task queue status, and network conditions. When the status of the new instance changes to **Running**, the restoration is complete.

16.5. 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.

Prerequisite

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.

Warning This operation is risky. We recommend that you restore data to a new ApsaraDB for MongoDB instance by point in time or backup set. Then, verify the data, and migrate the data back to the source instance by using Data Transmission Service (DTS). For more information, see Restore data to a new ApsaraDB for MongoDB instance by point in time or Create an instance from a backup.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.

4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. On the **Backup and Recovery** page, find the backup set and choose

   ⋮

   > **Data Rollback** 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.

7. 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**.

16.6. Restore data of an ApsaraDB for MongoDB instance to self-managed MongoDB databases by using logical backup

This topic describes how to restore the data of an ApsaraDB for MongoDB instance to self-managed MongoDB databases by using logical backup. Data restoration uses the mongorestore command. You must have created a logical backup and downloaded the logical backup file to the server where you plan to run the mongorestore command.

Prerequisites

- A replica set instance with three or more nodes is created.
- The ApsaraDB for MongoDB instance and the self-managed MongoDB databases run the same database version.

Context

Full logical backup uses the mongodump command to back up a database. During the backup process, you can still perform read/write operations on the database.

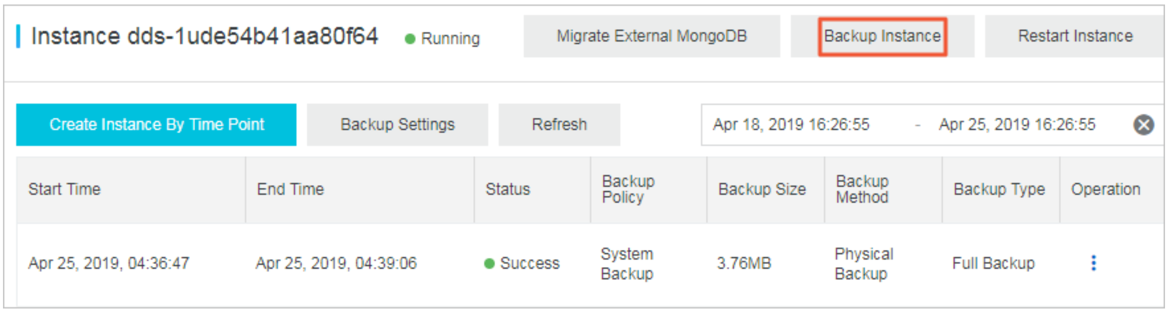
> **Note** Full logical backup runs on the hidden node of the ApsaraDB for MongoDB instance. This does not affect the read/write performance of the primary and secondary nodes. It may take a long time to back up a large volume of data.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target

instance.

3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the upper-right corner of the page, click **Backup Instance**.


Instance dds-1ude54b41aa80f64
Running
Migrate External MongoDB
Backup Instance
Restart Instance
Create Instance By Time Point
Backup Settings
Refresh
Apr 18, 2019 16:26:55 - Apr 25, 2019 16:26:55

Start Time	End Time	Status	Backup Policy	Backup Size	Backup Method	Backup Type	Operation
Apr 25, 2019, 04:36:47	Apr 25, 2019, 04:39:06	● Success	System Backup	3.76MB	Physical Backup	Full Backup	⋮

6. In the **Backup Instance** panel, select **Logical Backup** for **Backup Method**.
7. Click **OK**. Then, wait for the backup to complete.
8. On the **Backup and Recovery** page, find the logical backup file and choose

   ⋮

   > Download.
9. Copy the downloaded file to the server where you plan to run the mongorestore command.
10. Run the following command to import the file to self-managed MongoDB databases:

```
mongorestore -h <hostname> --port <server port> -u <username> -p <password> --drop --gzip --archive
=<backupfile> -vvvv --stopOnError
```

Parameter description:

- <hostname>: the address of the server where the self-managed MongoDB databases reside. If you also run the mongorestore command on this server, enter 127.0.0.1.
- <server port>: the port number of the self-managed MongoDB databases.
- <username>: the username you use to log on to the self-managed MongoDB databases.
- <password>: the password of the preceding account.
- <backupfile>: the name of the logical backup file you downloaded.

Example:

```
mongorestore -h 127.0.0.1 --port 27017 -u root -p xxxxxxxx --drop --gzip --archive=hins1111_data_20190
710.ar -vvvv --stopOnError
```

16.7. Recover physical backup data in a user-created MongoDB instance

16.7.1. Download the physical backup data of a replica set instance

You can download the physical backup data of a replica set instance based on the backup time and restore the downloaded data to a self-managed MongoDB database.

