Alibaba Cloud

ApsaraDB for MongoDB

Quick Start for Replica Set

Document Version: 20201124

⊖ 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.
⑦ Note	A note indicates supplemental instructions, best practices, tips, and other content.	⑦ **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.Notes

You can migrate data from a user-created MongoDB database to an ApsaraDB for MongoDB instance. Pay close attention to the limits of ApsaraDB for MongoDB.

Operation	Limit
Deploy a replica set instance	The database version must match the storage engine. For more information, see 版本及存储引擎.
Build replica set nodes	• A replica set instance automatically created by ApsaraDB for MongoDB consists of a primary node, a hidden secondary node (invisible to you), and one or more secondary nodes. • While a replica set instance is running, you can scale the instance to 3, 5, or 7 nodes as needed. For more information, see Change the number of nodes for a replica set instance. ⑦ Note You cannot connect to a replica set instance from a secondary node of a user-created database. If you want to synchronize data from a replica set instance to a user-created MongoDB database for data testing or analysis, you can use MongoShake. For more information, see Use MongoShake to implement one-way synchronization between ApsaraDB for MongoDB replica set instances.
Restart a replica set instance	You must restart the instance in the ApsaraDB for MongoDB console or by calling the RestartDBInstance operation.
Migrate data from a replica set instance	Migrate the replica set of a user-created MongoDB database to ApsaraDB for MongoDB by using DTS or Migrate user-created MongoDB databases to Alibaba Cloud by using the built-in commands of MongoDB.
Back up the data of a replica set instance	Configure automatic backup for an ApsaraDB for MongoDB instance: Only physical backup is supported. Manually back up an ApsaraDB for MongoDB instance: Both physical backup and logical backup are supported. ⑦ Note If the database version of the 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 the database version to 4.0 or 4.2. For more information, see Upgrade the database version of an ApsaraDB for MongoDB instance. Alternatively, you can select the database version 4.0 or 4.2 when you create the instance.
Restore the data of a replica set instance	• You can restore data from a point in time or a backup set. For more information, see Create an instance from a backup and Restore data to a new ApsaraDB for MongoDB instance by point in time. • You can restore data to your current replica set instance only if it has three nodes. For more information, see Restore data to the current ApsaraDB for MongoDB instance.

Operation	Limit
Modify parameters of a replica set instance	For security and stability, you are not allowed to modify certain parameters of a replica set instance. For more information, see Configure database parameters for an ApsaraDB for MongoDB instance.

2.ApsaraDB for MongoDB console

The ApsaraDB for MongoDB console is a web application for managing instances. In the ApsaraDB for MongoDB console, you can create and manage instances, configure the instance IP whitelists, passwords, and network types, and perform other operations.

The ApsaraDB for MongoDB console is part of the Alibaba Cloud Management Console. For more information about common settings and basic operations in the Alibaba Cloud Management Console, see Alibaba Cloud Management Console.

Prerequisites

Use your Alibaba Cloud account to log on to the ApsaraDB for MongoDB console. To create an Alibaba Cloud account, go to the Alibaba Cloud official website.

ApsaraDB for MongoDB console homepage

The console homepage displays the same information for all replica set instances.

Log on to the ApsaraDB for MongoDB console and go to the **Replica Set Instances** page, as shown in the following figure. This figure is only to be used for reference. The actual page may be different.
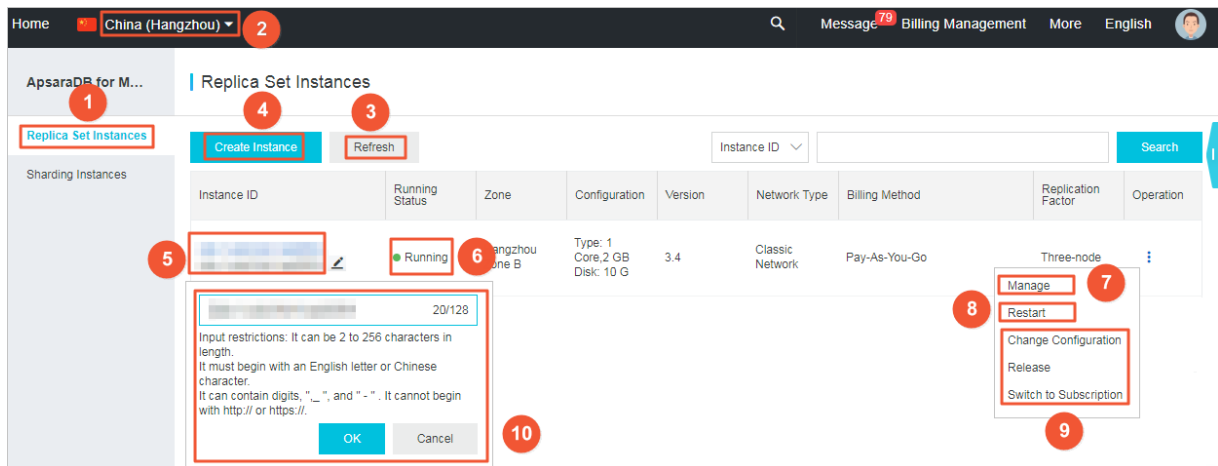
UI element description

No.	UI element	Description
1	Replica Set Instances	The ApsaraDB for MongoDB console homepage, which displays all replica set instances in a region that belong to the current account.
2	Region	You can click a region to display all instances that reside within the region.
3	Refresh	The button to refresh the instance information page.
4	Create Instance	The button to create an instance.
5	Instance ID	You can click an instance ID to go to the Basic Information page of the instance.

No.	UI element	Description
6	Status	The status of the instance. Instances may be in different states.
7	Manage	You can click this button to go to the Basic Information page of the instance. On this page, you can view the basic information about the instance, configure instance backup and recovery, view monitoring information, set alert rules, and configure a whitelist.
8	Restart	The button to restart an instance.
9	More	Other buttons, such as Change Configuration and Renew.
10	Edit icon	You can click this icon to modify the instance name. By default, the instance name is identical to the instance ID.

ApsaraDB for MongoDB instance console

Log on to the ApsaraDB for MongoDB console. Click an **Instance ID** or **Manage** in the Actions column corresponding to an instance. The **Basic Information** page is displayed. The following table lists the UI elements on the page.

UI element or page	Section	Description	Operation
Top navigation bar	N/A	You can migrate, back up, and restart the instance.	• Migrate the data of an instance • Back up an instance • Restart an instance
Basic Information	Basic Information	You can view the basic information about the instance, such as the instance ID, region, network type, specifications, and disk space. You can also change the configurations of the instance.	Change the configurations of an instance
	Accounts	You can view account information and reset passwords.	Reset a password
	Connection Info	You can view the endpoints and port numbers of the instance.	N/A
	Primary Instance Resources	You can view disk usage, IOPS usage, connections and CPU utilization of the instance.	N/A
	Instance Relationship	You can view the relationship between instance nodes.	N/A

UI element or page	Section	Description	Operation
Backup and Recovery	Backups	You can view and download a list of data backups for a specified time period, restore data from the specified time period, or create an instance from a specified backup point.	• Download backup data • Restore data to an instance
	Backup Settings	You can set a backup policy to automatically and periodically back up data based on the specified backup time.	Configure automatic backup for an instance
Monitoring Info	Resource monitoring	You can view the monitoring information of the primary and secondary nodes based on the specified metrics and time range.	N/A
Alert Rules	Set Alert Rule	You can set alert rules.	Set alert rules
Data Security	Whitelist Settings	You can configure an IP whitelist.	Configure an IP whitelist

3.Get started with a replica set instance

This topic describes how to get started with a replica set instance of ApsaraDB for MongoDB. In this topic, you can quickly become familiar with the operations for creating, configuring, and connecting to an ApsaraDB for MongoDB instance.

Flowchart for an ApsaraDB for MongoDB instance

The first time you use ApsaraDB for MongoDB, you can start with Before you start.

The following figure shows the operations that you must perform before you use an ApsaraDB for MongoDB instance.

1. 创建副本集实例.
2. Set a password for a replica set instance.
3. Configure a whitelist for a replica set instance.
4. (Optional)Apply for a public endpoint for a replica set instance.
5. Connect to a replica set instance. For more information, see Overview of replica set instance connections.

4.Create an instance

5.Set a password for a replica set instance

This topic describes how to set or reset a password for a replica set instance.

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 **Accounts**.

6. Click **Reset Password**.

Basic Information	Account Name	Account Type	Status	Operation
Accounts	root The permissions are root privileges under the admin database.	normal	● Available	Reset Password
Database Connection				

7. In the **Reset Password** dialog box that appears, enter and confirm the new password. Click **OK**.

> ⑦ Note
>
>   ○ The password must contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! # $ % ^ & * ( ) _ + - =
>
>   ○ The password must be 8 to 32 characters in length.

# 6.Configure a whitelist for a replica set instance

This topic describes how to configure a whitelist for a replica set instance after you create the instance. Only the devices whose IP addresses are added to the whitelists of the instance are allowed access to the instance. The default whitelist only contains the IP address 127.0.0.1, which indicates that no devices can connect to the instance.

Context

- You must configure a whitelist upon the first use of an instance. After the whitelist is configured, the connection address of the instance is displayed on the **Basic Information** and **Database Connection** pages.

- Proper configuration of the whitelists can enhance access security of ApsaraDB for MongoDB. We recommend that you regularly maintain the whitelist.

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 > Whitelist Settings**.

6. Click the ⋮ icon in the **Actions** column, and select **Manually Modify** or **Import ECS Intranet IP**.

   - Click **Manually Modify**. In the dialog box that appears, enter an IP address or CIDR block, and click **OK**.

   - Click **Import ECS Intranet IP**. In the dialog box that appears, the internal IP addresses of the ECS instances of your Alibaba Cloud account are displayed. You can select the desired IP addresses, add them to a whitelist, and click **OK**.
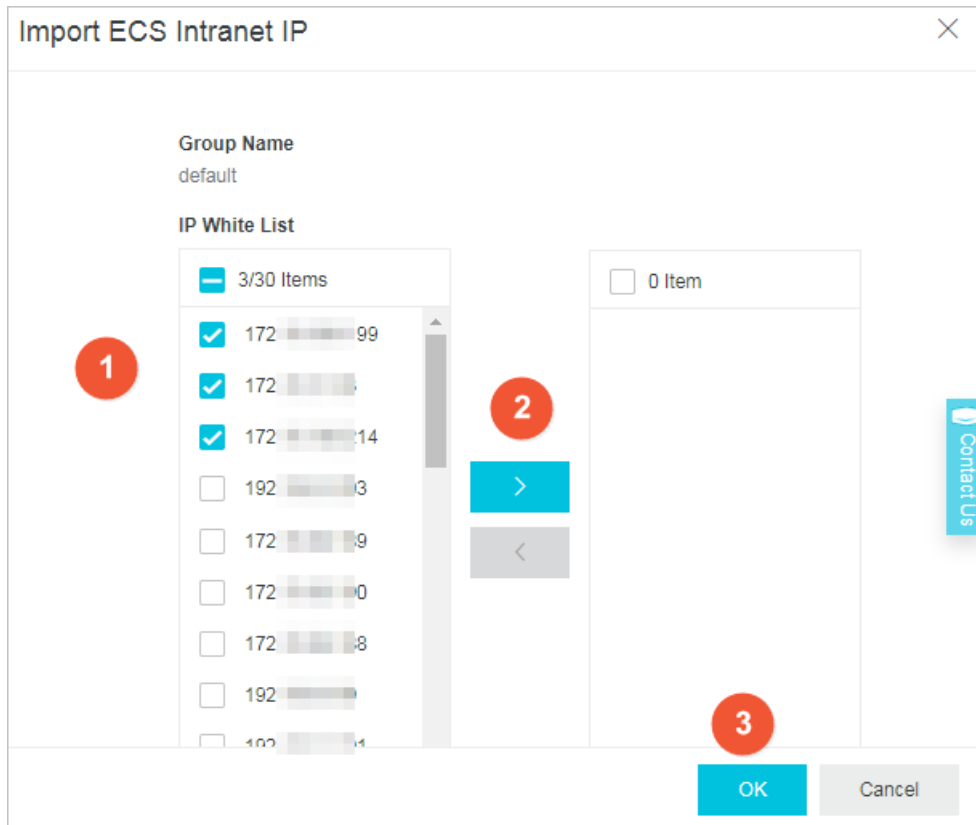
> **Note**
>
> ○ If a whitelist contains more than one IP address, separate them with commas (,). Every IP address in a whitelist must be unique. A whitelist can contain a maximum of 1,000 IP addresses.
>
>   Supported formats include 0.0.0.0/0, 10.23.12.24 (single IP address), and 10.23.12.24/24. 10.23.12.24/24 is a CIDR notation (for more information, see CIDR blocks), in which the suffix /24 indicates the number of bits for the prefix of the IP address. The prefix consists of 1 to 32 bits.
>
> ○ If the value is 0.0.0.0/0 or empty, the ApsaraDB for MongoDB instance can be accessed by all IP addresses. In this situation, the database is at high security risk.

