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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent
ial, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document.
# Document conventions

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal disclaimer</td>
<td>I</td>
</tr>
<tr>
<td>Document conventions</td>
<td>I</td>
</tr>
<tr>
<td>1 Before you start</td>
<td>1</td>
</tr>
<tr>
<td>2 ApsaraDB for MongoDB console</td>
<td>3</td>
</tr>
<tr>
<td>3 Get started with a sharded cluster instance</td>
<td>7</td>
</tr>
<tr>
<td>4 Create a sharded cluster instance</td>
<td>8</td>
</tr>
<tr>
<td>5 Set a password for a sharded cluster instance</td>
<td>15</td>
</tr>
<tr>
<td>6 Configure a whitelist for a sharded cluster instance</td>
<td>17</td>
</tr>
<tr>
<td>7 Apply for a public endpoint for a sharded cluster instance</td>
<td>21</td>
</tr>
<tr>
<td>8 Connect to an instance</td>
<td>25</td>
</tr>
<tr>
<td>8.1 Overview of sharded cluster instance connections</td>
<td>25</td>
</tr>
<tr>
<td>8.2 Connect to a sharded cluster instance by using the mongo shell</td>
<td>28</td>
</tr>
<tr>
<td>8.3 Connect to an ApsaraDB for MongoDB instance through the program code</td>
<td>30</td>
</tr>
<tr>
<td>9 Migrate data</td>
<td>36</td>
</tr>
<tr>
<td>9.1 Migrate user-created MongoDB databases to Alibaba Cloud by using DTS</td>
<td>36</td>
</tr>
<tr>
<td>9.2 Migrate user-created MongoDB databases to Alibaba Cloud by using the built-in commands of MongoDB</td>
<td>47</td>
</tr>
</tbody>
</table>
You can migrate data from a user-created MongoDB database to an ApsaraDB for MongoDB instance. Please pay close attention to the limits of ApsaraDB for MongoDB.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy a sharded cluster instance</td>
<td>The database version must match the storage engine. For more information, see #unique_4.</td>
</tr>
</tbody>
</table>
| Build cluster components      | · When you create a sharded cluster instance, you can specify the specifications and numbers of mongos and shards.  
                                · While the instance is running, you can add mongos and shards, but you cannot remove them. For more information, see #unique_5.                                                                                                                                               |
| Restart a sharded cluster instance | You must restart the instance in the ApsaraDB for MongoDB console or by calling the #unique_6 operation.                                                                                                                                                                                                                             |
| Migrate data from a sharded cluster instance | You can use the built-in commands of MongoDB or Data Transmission Service (DTS) to migrate data. For more information, see Use the built-in commands of MongoDB to migrate data of a sharded cluster instance or Use DTS to migrate data of a sharded cluster instance.                                                                                      |
| Back up the data of a sharded cluster instance | · Configure automatic backup for an ApsaraDB for MongoDB instance: Only physical backup is supported.  
                                · #unique_10: Both physical backup and logical backup are supported.                                                                                                                                                                                                     |

**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. For more information, see #unique_11. Alternatively, you can select the database version 4.0 when you create the instance.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore the data of a sharded cluster instance</td>
<td>You can only restore data from a point in time. For more information, see #unique_12.</td>
</tr>
<tr>
<td>Modify parameters of a sharded cluster instance</td>
<td>For security and stability, you are not allowed to modify the parameters of a sharded cluster instance.</td>
</tr>
</tbody>
</table>
2 ApsaraDB for MongoDB console

The ApsaraDB for MongoDB console is a Web application for managing MongoDB 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 console. For more information about common settings and basic operations in the Alibaba Cloud console, see Alibaba Cloud console.

Prerequisites

Use your Alibaba Cloud account to log on to the ApsaraDB for MongoDB console. If you do not have an Alibaba Cloud account, click Register.

Homepage

The console homepage displays the same information for all MongoDB sharded cluster instances.

Log on to the ApsaraDB for MongoDB console and go to the 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
### ApsaraDB for MongoDB console

<table>
<thead>
<tr>
<th>No.</th>
<th>UI element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sharding Instances</td>
<td>The ApsaraDB for MongoDB console homepage, which displays all instances in a region that belong to the current account.</td>
</tr>
<tr>
<td>2</td>
<td>Region</td>
<td>You can click a region to display all instances that reside within the region.</td>
</tr>
<tr>
<td>3</td>
<td>Refresh</td>
<td>The button to refresh the instance information page.</td>
</tr>
<tr>
<td>4</td>
<td>Create Instance</td>
<td>The button to create a new instance.</td>
</tr>
</tbody>
</table>
| 5   | Instance ID         | • You can click an instance ID to go to the Basic Information page of the instance.  
                             • You can click the icon following an instance ID to modify the name of the instance. |
| 6   | Running Status      | The status of the instance. Instances may be in different states.           |
| 7   | Management icon     | You can click this icon to manage, restart, or release an instance.        |
| 8   | Export              | #unique_14.                                                                  |

**MongoDB instance console**

Log on to the ApsaraDB for MongoDB console. Click the management icon in the Actions column corresponding to an Instance ID and choose Manage from the shortcut menu. The Basic Information page is displayed. The following table lists the parameters on the page.