Prerequisites

The instance is a replica set instance.

Context

After you set an automatic backup policy, ApsaraDB for MongoDB performs physical backups for the replica set instance on a regular basis.

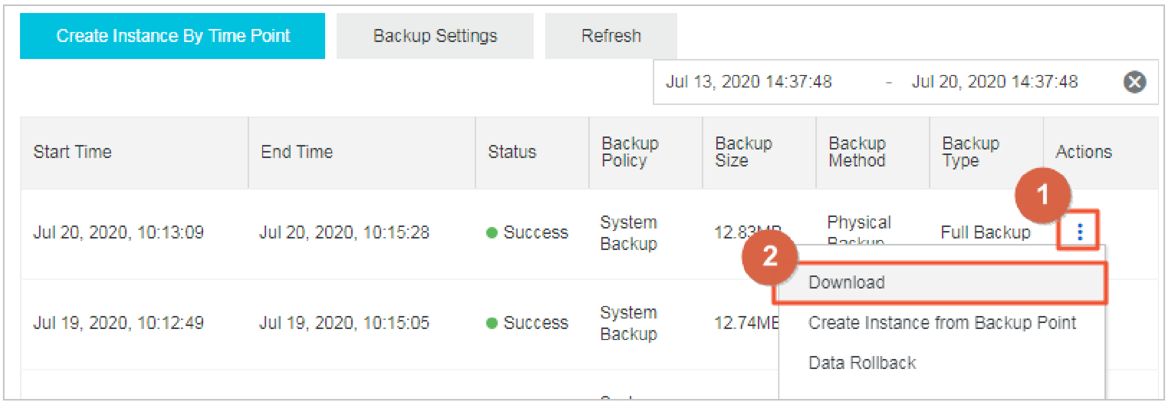
A physical backup is performed on the hidden node of a replica set instance. This does not affect the I/O performance of the primary and secondary nodes. It may take a long time to back up a large volume of data.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the resource group and the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. On the **Backup and Recovery** page, find the physical backup that you want to download. Then, choose

   ⋮

   > **Download**.


Create Instance By Time Point
Backup Settings
Refresh
Jul 13, 2020 14:37:48 - Jul 20, 2020 14:37:48
1
2
Download
Create Instance from Backup Point
Data Rollback

Start Time	End Time	Status	Backup Policy	Backup Size	Backup Method	Backup Type	Actions
Jul 20, 2020, 10:13:09	Jul 20, 2020, 10:15:28	● Success	System Backup	12.83MB	Physical Backup	Full Backup	⋮
Jul 19, 2020, 10:12:49	Jul 19, 2020, 10:15:05	● Success	System Backup	12.74MB			

References

After you download the backup file, you can use it to restore data to your self-managed databases. For more information, see 将MongoDB物理备份文件恢复至自建数据库.

17.Zone-disaster restoration solution

17.1. Create a multi-zone replica set instance

This topic describes how to create a multi-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 nodes of a three-node replica set instance to three different zones in one region. The nodes in these zones exchange data over an internal network. When one of the three zones becomes unavailable due to unexpected events such as a power or network failure, the high-availability (HA) system automatically switches services over to another zone.

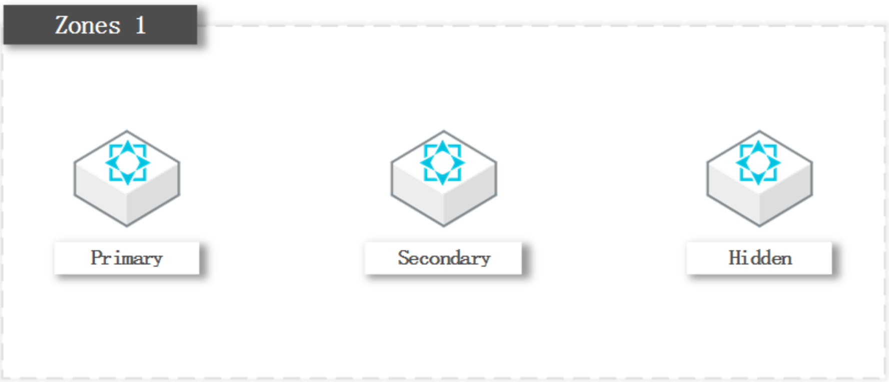
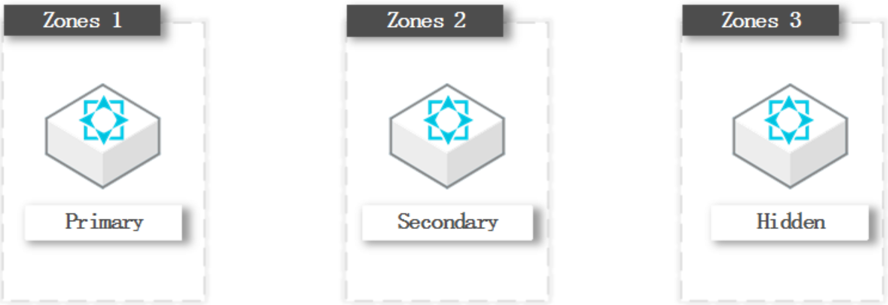
Notes