More operations

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

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to

the same Alibaba Cloud account

Result

After a whitelist is configured, the VPC connection address of the instance is displayed on the **Basic Information** and **Database Connection** pages.

Quick Start for Replica Set·Apply for
a public endpoint for a replica set in
stance

ApsaraDB for MongoDB

7.Apply for a public endpoint for a replica set instance

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

Context

The following table describes the connections supported by ApsaraDB for MongoDB.

Connection type	Description
Intranet Connection - VPC	• A VPC is an isolated network with higher security and performance than a classic network. • By default, ApsaraDB for MongoDB provides endpoints on a VPC.
Intranet Connection - Classic Network	Cloud services on a classic network are not isolated. Unauthorized access can only be blocked by using security groups or whitelists. You can switch the network type to VPC. For more information, see Switch the network type of an ApsaraDB for MongoDB instance.
Public IP Connection	• Connecting to a replica set instance over the Internet is risky. Therefore, ApsaraDB for MongoDB does not provide public endpoints. • If you want to connect to a replica set instance from a device outside Alibaba Cloud (for example, a local client), you must apply for a public endpoint.

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 **Database Connection**.

6. In the **Public IP Connection** section, click **Apply for Public Connection String**.
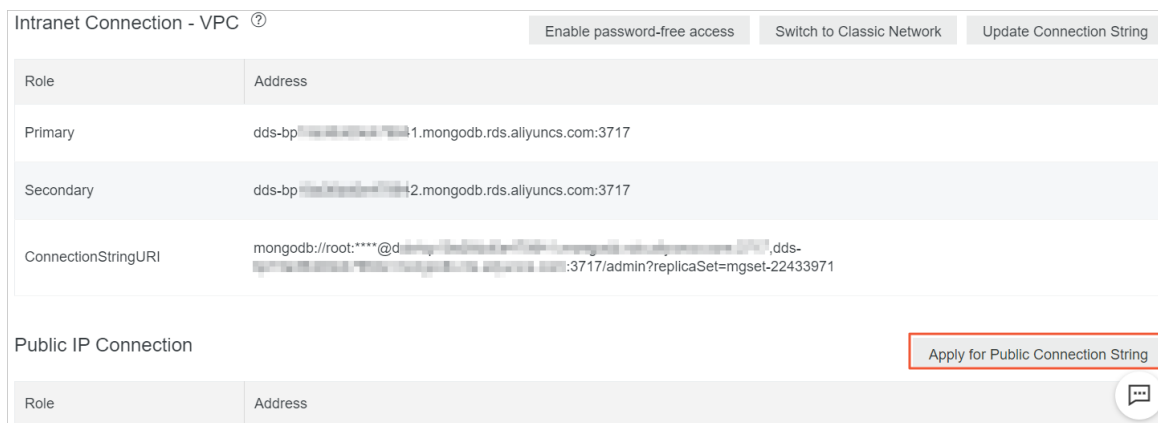
7. In the **Apply for Public Connection String** message that appears, click **OK**.

> ⑦ **Note**   If you want to connect to a replica set instance by using a public endpoint, you must add the public IP address of your client to a whitelist of this instance. For more information, see Configure a whitelist for a replica set instance.

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

8.Connect to an instance
8.1. Connect to a standalone ApsaraDB for MongoDB instance through DMS

Data Management (DMS) is an integrated database solution that offers data management, structure management, user authorization, security auditing, data trend analysis, data tracking, BI charts, performance optimization, and server management. You can use DMS to connect to a standalone ApsaraDB for MongoDB instance for easy management.

Preparations

Add the IP address of the DMS server (100.104.0.0/16) to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for a standalone ApsaraDB for MongoDB instance.
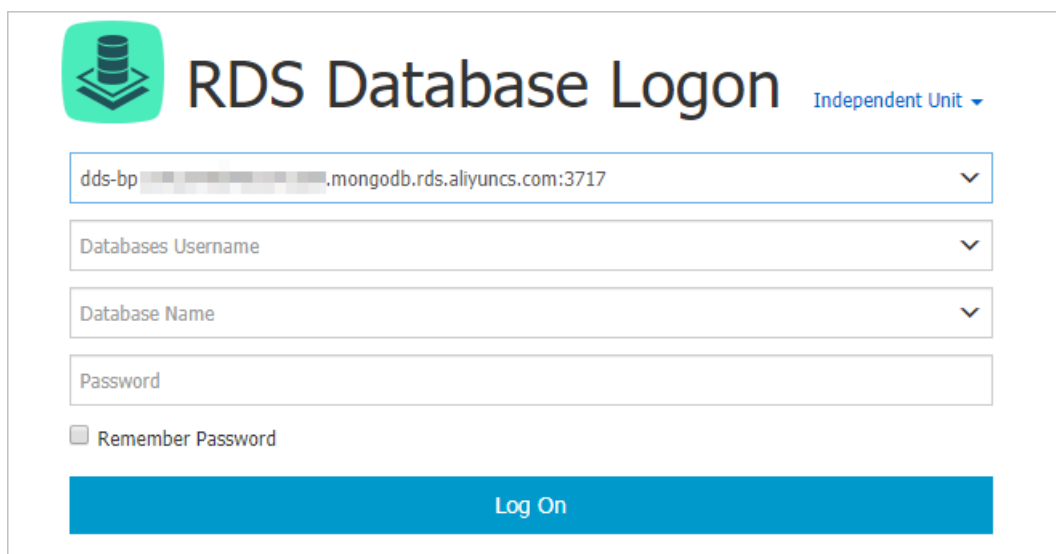
> ⑦ **Note** Skip this step if you have added the IP address of the DMS server to the whitelist of the ApsaraDB for MongoDB instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the 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. Click **Log On** and select **Primary** or **Secondary** in the upper-right corner of the Basic Information page . You are redirected to the **DMS console**.

   > ⑦ Note
   > - Primary: the primary node of the replica set instance. This node has read/write permissions on the database.
   > - Secondary: the secondary node of the replica set instance. This node only has read permissions on the database.

6. In the **DMS console**, enter the following information.

Item	Description
Network address:Port	The internal connection string of the primary node of the ApsaraDB for MongoDB instance is automatically entered.
Database Username	Enter the database account of the MongoDB instance. The initial account is root.
Database Name	Enter the name of the database to which the account belongs. ⑦ Note ○ If **Database Username** is set to root, the database name is admin. ○ We do not recommend that you log on to a database as the root user in the production environment. You can create users and grant permissions based on your business needs. For more information, see Use DMS to manage ApsaraDB for MongoDB users.
Password	The password of the specified account. ⑦ Note If you forget the password of the root account, you can reset the password by using the method specified in Set a password.

7. Click **Log On**.

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to

the same 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 high CPU utilization of ApsaraDB for MongoDB
- How to query and limit the number of connections

8.2. Connect to a replica set ApsaraDB for MongoDB instance through DMS

Data Management (DMS) is an integrated database solution that offers data management, structure management, user authorization, security auditing, data trend analysis, data tracking, BI charts, performance optimization, and server management. You can use DMS to connect to a replica set ApsaraDB for MongoDB instance for easy management.

Preparations

Add the IP address of the DMS server to the whitelist of the ApsaraDB for MongoDB instance based on the network type. For more information, see Configure a whitelist for a replica set instance.

> ⑦ **Note**   Skip this step if you have added the IP address of the DMS server 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 the 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 10.152.163.0/24

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. Click **Log On** and select **Primary** or **Secondary** in the upper-right corner of the Basic Information page. You are redirected to the **DMS console**.

> ⊘ Note
>
> ○ Primary: the primary node of the replica set instance. This node has read/write permissions on the database.
>
> ○ Secondary: the secondary node of the replica set instance. This node only has read permissions on the database.

6. In the **DMS console**, enter the following information.

Item	Description
Network address:Port	The internal connection string of the primary or secondary node of the ApsaraDB for MongoDB instance is automatically entered.
Database Username	Enter the database account of the ApsaraDB for MongoDB instance. The initial account is root.

Item	Description
Database Name	Enter the name of the database to which the account belongs. ⓘ **Note** ○ If **Database Username** is set to root, the database name is admin. ○ We do not recommend that you log on to a database as the root user in the production environment. You can create users and grant permissions based on your business needs. For more information, see Use DMS to manage ApsaraDB for MongoDB users.
Password	The password of the specified account. ⓘ **Note** If you forget the password of the root account, you can reset the password by using the method specified in Set a password.

7. Click **Log On**.

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same 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 high CPU utilization of ApsaraDB for MongoDB
- How to query and limit the number of connections

8.3. Connect to a sharded cluster ApsaraDB for MongoDB instance through DMS

Data Management (DMS) is an integrated database solution that offers data management, structure management, user authorization, security auditing, data trend analysis, data tracking, BI charts, performance optimization, and server management. You can use DMS to connect to a sharded cluster ApsaraDB for MongoDB instance for easy management.

Preparations

Add the IP address of the DMS server to the whitelist of the ApsaraDB for MongoDB instance based on the network type. For more information, see Configure a whitelist for a sharded cluster instance.
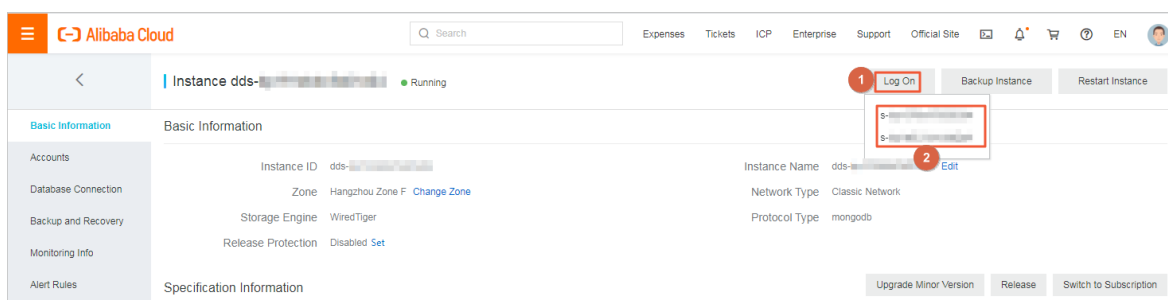
> ⑦ **Note**   Skip this step if you have added the IP address of the DMS server to the whitelist of the ApsaraDB for MongoDB instance.

IP addresses of the DMS server

Network type of ApsaraDB for MongoDB instance	IP address of the 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

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. Click **Log On** and select any Mongos node ID in the upper-right corner of the Basic Information page. You are redirected to the **DMS** console.