<table>
<thead>
<tr>
<th>Page name</th>
<th>Area</th>
<th>Description</th>
<th>Link of common operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation area in the window</td>
<td>-</td>
<td>You can back up and restart instances.</td>
<td>• Back up an instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Restart an instance</td>
</tr>
<tr>
<td>Basic Information</td>
<td>Basic Information</td>
<td>You can view the basic information of an instance, such as its ID, region, network type, and storage engine.</td>
<td></td>
</tr>
<tr>
<td>Page name</td>
<td>Area</td>
<td>Description</td>
<td>Link of common operation</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Specifications</td>
<td></td>
<td>You can view instance specifications such as the database version, maintenance period, billing method, creation time, and expiration time.</td>
<td>Specify a maintenance period</td>
</tr>
</tbody>
</table>
| Mongos Nodes or Shard Nodes |                          | · In the mongos list, you can locate a mongos ID and click the icon to change its configurations, and log on to or restart the mongos.  
· In the shard list, you can locate a shard ID and click the icon to log on to a shard, restart the shard, trigger a failover, or change the shard configurations.  
· Database read and write operations may fail when you restart nodes. We recommend that you do not perform the Create, Retrieve, Update, and Delete (CRUD) operations on databases when you restart nodes. | · Trigger a failover  
· Change configurations  
· Log on to a database  
· Restart a node  
· #unique_20                  |
<p>| Backup and Restore |                          | 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.                                                                                                                                                                                                                           | Create an instance at a specified time        |
| Monitoring information |                          | You can view monitoring information of mongos or shards for a specified time period based on specified metrics.                                                                                                                                                                                                                                         | -                                              |
| Security Control Whitelist Settings |                          | You can configure an IP whitelist.                                                                                                                                                                                                                                                                                                                  | Configure a whitelist.                        |</p>
<table>
<thead>
<tr>
<th>Page name</th>
<th>Area</th>
<th>Description</th>
<th>Link of common operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Log</td>
<td></td>
<td>MongoDB audit logs record all operations that you perform on a database. You can use these logs in analysis.</td>
<td>View audit logs.</td>
</tr>
</tbody>
</table>
3 Get started with a sharded cluster instance

This topic describes how to get started with a sharded cluster 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. Create a sharded cluster instance.
2. Set a password for a sharded cluster instance.
3. Configure a whitelist for a sharded cluster instance.
4. Apply for a public endpoint for a sharded cluster instance.
5. Connect to a sharded cluster instance. For more information, see Overview of sharded cluster instance connections.
4 Create a sharded cluster instance

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

Prerequisites

- An Alibaba Cloud account is registered. For more information, see Sign up with Alibaba Cloud.
- Your account balance is sufficient if you want to create a pay-as-you-go instance.

Billing

For more information, see Billing items and pricing.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click Sharding Instances.
3. On the Sharding Instances page, click Create Instance.
4. Click Subscription(Sharding) or Pay-As-You-Go(Sharding).

Note:

- Subscription: You must pay the subscription fee when you create an instance. We recommend that you select this billing method for long-term use, because it is more cost-effective than pay-as-you-go billing. Longer subscription periods have larger discounts.
- Pay-as-you-go: A pay-as-you-go instance is charged at an hourly rate based on your actual resource usage. We recommend that you select this billing method for short-term use. You can reduce costs by releasing your pay-as-you-go instance after you no longer need it.
5. Configure the instance.

Table 4-1: Parameters for creating a sharded cluster instance

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Configuration</td>
<td>Region</td>
<td>The region where the sharded cluster instance is deployed. After an instance is created, you cannot change its region. Exercise caution when you select the region. Only instances in the same region (for example, an ECS instance and a sharded cluster instance) can communicate with each other inside an internal network.</td>
</tr>
<tr>
<td>Zone</td>
<td>Zones</td>
<td>Zones are geographic areas in a region with independent power grids and networks. An ECS instance and a sharded cluster instance in the same zone can be interconnected over an internal network with the minimum network latency.</td>
</tr>
</tbody>
</table>

Note:
To implement zone-disaster recovery, you can deploy the sharded cluster instance across multiple zones. For more information, see #unique_29.
### Database Version

The version of the database engine for the sharded cluster instance. ApsaraDB for MongoDB supports 3.2, 3.4, and 4.0.

We recommend that you select MongoDB 3.2 or later. For more information, see [Versions and storage engines](#).