- You can create multi-zone replica set instances only in some regions. For more information about the supported regions, see the ApsaraDB for MongoDB console.
- When you create a multi-zone replica set instance, you must set **Replication Factor** to **Three Nodes Replica set**.

  > ? **Note** If you need more nodes, you can change the number of nodes after you create the instance. For more information, see Change the number of nodes for a replica set instance.

Node deployment policies

Deployment	Description

Deployment	Description
Single-zone deployment	The system deploys the primary, secondary, and hidden nodes in one zone. 
Multi-zone cluster	The system deploys the primary, secondary, and hidden nodes in three different zones.  Zones 1 Zones 2 Zones 3 Primary Secondary Hidden

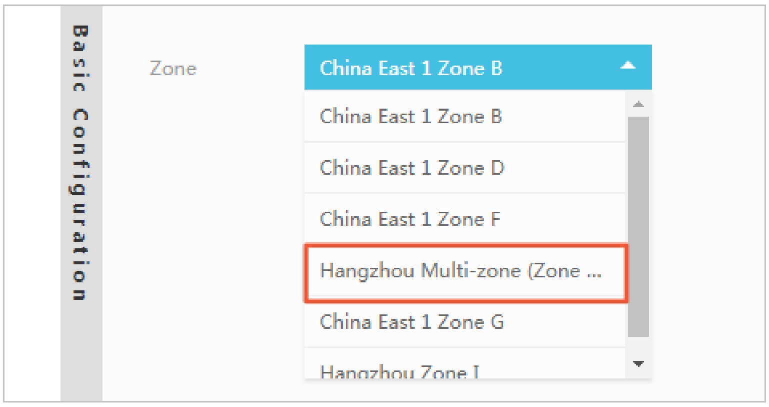
Procedure

1. 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. Click **Replica Set (Subscription)** or **Replica Set (Pay-as-you-go)**.

Note

- Subscription: You must pay for an instance when you create it. This method is more cost-effective than the pay-as-you-go method. We recommend that you select this method for long-term use. A longer subscription period enables a larger discount.
- Pay-as-you-go: You are billed on an hourly basis based on the used resources. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance after you no longer need it.

5. Set **Region** to **China (Hangzhou)**, **China (Beijing)**,**China (Shenzhen)**, or **Singapore (Singapore)**. Then, select a multi-zone configuration from the Zone drop-down list.



6. Configure other parameters. For more information, see 创建副本集实例.
7. Click **Buy Now** to go to the **Confirm Order** page.
8. On the Confirm Order page, read and select ApsaraDB for MongoDB Agreement of Service and complete the payment.

References

You can use the service availability feature to view the distribution of nodes in a replica set instance across zones. You can also switch the node roles of the instance based on your business deployment. This way, your applications can connect to the nodes closest to them. For more information, see Switch node roles.