6. In the **DMS** console, enter the following information.

Item	Description
Network address:Port	The internal connection string of the Mongos node of the ApsaraDB for MongoDB instance is automatically entered.
Database Username	Enter the database account of the MongoDB instance. The initial account is root.
Database Name	Enter the name of the database to which the account belongs. ⑦ **Note** ○ If **Database Username** is set to root, the database name is admin. ○ We do not recommend that you log on to a database as the root user in the production environment. You can create users and grant permissions based on your business needs. For more information, see Use DMS to manage ApsaraDB for MongoDB users.
Password	The password of the specified account. ⑦ **Note** If you forget the password of the root account, you can reset the password by using the method specified in Set a password.

7. Click **Log On**.

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to

the same 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 high CPU utilization of ApsaraDB for MongoDB
- How to query and limit the number of connections

8.4. Connect to a standalone ApsaraDB for MongoDB instance by using the mongo shell

This topic describes how to connect to a standalone ApsaraDB for MongoDB instance by using the mongo shell, which is a database management tool built in MongoDB. You can install the mongo shell on your client or in an ECS instance.

Prerequisites

- Mongo shell 3.0 or later is installed to ensure successful authentication. For more information about the installation process, visit Install MongoDB at the official MongoDB website.
- The IP address of your client is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for a standalone ApsaraDB for MongoDB instance.

> ⑦ **Note**   If you want to connect to the instance over the Internet, you must apply for a public endpoint.

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 **Database Connection** to obtain the connection addresses of the primary node.

Database connection information

Item	Description
Address type	○ **Intranet Connection**: A VPC is an isolated virtual network with better security and performance than a classic network. By default, an ApsaraDB for MongoDB instance provides VPC connection addresses. ○ **Public IP Connection**: By default, ApsaraDB for MongoDB instances do not provide public connection addresses because connecting to instances over the Internet poses security risks. If you want to connect to an ApsaraDB for MongoDB instance from a device outside of Alibaba Cloud (such as a local device), you must apply for a public endpoint. For more information, see Apply for a public endpoint for a standalone ApsaraDB for MongoDB instance.
Node	Primary: indicates the primary node of the ApsaraDB for MongoDB instance. You can connect to this node to perform read/write operations on the database.
Connection string	The address of the primary node is in the format of `<host>:<port>`. ⓘ **Note** 　○ <host>: the endpoint of the primary node. 　○ <port>: the service port of the primary node.

Item	Description
	The connection string URI is in the following format: mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][? options]] ○ mongodb:// : the prefix, indicating a connection string URI. ○ username:password@ : the username and password used to connect to the ApsaraDB for MongoDB instance. Separate them with a colon (:). ○ hostX:portX : the endpoint and port number of the instance. ○ /database : the name of the authentication database. It is the database where the database account is created. ○ ? options : additional connection options.

6. Run the following command on the local server or ECS instance where the mongo shell is installed to connect to the database:

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

> ⑦ Note
> ○ <host:port>: the connection string of the primary node, including the endpoint and port number.
> ○ <username>: the database account of the ApsaraDB for MongoDB instance. The initial account is root. We recommend that you do not log on to a database as the root user in the production environment. You can create users and grant permissions based on your business needs. For more information, see Manage MongoDB users though DMS.
> ○ <database>: the name of the authentication database. It is the database where the database account is created. If the database account is root, enter admin.

Example:

```
mongo --host dds-bpxxxxxxxxx.mongodb.rds.aliyuncs.com:3717 -u root -p --authenticationDatabase admin
```

7. When Enter password: is displayed, enter the password for the database user and press Enter. If you forgot the password for the root user, you can reset it. For more information, see Set a password.

> ⑦ Note   The password you enter is not displayed.

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same 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
- How to troubleshoot high CPU utilization of ApsaraDB for MongoDB
- How to query and limit the number of connections

8.5. Connect to a replica set instance by using the mongo shell

This topic describes how to connect to a replica set instance by using the mongo shell, which is a database management tool provided with MongoDB. You can install the mongo shell on your client or in an ECS instance.

Prerequisites

- Mongo shell 3.0 or later is installed. For more information about the installation procedure, visit Install MongoDB at the official MongoDB website.
- The IP address of your client is added to a whitelist of the replica set instance. For more information, see Configure a whitelist for a replica set instance.

> ⑦ Note   If you want to connect to the instance over the Internet, you must apply for a public endpoint.

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 **Database Connection** to obtain the connection addresses of a node.

   > ⑦ Note   For more information about the connection addresses, see Introduction to connection addresses.

6. Connect to the replica set instance from your client or ECS instance that has the mongo shell installed.

○ Single-node connection

During regular tests, you can directly connect to a primary or secondary node. Note that a failover changes the roles of connected nodes, which affects read/write operations.
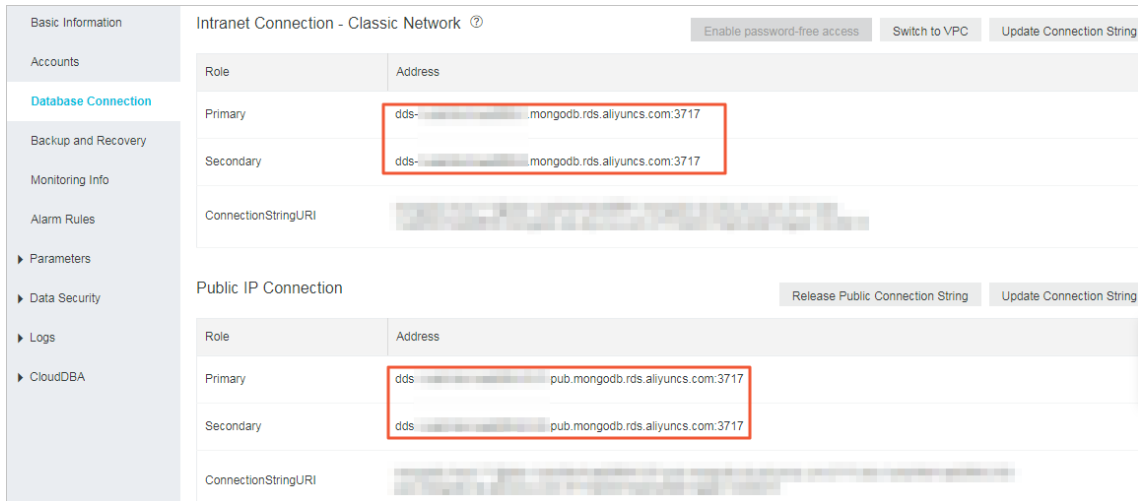
Command format:

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

> ⑦ **Note**
>
>   ▪ <host>: the connection string of the primary or secondary node.
>
>   ▪ Primary node: If you connect to this node, you can perform read/write operations on the databases of the replica set instance.
>
>   ▪ Secondary node: If you connect to this node, you can only perform read operations on the databases of the replica set instance.
>
>   ▪ <username>: the username you use to log on to a database of the replica set instance. The initial username is root. 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 them as needed. For more information, see Manage MongoDB users though DMS.
>
>   ▪ <database>: the name of the authentication database. It is the database where the database user is created. If the database username is root, enter admin.

Example:

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

When ` Enter password: ` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see Set a password for a replica set instance.

> ⑦ **Note**   The password you enter is not displayed.

○ HA connection (recommended): You can use a connection string URI to connect to both the primary and secondary nodes of a replica set instance. This guarantees that your application is always connected to the primary node and the read/write operations of your application are not affected even if the roles of the primary and secondary nodes are switched.

Command format:

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

> **?** **Note**
>
> ■ The connection string URI must be enclosed in a pair of double quotation marks ("").
>
> ■ <ConnectionStringURI>: the Connection String URI of the replica set instance.
>
> You must replace ******** in the Connection String URI with the database password. For more information about how to set a database password, see Set a password for a replica set instance.

Common connection scenarios

● Connect a local client to an ApsaraDB for MongoDB instance over the Internet

> **?** **Note** Before you connect to the replica set instance over the Internet, we recommend that you enable SSL encryption. For more information, see Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode.

● 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

● How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same 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

- How to troubleshoot the high CPU usage of ApsaraDB for MongoDB

- How to query and limit the number of connections

8.6. Connect to a sharded cluster instance by using the mongo shell

This topic describes how to connect to a sharded cluster instance by using the mongo shell, which is a database management tool provided with MongoDB. You can install the mongo shell on your client or an ECS instance.

Prerequisites

- To ensure successful authentication, the version of the mongo shell must match with that of the ApsaraDB for MongoDB instance. For more information about the installation procedure, visit Install MongoDB. Select the correct version based on your client.

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

> ⑦ **Note**   If you want to connect to the instance over the Internet, you must apply for a public endpoint.

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** to obtain the connection string of a mongos.

6. Connect to the sharded cluster instance from your client or ECS instance that has the mongo shell installed.

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

> ⑦ Note
> ○ <mongos_host>: the connection string of a mongos in the sharded cluster instance.
> ○ <username>: the username used to log on to a database of the sharded cluster 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 accounts and grant permissions to them as needed. For more information, see Manage MongoDB users though DMS.
> ○ <database>: the name of database corresponding to the username if authentication is enabled. If the username is root, enter admin.

Example:

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

7. When `Enter password:` is displayed, enter the password of the database user and press Enter. If you forget the password of the root user, you can reset it. For more information, see Set a password for a sharded cluster instance.

> ⑦ Note   The password characters are not displayed when you enter the password.

Common connection scenarios

- Connect a local client 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
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same 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
- How to troubleshoot the high CPU utilization of ApsaraDB for MongoDB
- How to query and limit the number of connections

8.7. Connection sample code for MongoDB drivers

Related links

- MongoDB Drivers
- Connection String URI Format

> ⑦ **Note**   The connection sample code in this topic applies only when you use intranet addresses provided by Alibaba Cloud to connect to ApsaraDB for MongoDB.

Node.js

Related link: MongoDB Node.js Driver

1. Initialize a project.

```
mkdir node-mongodb-demo
cd node-mongodb-demo
npm init
```

2. Install the driver package and tool kit.

```
npm install mongodb node-uuid sprintf-js -save
```

3. Obtain the information required to connect to an ApsaraDB for MongoDB instance.

   For more information, see Overview of replica set instance connections.

4. Use the following Node.js sample code.

```
'use strict';
var uuid = require('node-uuid');
var sprintf = require("sprintf-js").sprintf;
var mongoClient = require('mongodb').MongoClient;
var host1 = "demotest-1.mongodb.tbc3.newtest.rdstest.aliyun-inc.com";
var port1 = 27017;
var host2 = "demotest-2.mongodb.tbc3.newtest.rdstest.aliyun-inc.com";
var port2 = 27017;
var username = "demouser";
var password = "123456";
var replSetName = "mgset-1441984991";
var demoDb = "test";
var demoColl = "testColl";
// The officially recommended solution.
var url = sprintf("mongodb://%s:%d,%s:%d/%s?replicaSet=%s", host1, port1, host2, port2, demoDb, re
```

```
plSetName);
console.info("url:", url);
// Obtain the MongoClient.
mongoClient.connect(url, function (err, db) {
  if (err) {
    console.error("connect err:", err);
    return 1;
  }
  // Authenticate the username and password used to log on to ApsaraDB for MongoDB. The username i
n this sample code is used to log on to the admin database.
  var adminDb = db.admin();
  adminDb.authenticate(username, password, function (err, result) {
    if (err) {
      console.error("authenticate err:", err);
      return 1;
    }
    // Obtain the collection handle.
    var collection = db.collection(demoColl);
    var demoName = "NODE:" + uuid.v1();
    var doc = { "DEMO": demoName, "MESG": "Hello AliCoudDB For MongoDB" };
    console.info("ready insert document: ", doc);
    // Insert data.
    collection.insertOne(doc, function (err, data) {
      if (err) {
        console.error("insert err:", err);
        return 1;
      }
      console.info("insert result:", data["result"]);
      // Read data.
      var filter = { "DEMO": demoName };
      collection.find(filter).toArray(function (err, items) {
        if (err) {
          console.error("find err:", err);
          return 1;
        }
        console.info("find document: ", items);
        // Close the client and release resources.
        db.close();
      });
    });
  });
```

```
});
```

PHP

Related link:

MongoDB PHP Driver

1. Install the driver package and tool kit.

```
$ pecl install mongodb
$ echo "extension=mongodb.so" >> `php --ini | grep "Loaded Configuration" | sed -e "s|.*:\s*||"`
$ composer require "mongodb/mongodb=^1.0.0"
```

2. Obtain the information required to connect to an ApsaraDB for MongoDB instance.

   For more information, see Overview of replica set instance connections.