17.2. Create a multi-zone sharded cluster instance

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

Precautions

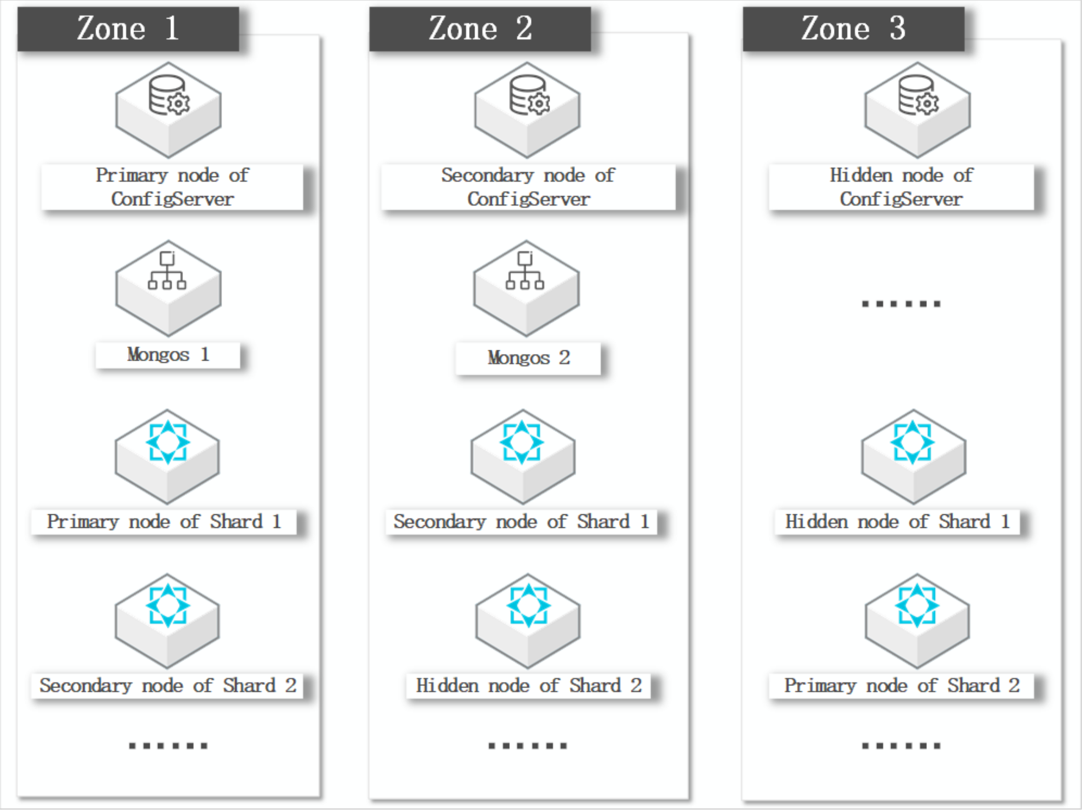
You can create a multi-zone sharded cluster instance only in some regions. For more information about the supported regions, see the ApsaraDB for MongoDB console.

Node deployment policies

If you use the single-zone deployment solution, the system deploys all components of the sharded cluster instance to one zone. If you use the multi-zone deployment solution, the system deploys all components to three different zones.

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

Deployment policy for the components in a multi-zone sharded cluster instance



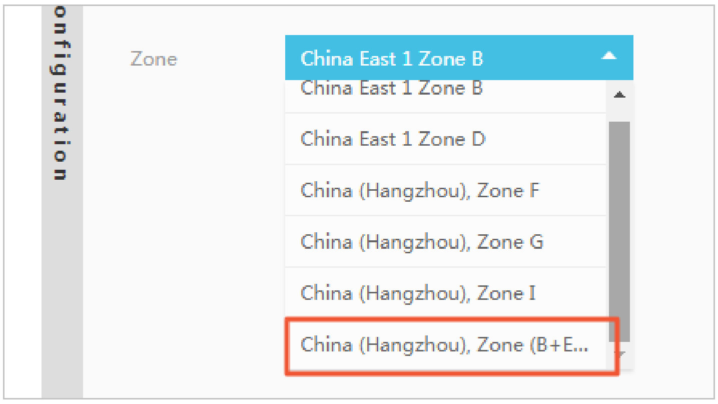
Procedure

1. 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. Click **Sharded Cluster (Subscription)** or **Sharded Cluster (Pay-as-you-go)**.

Note

- Subscription: You must pay for an instance when you create it. This method is more cost-effective than the pay-as-you-go method. We recommend that you select this method for long-term use. A longer subscription period enables a larger discount.
- Pay-as-you-go: You are billed on an hourly basis based on the used resources. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance after you no longer need it.

5. Set **Region** to **China (Hangzhou)**, **China (Beijing)**, **China (Shenzhen)**, or **Singapore (Singapore)** and select a multi-zone.



6. Configure other parameters. For more information, see 创建分片集群实例.
7. Click **Buy Now** to go to the **Confirm Order** page.
8. Read and select Agreement of Service and complete the payment.

References

You can use the Service Availability function to view the distribution of nodes in a replica set instance across zones. You can also switch the node roles of the instance based on your business deployment. This way, your applications can connect to the nodes closest to them. For more information, see Switch node roles.