3. Use the following PHP sample code.

```php
<?php
require 'vendor/autoload.php'; // include Composer goodies
# Specify the instance information.
$demo_seed1 = 'demotest-1.mongodb.test.aliyun-inc.com:3717';
$demo_seed2 = 'demotest-2.mongodb.test.aliyun-inc.com:3717';
$demo_replname = "mgset-1441984463";
$demo_user = 'root';
$demo_password = '123456';
$demo_db = 'admin';
# Construct a MongoDB connection string URI based on the instance information.
# mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?opti
ons]]
$demo_uri = 'mongodb://' . $demo_user . ':' . $demo_password . '@' .
   $demo_seed1 . ',' . $demo_seed2 . '/' . $demo_db . '?replicaSet=' . $demo_replname;
$client = new MongoDB\Client($demo_uri);
$collection = $client->testDb->testColl;
$result = $collection->insertOne(['name' => 'ApsaraDB for Mongodb', 'desc' => 'Hello, Mongodb']);
echo "Inserted with Object ID '{$result->getInsertedId()}'", "\n";
$result = $collection->find(['name' => 'ApsaraDB for Mongodb']);
foreach ($result as $entry) {
   echo $entry->_id, ': ', $entry->name, "\n";
}
?>
```

Java

Related links:

- Official Quick Start
- JAR package download

1. Obtain the information required to connect to an ApsaraDB for MongoDB instance.

   For more information, see Overview of replica set instance connections.

2. Use the following Java sample code.

   ○ Maven configuration

```xml
<dependencies>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.0.4</version>
  </dependency>
</dependencies>
```

   ○ Java sample code

```java
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import org.bson.BsonDocument;
import org.bson.BsonString;
import org.bson.Document;
import com.mongodb.MongoClient;
import com.mongodb.MongoClientOptions;
import com.mongodb.MongoClientURI;
import com.mongodb.MongoCredential;
import com.mongodb.ServerAddress;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
 public class Main {
     public static ServerAddress seed1 = new ServerAddress("demotest-1.mongodb.tbc3.newtest.rds
test.aliyun-inc.com",
             27017);
     public static ServerAddress seed2 = new ServerAddress("demotest-2.mongodb.tbc3.newtest.rds
test.aliyun-inc.com",
             27017);
     public static String username = "demouser";
     public static String password = "123456";
     public static String ReplSetName = "mgset-1441984463";
     public static String DEFAULT_DB = "admin";
```

```java
    public static String DEFAULT_DB = "admin";
    public static String DEMO_DB = "test";
    public static String DEMO_COLL = "testColl";
    public static MongoClient createMongoDBClient() {
        // Construct a seed list.
        List<ServerAddress> seedList = new ArrayList<ServerAddress>();
        seedList.add(seed1);
        seedList.add(seed2);
        // Construct authentication information.
        List<MongoCredential> credentials = new ArrayList<MongoCredential>();
        credentials.add(MongoCredential.createScramSha1Credential(username, DEFAULT_DB,
                password.toCharArray()));
        // Construct operation options. Configure options other than requiredReplicaSetName based
on your actual requirements. Default parameter settings can meet requirements in most scenarios.
        MongoClientOptions options = MongoClientOptions.builder().requiredReplicaSetName(ReplS
etName)
                .socketTimeout(2000).connectionsPerHost(1).build();
        return new MongoClient(seedList, credentials, options);
    }
    public static MongoClient createMongoDBClientWithURI() {
        // Use a URI to initialize the MongoClient.
        // mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[datab
ase][?options]]
        MongoClientURI connectionString = new MongoClientURI("mongodb://" + username + ":" + pa
ssword + "@"
                + seed1 + "," + seed2 + "/" + DEFAULT_DB + "?replicaSet=" + ReplSetName);
        return new MongoClient(connectionString);
    }
    public static void main(String args[]) {
        MongoClient client = createMongoDBClient();
        // or
        // MongoClient client = createMongoDBClientWithURI();
        try {
            // Obtain the collection handle.
            MongoDatabase database = client.getDatabase(DEMO_DB);
            MongoCollection<Document> collection = database.getCollection(DEMO_COLL);
            // Insert data.
            Document doc = new Document();
            String demoname = "JAVA:" + UUID.randomUUID();
            doc.append("DEMO", demoname);
            doc.append("MESG", "Hello AliCoudDB For MongoDB");
```

```
        collection.insertOne(doc);
        System.out.println("insert document: " + doc);
        // Read data.
        BsonDocument filter = new BsonDocument();
        filter.append("DEMO", new BsonString(demoname));
        MongoCursor<Document> cursor = collection.find(filter).iterator();
        while (cursor.hasNext()) {
            System.out.println("find document: " + cursor.next());
        }
    } finally {
        // Close the client and release resources.
        client.close();
    }
    return;
    }
}
```

Python

Related links:

- PyMongo download
- Official documentation

1. Install PyMongo.

   ```
   pip install pymongo
   ```

2. Obtain the information required to connect to an ApsaraDB for MongoDB instance.

   For more information, see Overview of replica set instance connections.

3. Use the following Python sample code.

```
import uuid
from pymongo import MongoClient
 # Specify two addresses for connecting to the primary and secondary nodes of the instance.
CONN_ADDR1 = 'demotest-1.mongodb.tbc3.newtest.rdstest.aliyun-inc.com:27017'
CONN_ADDR2 = 'demotest-2.mongodb.tbc3.newtest.rdstest.aliyun-inc.com:27017'
REPLICAT_SET = 'mgset-1441984463'
 username = 'demouser'
password = '123456'
 # Obtain the MongoClient.
client = MongoClient([CONN_ADDR1, CONN_ADDR2], replicaSet=REPLICAT_SET)
 # Authenticate the username and password used to log on to ApsaraDB for MongoDB. The username in
this sample code is used to log on to the admin database.
client.admin.authenticate(username, password)
 # Use the collection:testColl of the test database as an example. Insert doc and search for documents b
ased on the demo name.
demo_name = 'python-' + str(uuid.uuid1())
print 'demo_name:', demo_name
 doc = dict(DEMO=demo_name, MESG="Hello ApsaraDB For MongoDB")
doc_id = client.test.testColl.insert(doc)
print 'doc_id:', doc_id
 for d in client.test.testColl.find(dict(DEMO=demo_name)):
    print 'find documents:', d
```

C#

Related links:MongoDB C# Driver.

1. Obtain the information required to connect to an ApsaraDB for MongoDB instance. For more
   information, see Overview of replica set instance connections.

2. Install the driver package.

   ```
   mongocsharpdriver.dll
   ```

3. C# Demo Code.

   ```
   using MongoDB.Driver;
   using System;
   using System.Collections.Generic;
   namespace Aliyun
   {
     class Program
     {
       static void Main(string[] args)
       {
   ```

```
//Mongo Instance Information
const string host1 = "dds-t4nffb7491eb87541.mongodb.singapore.rds.aliyuncs.com";
const int port1 = 3717;
const string host2 = "dds-t4nffb7491eb87542.mongodb.singapore.rds.aliyuncs.com";
const int port2 = 3717;
const string replicaSetName = "mgset-300374302";
const string admin = "admin";
const string userName = "root";
const string passwd = "Lsx111222";
try
{
  Console.WriteLine("connecting...");
  MongoClientSettings settings = new MongoClientSettings();
  List<MongoServerAddress> servers = new List<MongoServerAddress>();
  servers.Add(new MongoServerAddress(host1, port1));
  servers.Add(new MongoServerAddress(host2, port2));
  settings.Servers = servers;
  //set ReplicaSetName
  settings.ReplicaSetName = replicaSetName;
  //set ConnectTimeout to 3
  settings.ConnectTimeout = new TimeSpan(0, 0, 0, 3, 0);
  MongoCredential credentials = MongoCredential.CreateCredential(admin, userName, passwd);
  settings.Credential = credentials;
  MongoClient client = new MongoClient(settings);
  var server = client.GetServer();
  MongoDatabase database = server.GetDatabase("test");
  var collection = database.GetCollection<User>("test_collection");
  User user = new User();
  user.id = "1";
  user.name = "mongo_test";
  user.sex = "Female";
  //insert data user
  collection.Insert(user);
  //obtain a data
  User result = collection.FindOne();
  Console.WriteLine("id:" + result.id + " name:" + result.name + " sex:"+result.sex);
  Console.WriteLine("connection successful........");
}
catch (Exception e)
{
  Console.WriteLine("connection failed:"+e.Message);
```

```
            Console.WriteLine( connection failed: +e.Message);
        }
    }
}
class User
{
    public string id { set; get; }
    public string name { set; get; }
    public string sex { set; get; }
}
}
```

9.Migrate data

9.1. Migrate user-created standalone MongoDB databases to Alibaba Cloud by using DTS

This topic describes how to use Data Transmission Service (DTS) to migrate data from standalone user-created MongoDB databases to Alibaba Cloud. DTS supports full data migration and incremental data migration.

To migrate all data without service interruption, you can select both full data migration and incremental data migration. You can also use the built-in commands of MongoDB to migrate user-created MongoDB databases. For more information, see Migrate user-created databases to Alibaba Cloud by using tools provided by MongoDB.

For more information about data migration or synchronization solutions, see Overview.

Prerequisites

- The version of the user-created MongoDB database is 4.2, 4.0, 3.0, 3.2, or 3.4.

- The storage space of the ApsaraDB for MongoDB instance is larger than the size of the user-created MongoDB database.

Precautions

- To avoid service disruptions, we recommend that you migrate data 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 databases. For more information about MongoDB versions and storage engines that are supported by ApsaraDB for MongoDB, see 版本及存储引擎.

- To migrate incremental data from a standalone user-created MongoDB database, you must enable the oplog feature for the database. For more information, see Preparations for incremental data migration.

Billing

Migration type	Instance configurations	Internet traffic
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

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

> ⑦ **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 MongoDB database.

  > ⑦ **Note**
  >
  > ○ The create and delete operations for databases, collections, and indexes can be synchronized.
  >
  > ○ The create, delete, and update operations for documents can also be synchronized.

Permissions required for database accounts

Database	Full data migration	Incremental data migration
Source user-created MongoDB database	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

For more information about how to create and authorize a database account:

- Create User for MongoDB for a user-created MongoDB database
- Manage MongoDB users though DMS for an ApsaraDB for MongoDB instance

Preparations for incremental data migration

Before you use DTS to migrate incremental data, enable the oplog feature for the source database. If you only perform full data migration, skip the following steps.

> ⑦ **Note**   This operation restarts the MongoDB database. We recommend that you perform this operation during off-peak hours.

1. Use Mongo Shell to connect to the user-created MongoDB database.

2. Run the following commands to shut down the MongoDB database:

   ```
   use admin
   db.shutdownServer()
   ```

3. Run the following command to start the MongoDB database in the background as a replica set:

   ```
   mongod --port 27017 --dbpath /var/lib/mongodb --logpath /var/log/mongodb/mongod.log --replSet rs0
   --bind_ip 0.0.0.0 --auth --fork
   ```

> ⑦ **Note**
>
> - In this command, */var/lib/mongodb* is the database path, and */var/log/mongodb/mongod.log* is the log file path. Specify the paths based on business needs.
>
> - This command uses 0.0.0.0 as the associated IP address of the MongoDB database. This allows you to access the database by using all IP addresses. After the migration is complete, run the `kill` command to end the process, and start the MongoDB database by using the original configuration file.
>
> - This command enables authentication. You can only access the database after you pass the authentication.

4. Use Mongo Shell to reconnect to the user-created MongoDB database.

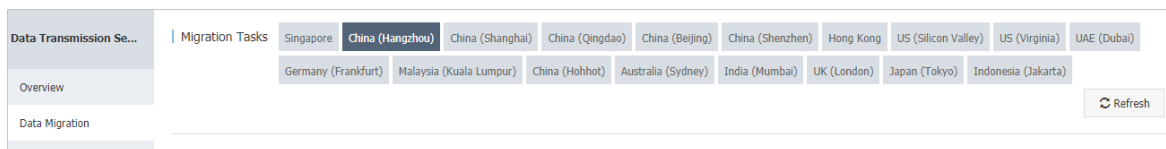5. Run the following commands to initialize the replica set:

```
use admin
rs.initiate()
```

6. The role of the current node changes to primary.

> ⑦ **Note** You can run the `rs.printReplicationInfo()` command to view the status of oplog.

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 in which the ApsaraDB for MongoDB instance resides.

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

5. Configure the source and destination databases.

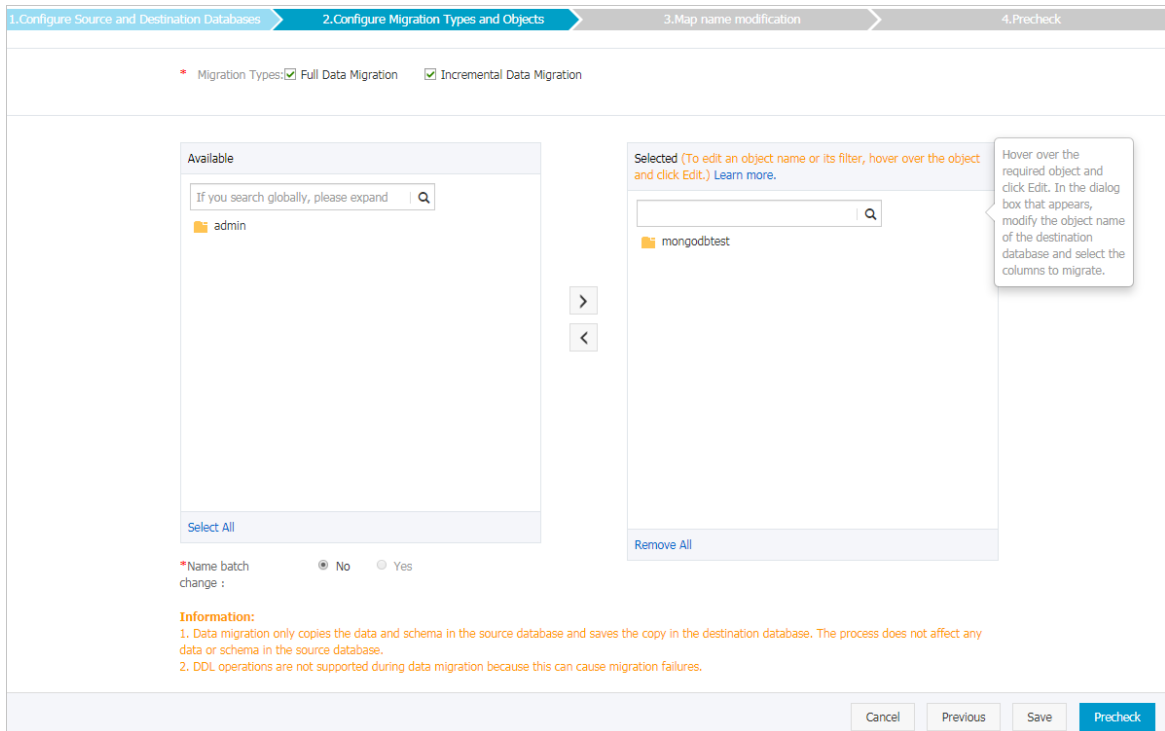| Section	Parameter	Description
N/A	Task name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select an instance type based on the location where the database is deployed. In this topic, a **User-Created Database with Public IP Address** is used as an example. ⑦ **Note** If you select other instance types, you must prepare the environment that is required for the source database. For more information, see Preparation overview.
	Instance Region	If Instance Type is set to **User-Created Database with Public IP Address**, you do not need to specify the **Instance Region** parameter. ⑦ **Note** If you have configured a whitelist for the user-created MongoDB database, you must add the CIDR blocks of DTS servers to the whitelist. You can click **Get IP Address Segment of DTS** next to **Instance Region** to obtain the CIDR blocks of DTS servers.
	Database Type	Select **MongoDB**.

Section	Parameter	Description
Source Database	Hostname or IP Address	Enter the endpoint that is used to connect to the user-created MongoDB database. In this example, enter the public IP address.
	Port Number	Enter the service port of the user-created MongoDB database. ⑦ **Note** In this example, the service port of the user-created MongoDB database must be open to the public network.
	Database Name	Enter the name of the authentication database to which the database account belongs.
	Database Account	Enter the username of the database account that you use to manage the source 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. ⑦ **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 **Failed** message is displayed, you can click **Check** next to the **Failed** message to modify the information as prompted.
	Connection Method	Select **Non-encrypted**. ⑦ **Note** The **SSL-encrypted** option is available only when you migrate MongoDB Atlas.
	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region in which 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 to which the database account belongs. ⑦ **Note** If you want to use the root account, specify admin for the Database Name parameter.
Destinatio n Database		

Database Section	Parameter	Description
	Database Account	Enter the username of the database account that you use to manage the destination 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. ⑦ **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 **Failed** message is displayed, click **Check** next to the **Failed** message to modify the information as prompted.

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

⑦ **Note**   The CIDR blocks of DTS servers are automatically added to the whitelist of the destination RDS instance. This ensures that DTS servers can connect to the destination ApsaraDB for MongoDB instance. After the migration is completed, you can remove these CIDR blocks from the whitelist. For more information, see Configure a whitelist for a replica set instance.

7. Configure migration types and objects to be migrated.

Paramet er	Description
Migratio n Types	- To perform only full data migration, select **Full Data Migration**. - To migrate data with minimal downtime, select both **Full Data Migration** and **Incremental Data Migration**. ⑦ Note - Before migrating incremental data from a standalone user-created MongoDB database, you must enable oplog for the database. For more information, see Preparations for incremental data migration. - If the **Incremental Data Migration** option is not selected, do not write new data to the user-created MongoDB database when full data migration is in progress. Otherwise, data inconsistency may occur.
Migratio n objects	- Select objects from the **Available** section and click the ⟩ icon to move the objects to the **Selected** section. ⑦ Note - Data in the admin and local databases cannot be migrated. - The config database is an internal database. We recommend that you do not migrate data in the config database. - A migration object can be a database, collection, or function. - By default, the name of an object remains unchanged after migration. You can change the names of the objects in the destination RDS instance by using the object name mapping feature. For more information, see Object name mapping.

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

   ⑦ 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.
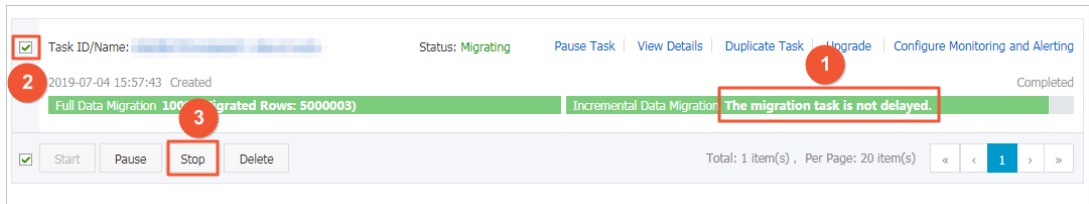
    ○ Full data migration

      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.

    > ⑦ **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.

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

9.2. Migrate the replica set of a user-created MongoDB database to ApsaraDB for MongoDB by using DTS

This topic describes how to migrate the replica set of a user-created MongoDB database to ApsaraDB for MongoDB by using Data Transmission Service (DTS). DTS supports full data migration and incremental data migration.

For more information about data migration or synchronization solutions, see Overview.

Prerequisites

● The version of the user-created MongoDB database is 3.0, 3.2, 3.4, 3.6, or 4.0.

● The storage capacity of the ApsaraDB for MongoDB instance is larger than the size of the user-created MongoDB database.

Precautions

● During full data migration, DTS occupies some storage space of the source and destination

databases. This may increase the load of the database servers. If you migrate a large volume of data
or the server specifications cannot meet your requirements, the databases may be overloaded or
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
  databases. For more information about MongoDB versions and storage engines that are supported
  by ApsaraDB for MongoDB, see 版本及存储引擎.

Billing

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

Migration types

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

  > ② 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 MongoDB database.

  > ② Note
  > - The create and delete operations for databases, collections, and indexes can also be
  >   synchronized.
  > - The create, delete, and update operations for documents can be synchronized.

Permissions required for database accounts

Database	Full data migration	Incremental data migration
Source user-created MongoDB database	Read permissions on the source database	Read permissions on the source database, admin database, and local database
Destination ApsaraDB for MongoDB database	Read/write permissions on the destination database	Read/write permissions on the destination database

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
- Create User in MongoDB for a user-created MongoDB database

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 in which the ApsaraDB for MongoDB instance resides.

Data Transmission Se...	Migration Tasks	Singapore	China (Hangzhou)	China (Shanghai)	China (Qingdao)	China (Beijing)	China (Shenzhen)	Hong Kong	US (Silicon Valley)	US (Virginia)	UAE (Dubai)
Overview		Germany (Frankfurt)	Malaysia (Kuala Lumpur)	China (Hohhot)	Australia (Sydney)	India (Mumbai)	UK (London)	Japan (Tokyo)	Indonesia (Jakarta)		
Data Migration										Refresh	

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

5. Configure the source and destination database.

Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.

Section	Parameter	Description
Source Database	Instance Type	Select an instance type based on the location where the database is deployed. In this topic, a **User-Created Database with Public IP Address** is used as an example. ⑦ **Note** If you select other instance types, you must prepare the environment that is required for the source database. For more information, see Preparation overview.
	Instance Region	If Instance Type is set to **User-Created Database with Public IP Address**, you do not need to specify the **Instance Region** parameter. ⑦ **Note** If you have configured a whitelist for the user-created MongoDB database, you must add the CIDR blocks of DTS servers to the whitelist. You can click **Get IP Address Segment of DTS** next to **Instance Region** to obtain the CIDR blocks of DTS servers.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the IP address of the user-created MongoDB database. In this example, enter the public IP address.
	Port Number	Enter the service port of the user-created MongoDB database. ⑦ **Note** In this example, the service port of the user-created MongoDB database must be open to the public network.
	Database Name	Enter the name of the authentication database to which the database account belongs.
	Database Account	Enter the account that is used to log on to the user-created MongoDB database. For more information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the source database account. ⑦ **Note** After you specify the source database information, click **Test Connectivity** next to **Database Password** to check whether the information is correct. If the source database configuration is correct, the **Passed** message is displayed. If the **Test Failed** message is displayed, click **Check** next to the **Failed** message to modify the information as prompted.

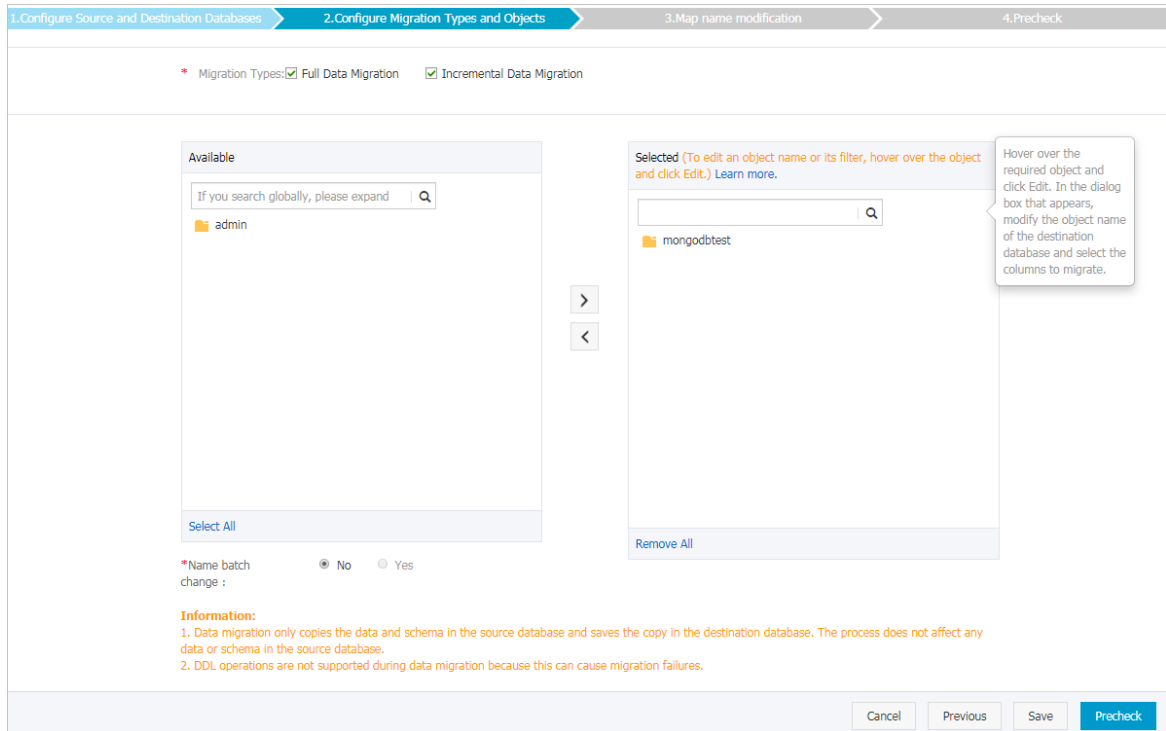| Section	Parameter	Description
	Connection Method	Select **Non-encrypted**. ⑦ **Note** The **SSL-encrypted** option is available only when you migrate MongoDB Atlas.
Destination Database	Instance Type	Select **MongoDB Instance**.
	Instance Region	Select the region in which the ApsaraDB for MongoDB instance resides.
	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. ⑦ **Note** If you want to use the root account, specify admin for the Database Name parameter.
	Database Account	Enter the account that is used to log on to 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. ⑦ **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 of the page, click **Set Whitelist and Next**.

> ⑦ **Note**   The CIDR blocks of DTS servers are automatically added to the whitelist of the destination RDS instance. This ensures that DTS servers can connect to the destination ApsaraDB for MongoDB instance. After the migration is completed, you can remove these CIDR blocks from the whitelist. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

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

Parameter	Description
Migration Types	- To perform only full data migration, select **Full Data Migration**. - To migrate data with minimal downtime, select both **Full Data Migration** and **Incremental Data Migration**. ⑦ **Note** If the **Incremental Data Migration** option is not selected, do not write new data to the user-created MongoDB database when full data migration is in progress. Otherwise, data inconsistency may occur.
Migration objects	- Select objects from the **Available** section and click the ⟩ icon to move the objects to the **Selected** section. ⑦ **Note** - Data in the admin and local databases cannot be migrated. - The config database is an internal database. We recommend that you do not migrate data in the config database. - A migration object can be a database, collection, or function. - By default, the name of an object remains unchanged after migration. You can change the names of the objects in the destination RDS instance by using the object name mapping feature. 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 for each failed check 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 the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

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

      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 succeeds.

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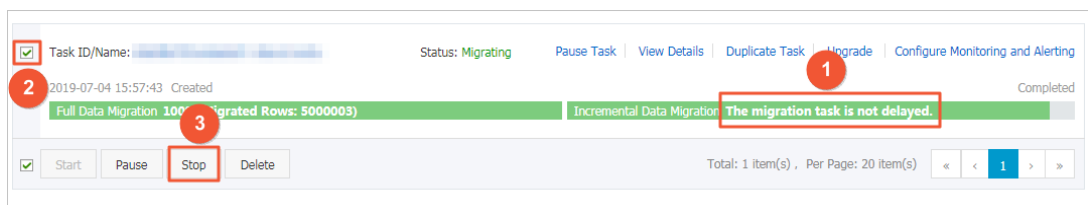
12. Switch your workloads to the ApsaraDB for MongoDB instance.

References

How do I connect to the replica set instance of ApsaraDB for MongoDB?

9.3. Migrate a user-created sharded MongoDB database to ApsaraDB for MongoDB by using DTS

This topic describes how to migrate shards of a user-created sharded MongoDB database to ApsaraDB for MongoDB by using Data Transmission Service (DTS). DTS allows you to migrate historical and incremental data without service disruptions.

For more information about data migration and synchronization solutions, see Overview of data migration and synchronization.

Prerequisite

- The version of the user-created MongoDB database is 3.0, 3.2, 3.4, 3.6, or 4.0.
- Each shard in the destination sharded cluster instance has sufficient storage space.

> ② Note   For example, a user-created MongoDB database has three shards, and one of these shards occupies the maximum storage space of 500 GB. In this case, the storage space of each shard in destination instance must be greater than 500 GB.

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.

> ② 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.
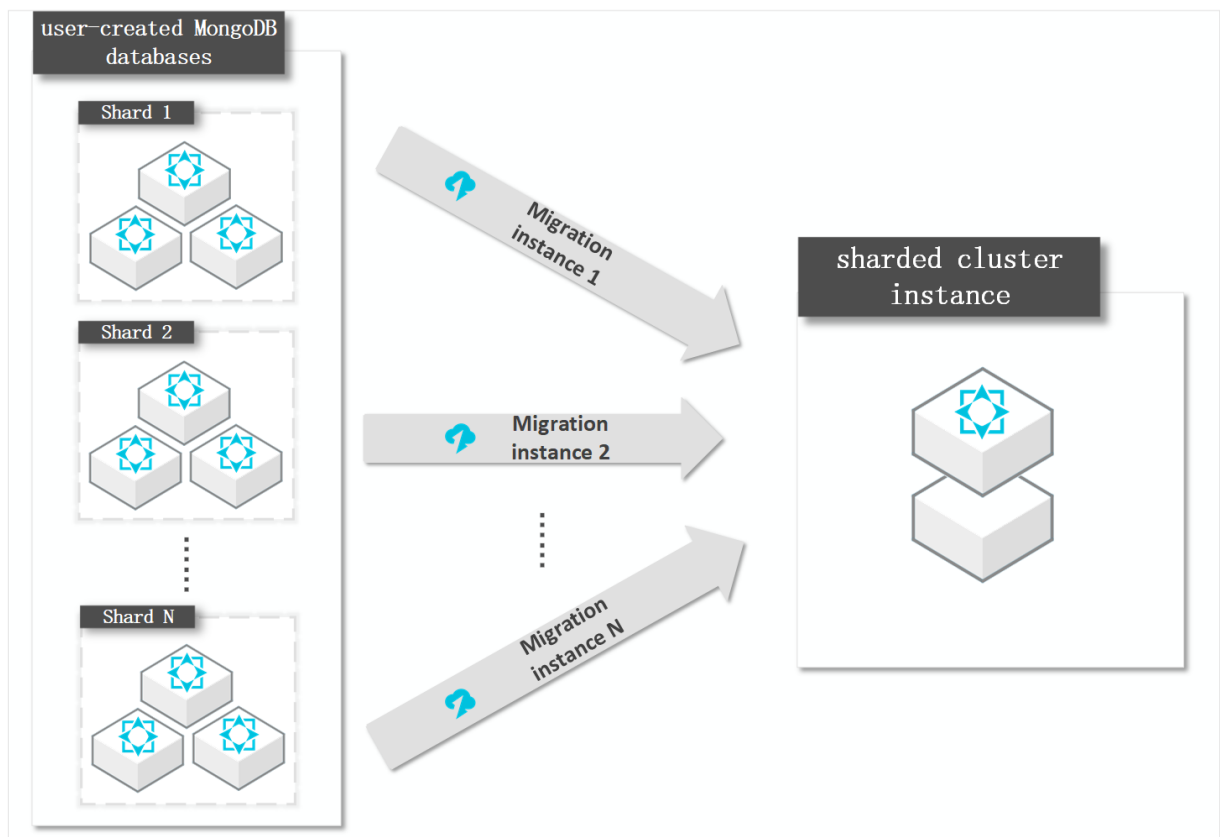
Usage notes

- 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 版本及存储引擎.

Billing information

Migration type	Instance 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 Data Transmission Service Pricing.
Incremental data migration	Charged. For more information, seeData Transmission Service Pricing.	

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.

Required database account permissions

Data source	Support for full data migration	Support for incremental data migration
Source user-created MongoDB database	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

How to create and authorize a database account:

- For a user-created MongoDB database, see db.createUser().
- For an ApsaraDB for MongoDB instance, see Manage MongoDB users through DMS.

Prerequisites

1. To avoid the impact of block migration on data consistency, you must disable the balancer of the user-created MongoDB databases during migration. For more information, see Manage the ApsaraDB for MongoDB balancer.

> 🔔 **Warning**   If the balancer is not disabled, block migration will affect the consistency of the data read by DTS.

2. Delete the orphaned documents generated due to migration failures from the user-created MongoDB database.

> ⑦ **Note**   If the orphaned documents are not deleted, the documents with `_id` conflicts may exist during migration and unwanted data may be migrated.

   i. Download the cleanupOrphaned.js file.

   ```
   wget "http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/120562/cn_zh/156445123
   7979/cleanupOrphaned.js"
   ```

   ii. Replace `test` in the *cleanupOrphaned.js* file with the name of the database where you want to delete orphaned documents.

   > ⑦ **Note**   If you want to delete orphaned documents from multiple databases, repeat Steps ii and iii.

   ```
   function cleanupOrphaned(coll) {
     var nextKey = { };
     var result;

     while ( nextKey != null ) {
        result = db.adminCommand( { cleanupOrphaned: coll, startingFromKey: nextKey } );

        if (result.ok != 1)
            print("Unable to complete at this time: failure or timeout.")

        printjson(result);

        nextKey = result.stoppedAtKey;
     }
   }

   var dbName = 'test'
   db = db.getSiblingDB(dbName)
   db.getCollectionNames().forEach(function(collName) {
          cleanupOrphaned(dbName + "." + collName);
   });
   ```

iii. Run the following command on a shard to delete the orphaned documents from all collections in the specified database:

> ⑦ **Note**   You must repeat this step on each shard.

```
mongo --host <Shardhost> --port <Primaryport>  --authenticationDatabase <database> -u <username> -p <passowrd> cleanupOrphaned.js
```

> ⑦ **Note**
> - <Shardhost>: the IP address of the shard.
> - <Primaryport>: the service port of the primary node of the shard.
> - <database>: the database corresponding to the username and password if authentication is enabled.
> - <username>: the account used to log on to the user-created MongoDB database.
> - <password>: the password used to log on to the user-created MongoDB database.

The following figure shows an example.

In this example, a user-created MongoDB database has three shards, and you must delete the orphaned documents on each of the shards.

```
mongo --host 172.16.1.10 --port 27018  --authenticationDatabase admin -u root -p 'Test123456' cleanupOrphaned.js
```

```
mongo --host 172.16.1.11 --port 27021 --authenticationDatabase admin -u root -p 'Test123456' cleanupOrphaned.js
```

```
mongo --host 172.16.1.12 --port 27024  --authenticationDatabase admin -u root -p 'Test123456' cleanupOrphaned.js
```

3. Create required databases and collections in the destination sharded cluster instance, and configure data sharding for the databases and collections. For more information, see Configure sharding to maximize the performance of shards.

> ⑦ **Note**   If you configure data sharding before you start data migration, data in the user-created MongoDB database is evenly migrated to the shards in the destination sharded cluster instance. This prevents overloading a single shard.

Procedure

1. Log on to the Data Transmission Service console.

2. In the left-side navigation pane, click **Data Migration**.

3. In the **Migration Tasks** section, select the region in which the ApsaraDB for MongoDB instance is deployed.
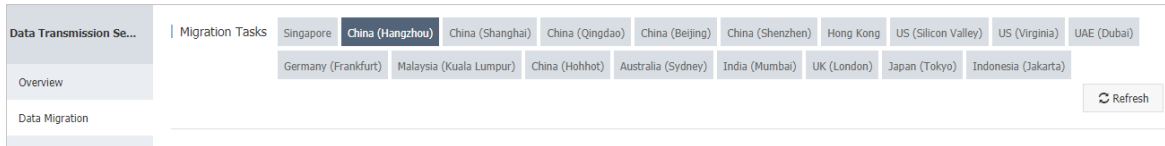
4. In the upper-right corner, click **Create Migration Task**.

5. Click Create Migration Task. In the Configure Source and Destination step, configure the source and destination databases for the migration task.

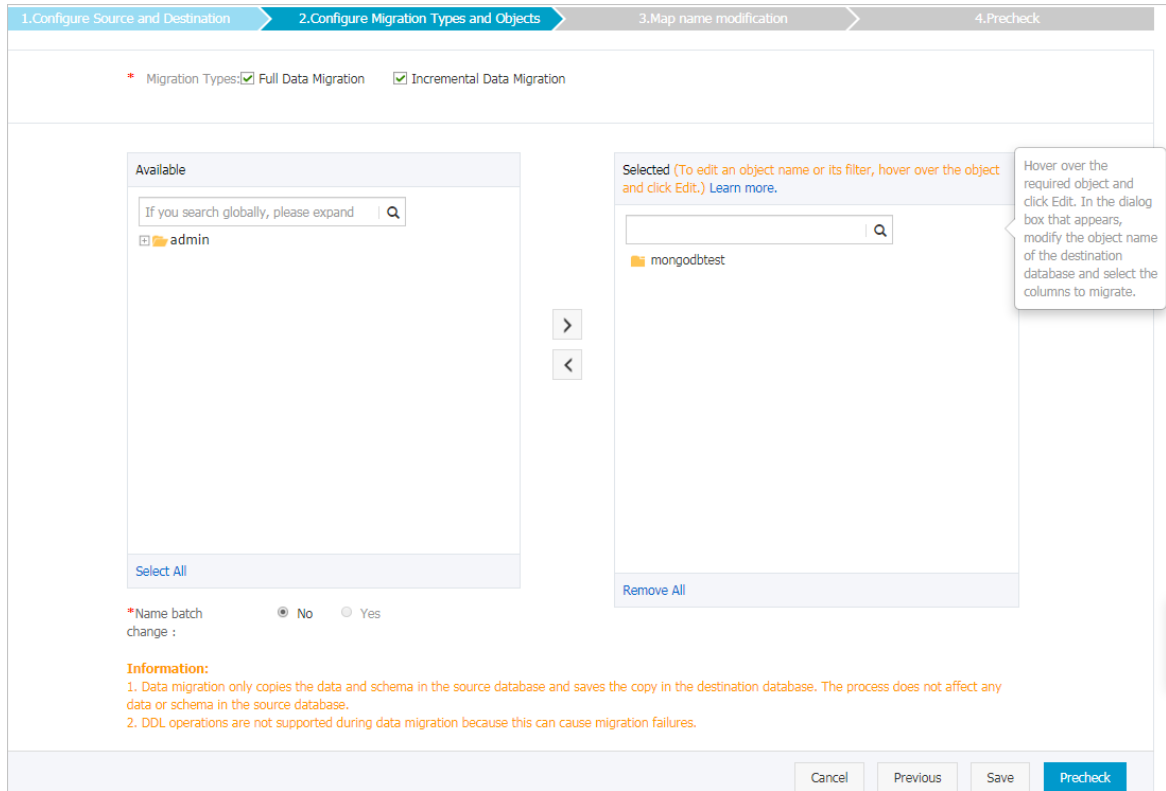| Section	Parameter	Description
N/A	Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select an instance type based on the location where the database is deployed. In this topic, a **User-Created Database with Public IP Address** is used as an example. ⑦ **Note** If you select other instance types, you must prepare the environment that is required for the source database. For more information, see Preparation overview.

Section	Parameter	Description
Source Database	Instance Region	If Instance Type is set to **User-Created Database with Public IP Address**, you do not need to specify the **Instance Region**. ⑦ **Note** If you have configured a whitelist for the user-created MongoDB database, you must add the CIDR blocks of DTS servers to the whitelist. You can click **Get IP Address Segment of DTS** next to **Instance Region** to obtain the CIDR blocks of DTS servers.
	Database Type	Select **MongoDB**.
	Hostname or IP Address	Enter the endpoint of a shard for the source database. In this example, enter the public IP address of the shard. ⑦ **Note** DTS migrates each shard of the source database in turn. In this example, enter the endpoint of the first shard. Then enter the endpoint of the second shard in the second migration task. Repeat this until all shards are migrated.
	Port Number	Enter the service port of the shard. ⑦ **Note** The service port of each shard for user-created MongoDB database must be open to the public network.
	Database Name	Enter the name of the authentication database to which the database account belongs.
	Database Account	Enter the username of the database account used to manage the source database. 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. ⑦ **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 **Failed** message is displayed, click **Check** in the **Failed** message to modify the information as prompted.

Section	Parameter	Description
	Encryption	Select **Non-encrypted**. ② **Note** The **SSL-encrypted** option is available only when you migrate MongoDB Atlas.
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 resides.
	MongoDB Instance ID	Select the ID of the ApsaraDB for MongoDB instance.
	Database Name	Enter the name of the authentication database to which the database account belongs. ② **Note** If you want to use the root account, specify admin for the Database Name parameter.
	Database Account	Enter the username of the database account used to manage the source database. 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. ② **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 of the page, click **Set Whitelist and Next**.

> ② **Note**   The CIDR blocks of DTS servers are automatically added to the whitelist of the destination ApsaraDB for MongoDB instance. This ensures that DTS servers can connect to the destination ApsaraDB for MongoDB instance. After the migration is completed, you can remove these CIDR blocks from the whitelist. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

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

Parameter	Description
Migration Types	○ To perform only full data migration, select **Full Data Migration**. ○ To migrate data with minimal downtime, select both **Full Data Migration** and **Incremental Data Migration**. ⑦ **Note** If the **Incremental Data Migration** option is not selected, do not write new data to the user-created MongoDB database when full data migration is in progress. Otherwise, data inconsistency may occur.
Migration objects	○ Select objects from the **Available** section and click the ＞ icon to move the objects to the **Selected** section. ⑦ **Note** ■ Data in the admin and local databases cannot be migrated. ■ The config database is an internal database. We recommend that you do not migrate data in the config database. ○ A migration object can be a database, collection, or function. ○ By default, the name of an object remains unchanged after migration. You can change the names of the objects in the destination RDS instance by using the object name mapping feature. For more information, see Object name mapping.

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

> ⑦ **Note**
>   ○ A precheck is performed before the migration task starts. 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.

12. Repeat Steps 1 to 11 to create data 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.

      > ⑦ **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.

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

9.4. Migrate user-created databases to Alibaba Cloud by using tools provided by MongoDB

mongodump and mongorestore are both built in MongoDB for backup and restoration. You can install the MongoDB database on a local device or ECS instance and use mongodump and mongorestore to migrate a user-created MongoDB database to an ApsaraDB for MongoDB instance.

We recommend that you use DTS to migrate user-created MongoDB databases to Alibaba Cloud, which ensures data migration without service downtime. For more information, see Migrate user-created standalone MongoDB databases to Alibaba Cloud by using DTS.

For more data migration and synchronization solutions, see Overview.

Prerequisites

- The version of mongodump and mongorestore is the same as that of the user-created MongoDB database. For more information about the installation procedure, visit Install MongoDB at the official MongoDB website.

> ⑦ Note   You can also run the mongodump and mongorestore commands on the server where the user-created MongoDB databases reside.

- Standalone ApsaraDB for MongoDB instances only support MongoDB 3.4. To ensure compatibility, the version of the user-created MongoDB database must be 3.0, 3.2, 3.4, 4.0, or 4.2 .

> ⑦ Note   If the source user-created MongoDB databases and the destination ApsaraDB for MongoDB instance run different database versions or storage engines, ensure that 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 版本及存储引擎.

- The storage space of the standalone ApsaraDB for MongoDB instance must be larger than that required by the user-created MongoDB database. If the storage space is insufficient, you can expand the storage space. For more information, see Configuration change overview.

Precautions

- This is full data migration. To ensure data consistency, stop related services and data writing operations on the source MongoDB database before the migration starts.

- If you have run the mongodump command to back up the database, move the backup files in the *dump* folder to another directory. Ensure that the *dump* folder is empty. Otherwise, the historical backup files in this folder is overwritten during data backup.

- Run the mongodump and mongorestore commands on the server where the user-created MongoDB database resides. Do not run them in the mongo shell.

Step 1: Back up the user-created database

1. On the server where the user-created database resides, run the following command to back up the whole data:

   ```
   mongodump --host <mongodb_host> --port <port> -u <username> --authenticationDatabase <database>
   ```

   > ⑦ Note
   >   ○ <mongodb_host>: the server address of the user-created MongoDB database. If this database is deployed on the current server, set this parameter to 127.0.0.1.
   >   ○ <port>: the service port number for the user-created database. The default port number is 27017.
   >   ○ <username>: the account used to log on to the user-created MongoDB database.
   >   ○ <database>: the name of the authentication database. It is the database where the database account is created.

   Example:

   ```
   mongodump --host 127.0.0.1 --port 27017 -u root --authenticationDatabase admin
   ```

2. Enter the password for the database user in the Enter password: prompt and press Enter to start the backup.

   > ⑦ Note   The password you enter is not displayed.

Wait until the data backup is complete. The data of the user-created database is backed up in the *dump* folder of the current directory.

Step 2: Migrate data to the ApsaraDB for MongoDB instance

1. Obtain the connection address of the primary node of the ApsaraDB for MongoDB instance.

   i. Log on to the ApsaraDB for MongoDB console.

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

   iii. In the left-side navigation pane, click **Replica Set Instances**.

   iv. Find the target instance and click its ID.

   v. In the left-side navigation pane, click **Database Connection** to view the database connection details.
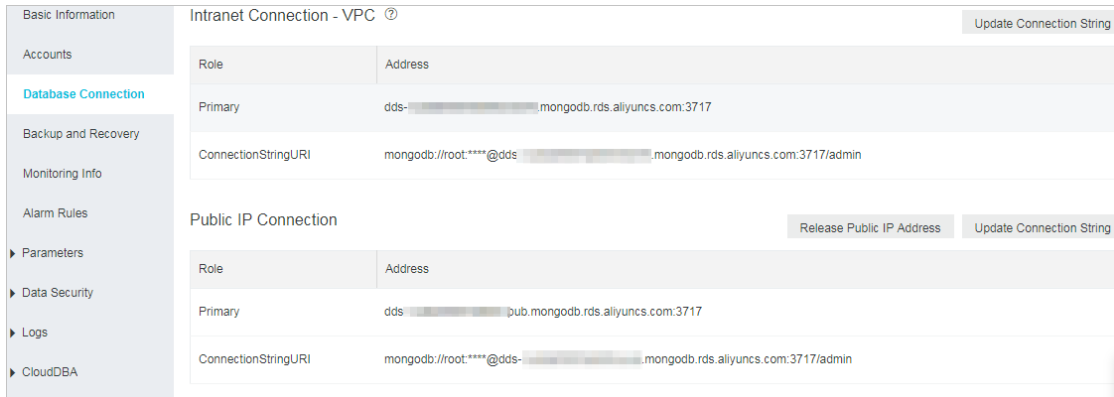

Connection addresses

Address type	Description	Application scenario
VPC connection address	A VPC is an isolated virtual network with better security and performance than a classic network.	The user-created MongoDB database is deployed on the ECS instance. ⑦ **Note** The ECS instance and ApsaraDB for MongoDB instance must be located in the same region and VPC.
Public connection address	By default, ApsaraDB for MongoDB instances do not provide public connection addresses. You need to apply for a public endpoint if required. For more information, see Apply for a public endpoint.	The user-created MongoDB database is deployed on a local device.

2. Add the IP address of the server where the user-created database resides to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

> ⑦ **Note**
> - When you connect to an ApsaraDB for MongoDB instance over a VPC, you must add the internal IP address of the ECS instance where the user-created database resides to the whitelist of the ApsaraDB for MongoDB instance.
> - When you connect to an ApsaraDB for MongoDB instance over the Internet, you must add the public IP address of the local server where the user-created database resides to a whitelist of the ApsaraDB for MongoDB instance.

3. On the server where the user-created database resides, run the following command to migrate the whole data to the ApsaraDB for MongoDB instance:

```
mongorestore --host <Primary_host> -u <username> --authenticationDatabase <database> <Backup directory>
```

> ⑦ **Note**
> - <Primary_host>: the connection address of the primary node in the ApsaraDB for MongoDB instance.
> - <username>: the database account of the ApsaraDB for MongoDB instance. The initial account is root.
> - <database>: the name of the authentication database. It is the database where the database account is created. If the database account is root, enter admin.
> - <Backup directory>: the directory that stores backup files. The default backup directory is *dump*.

Example:

```
mongorestore --host dds-bp**********-pub.mongodb.rds.aliyuncs.com:3717 -u root --authenticationDatabase admin dump
```

4. Enter the password for the database user of the ApsaraDB for MongoDB instance in the `Enter password:` prompt and press Enter to start data migration.

> ⑦ **Note**
> - The password you enter is not displayed.
> - If you forget the password for the root user, you can reset it. For more information, see Set a password.

Wait until the data migration is complete. Switch your business to the ApsaraDB for MongoDB instance during off-peak hours.

References

After the database is migrated to an ApsaraDB for MongoDB instance, you can connect to the database and manage the database and database account.

- Connect to a standalone ApsaraDB for MongoDB instance by using the mongo shell

- Manage MongoDB users though DMS

9.5. Migrate user-created MongoDB databases to Alibaba Cloud by using the built-in commands of MongoDB

This topic describes how to migrate user-created MongoDB databases to Alibaba Cloud by using mongodump and mongorestore, which are both built in MongoDB for backup and restoration. You can install user-created MongoDB databases on a local server or an ECS instance, and use mongodump and mongorestore to migrate these databases to a replica set instance of ApsaraDB for MongoDB.

To avoid service disruption, we recommend that you use DTS to migrate user-created MongoDB databases to Alibaba Cloud. For more information, see Migrate the replica set of a user-created MongoDB database to ApsaraDB for MongoDB by using DTS.

For more information about data migration and synchronization solutions, see Overview.

Prerequisites

- mongodump and mongorestore are installed on a different server from the user-created MongoDB databases, but run the same version as the databases. For more information about the installation procedure, visit Install MongoDB at the official MongoDB website.

  > ⑦ Not e   You can also run the mongodump and mongorestore commands on the server where the user-created MongoDB databases reside.

- The storage capacity of the destination replica set instance is greater than the occupied storage space of the user-created MongoDB databases. If the storage capacity is insufficient, you can upgrade the instance. For more information, see Configuration change overview.

Precautions

- This is full data migration. To ensure data consistency, we recommend that you stop writing data to the user-created MongoDB databases before you migrate data.

- If you have used the mongodump command to back up a user-created MongoDB database, move the backup files in the *dump* folder to another directory and make sure that the *dump* folder is empty. If it is not empty, its historical backup files are overwritten the next time you back up a database.

- Run the mongodump and mongorestore commands on the servers. Do not run these commands in the mongo shell.

Step 1 Back up the user-created MongoDB databases

1. On the server where the user-created MongoDB databases reside, run the following command to back up all the databases:

   ```
   mongodump --host <mongodb_host> --port <port> -u <username> --authenticationDatabase <databa
   se>
   ```

> ⑦ Note
>
> - <mongodb_host>: the address of the server where the user-created MongoDB
>   databases reside. In this case, enter 127.0.0.1.
> - <port>: the service port of the user-created MongoDB databases. The default value is
>   27017.
> - <username>: the username you use to log on to a user-created MongoDB database.
> - <database>: the name of the authentication database. It is the database where the
>   database user is created.

Example:

```
mongodump --host 127.0.0.1 --port 27017 -u root --authenticationDatabase admin
```

2. When Enter password: is displayed, enter the password of the database user and press Enter. The
   data backup operation starts.

> ⑦ Note   The password you enter is not displayed.

Wait until data backup is complete. The data of the user-created MongoDB databases is backed up to
the *dump* folder of the directory where you run this command.

Step 2 Migrate data to the destination replica set instance

1. Obtain the public or internal connection string of the primary node in the destination replica set
   instance. For more information, see Overview of replica set instance connections.

   > ⑦ Note   You must apply for a public endpoint manually. For more information, see Apply
   > for a public endpoint for a replica set instance.

2. Add the IP address of the server where the user-created MongoDB databases reside to a whitelist
   of the destination replica set instance. For more information, see Configure a whitelist for a replica set
   instance.

   > ⑦ Note
   >
   > - If you want to connect to a replica set instance over an internal network, you must add
   >   the private IP address of the ECS instance where the user-created MongoDB databases
   >   reside to a whitelist of the replica set instance.
   > - If you want to connect to a replica set instance over the Internet, you must add the
   >   public IP address of the server where the user-created MongoDB databases reside to a
   >   whitelist of the replica set instance.

3. On the server where the user-created MongoDB databases reside, run the following command to
   restore all the backup files to the destination replica set instance:

   ```
   mongorestore --host <Primary_host>  -u <username> --authenticationDatabase <database> <Backup di
   rectory>
   ```

> **Note**
> - <Primary_host>: the connection string of the primary node in the destination replica set instance.
> - <username>: the username you use to log on to a database of the destination replica set instance. The initial username is root.
> - <database>: the name of the authentication database. It is the database where the database user is created. If the database username is root, enter admin.
> - <Backup directory>: the directory where the backup files are stored. The default value is *dump*.

Example:

```
mongorestore --host dds-bp**********-pub.mongodb.rds.aliyuncs.com:3717 -u root --authenticationD
atabase admin dump
```

4. When `Enter password:` is displayed, enter the password of the database user and press Enter. The data restoration operation starts.

> **Note**
> - The password you enter is not displayed.
> - If you forget the password of the root user, you can reset it. For more information, see Set a password for a replica set instance.

After data restoration is complete, switch over your business to the destination replica set instance. We recommend you perform the switchover during off-peak hours to minimize impact on your business.

9.6. Migrate a user-created MongoDB database to ApsaraDB for MongoDB by using tools provided by MongoDB

This topic describes how to migrate a user-created MongoDB database to ApsaraDB for MongoDB by using mongodump and mongorestore, which are built in MongoDB for data backup and restoration. You can install user-created MongoDB databases on a local server or an ECS instance, and use mongodump and mongorestore to migrate these databases to an ApsaraDB for MongoDB sharded cluster instance.

Background information

- To avoid service disruption, we recommend that you use DTS to migrate user-created sharded MongoDB databases to ApsaraDB for MongoDB. For more information, see Migrate a user-created sharded MongoDB database to ApsaraDB for MongoDB by using DTS.
- For more information about data migration and synchronization solutions, see Overview of data migration and synchronization.

Prerequisites

- mongodump and mongorestore are installed on a different server from the user-created MongoDB databases, but run the same version as the databases. For more information about the installation procedure, visit Install MongoDB at the MongoDB official website.

  > Note   You can also run the mongodump and mongorestore commands on the server where the user-created MongoDB databases reside.

- The storage capacity of the destination sharded cluster instance is greater than the storage space occupied by the user-created MongoDB databases. If the storage capacity is insufficient, you can upgrade the instance. For more information, see Configuration change overview.

Precautions

- This is full data migration. To ensure data consistency, we recommend that you stop writing data to the user-created MongoDB databases before you migrate data.

- If you have run the mongodump command to back up a user-created MongoDB database, move the backup files in the *dump* folder to another directory and make sure that the *dump* folder is empty. If it is not empty, its historical backup files are overwritten the next time you back up a database.

- Run the mongodump and mongorestore commands on the servers. Do not run these commands on the mongo shell.

Step 1: Back up the user-created MongoDB databases

1. On the server where the user-created MongoDB databases reside, run the following command to back up all the databases:

   ```
   mongodump --host <mongodb_host> --port <port> -u <username> --authenticationDatabase <database>
   ```

   > Note
   > - <mongodb_host>: the IP address of the server where the user-created MongoDB databases reside. In this case, enter 127.0.0.1.
   > - <port>: the service port of the user-created MongoDB databases. The default value is 27017.
   > - <username>: the username used to log on to a user-created MongoDB database.
   > - <database>: the database corresponding to the username if authentication is enabled.

   Example:

   ```
   mongodump --host 127.0.0.1 --port 27017 -u root --authenticationDatabase admin
   ```

2. When Enter password: is displayed, enter the password of the database user and press Enter. The data backup operation starts.

Wait until data backup is complete. The data of the user-created MongoDB databases is backed up to the *dump* folder of the directory where you run this command.

Step 2: (Optional) Configure data sharding

If data sharding is not configured, data is only written to the primary shard. Then the storage and computing resources of other shards are not used. For more information, see Configure sharding to maximize the performance of shards.

> ⓘ **Note** You must create required databases and collections in the destination sharded cluster instance before data migration. However, you can configure data sharding for the databases and collections either before or after data migration.

Step 3: Migrate data to the destination sharded cluster instance

1. Obtain the public or internal connection string of a mongos in the destination sharded cluster instance. For more information, see Overview of sharded cluster instance connections.

   > ⓘ **Note** You must apply for a public endpoint manually. For more information, see Apply for a public endpoint for a sharded cluster instance.

2. Add the IP address of the server where the user-created MongoDB databases reside to a whitelist of the destination sharded cluster instance. For more information, see Configure a whitelist for a sharded cluster instance.

   > ⓘ **Note**
   > - If you want to connect to a sharded cluster instance over the internal network, you must add the private IP address of the ECS instance where the user-created MongoDB databases reside to a whitelist of the sharded cluster instance.
   > - If you want to connect to a sharded cluster instance over the Internet, you must add the public IP address of the server where the user-created MongoDB databases reside to a whitelist of the sharded cluster instance.

3. On the server where the user-created MongoDB databases reside, run the following command to restore all the backup files to the destination sharded cluster instance:

   ```
   mongorestore --host <Mongos_host> -u <username> --authenticationDatabase <database> <Backup directory>
   ```

   > ⓘ **Note**
   > - <Mongos_host>: the connection string of any mongos in the ApsaraDB for MongoDB instance.
   > - <username>: the username used to log on to a database of the destination sharded cluster instance. The initial username is root.
   > - <database>: the database corresponding to the username if authentication is enabled. If the username is root, enter admin.
   > - <Backup directory>: the directory where the backup files are stored. The default value is *dump*.

   Example:

```
mongorestore --host s-bp**********-pub.mongodb.rds.aliyuncs.com:3717 -u root --authenticationData
base admin dump
```

4. When `Enter password:` is displayed, enter the password of the database user and press Enter. The
   data restoration operation starts.

> ⑦ Note
> - The password characters are not displayed when you enter the password.
> - If you forget the password of the root user, you can reset it. For more information, see
>   Set a password for a sharded cluster instance.

After data restoration is complete, switch over your business to the destination sharded cluster
instance. We recommend you perform the switchover during off-peak hours to minimize impact on your
business.