**Note:**
You can manually upgrade the database version while an instance is running. For more information, see [Upgrade the database version of an ApsaraDB for MongoDB instance](#).

### Storage Engine

The storage engine for a sharded cluster instance is WiredTiger, which cannot be changed.

### Network Type

**Classic**

Cloud services on a classic network are not isolated. Unauthorized access can only be blocked by using security groups or whitelists.
<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VPC</td>
<td>A VPC is an isolated virtual network with better security and performance than a classic network. We recommend that you select the VPC network type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You must create a VPC before you create an instance. For more information, see Create a VPC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You can change the network type after you create an instance. For more information, see Switch the network type of an ApsaraDB for MongoDB instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If you want to migrate your applications to the cloud, you can build a virtual data center by connecting your on-premises data center to the resources in a VPC through a leased line or a virtual private network (VPN). For more information, see Configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC.</td>
</tr>
<tr>
<td></td>
<td>Mongos</td>
<td>The specifications of each mongos. For more information, see Instance specifications.</td>
</tr>
<tr>
<td>Specification</td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can add new mongos or change the configurations of existing mongos while a sharded cluster instance is running.</td>
</tr>
<tr>
<td></td>
<td>Quantity</td>
<td>The number of mongos.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A sharded cluster instance can contain 2 to 32 mongos.</td>
</tr>
<tr>
<td>Shard</td>
<td>Specification</td>
<td>The specifications of each shard. For more information, see Instance specifications.</td>
</tr>
<tr>
<td>Specifications</td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can add new shards or change the configurations of existing shards while a sharded cluster instance is running.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Storage Space</td>
<td>Storage Space</td>
<td>The storage space of each shard. Value range: 10 GB to 1,000 GB.</td>
</tr>
<tr>
<td></td>
<td>Note: Shards are used to store data file, system files, and log files.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quantity</td>
<td>The number of shards.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A sharded cluster instance can contain 2 to 32 shards.</td>
</tr>
<tr>
<td>Configserver Specifications</td>
<td>Specification</td>
<td>The specifications of each config server: 1 core, 2 GB of memory, and 20 GB of storage space. The config server specifications are not user-configurable.</td>
</tr>
<tr>
<td>Set Password</td>
<td>· Set Now</td>
<td>The password of the root user. You can set the password when you create the instance or while the instance is running. For more information, see</td>
</tr>
<tr>
<td></td>
<td>· Set Later</td>
<td>Set a password for a sharded cluster instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· The password must be 8 to 32 characters in length.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>· The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters . Special characters include ! # $ % ^ &amp; * ( ) _ + - =</td>
</tr>
<tr>
<td>Validity</td>
<td>N/A</td>
<td>You must specify this parameter when you create a sharded cluster instance that uses subscription billing. You can select one to nine months for a monthly subscription or one to three years for an annual subscription.</td>
</tr>
</tbody>
</table>

6. Click Buy Now.

7. On the Confirm Order page, read and select ApsaraDB for MongoDB Agreement of Service, and complete the payment as prompted.
View the created instance

1. Log on to the ApsaraDB for MongoDB console.

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

3. In the left-side navigation pane, click Sharding Instances.

Troubleshoot if you cannot find the instance

<table>
<thead>
<tr>
<th>Possible cause</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>You selected the wrong region in the console.</td>
<td>Select the region where the instance is deployed. For more information, see View the created instance.</td>
</tr>
<tr>
<td>You opened the Replica Set Instances page.</td>
<td>In the left-side navigation pane, click Sharding Instances. For more information, see View the created instance.</td>
</tr>
<tr>
<td>The instance list in the ApsaraDB for MongoDB console was not updated or was updated before the instance was created.</td>
<td>Wait several minutes and then update the instance list to check whether the instance is added to the list.</td>
</tr>
</tbody>
</table>
### Possible cause

<table>
<thead>
<tr>
<th>Resources are insufficient.</th>
</tr>
</thead>
</table>

### Troubleshooting

The system may fail to create the instance due to insufficient resources. In this case, your payment is refunded. You can check the refund on the *Orders* page.

After you confirm the refunded fees, you can try to create the instance in another zone. You can also submit a ticket to process the failure.

---

**What's next**

After you create a sharded cluster instance, you must configure a whitelist. For more information, see *Configure a whitelist for a sharded cluster instance*. If you want to connect to the instance over the Internet, you must apply for a public endpoint. For more information, see *Apply for a public endpoint for a sharded cluster instance*.

For more information about instance connection methods and connection scenarios, see *#unique_33*. 
5 Set a password for a sharded cluster instance

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click Accounts.
6. Click Reset Password.
7. In the Reset Password dialog box that appears, enter and confirm the new password. Click OK.

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 sharded cluster instance

This topic describes how to configure a whitelist for a sharded cluster 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 region where the target instance resides.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose Data Security > Whitelist Setting.
6. Click the icon in the Operation 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.

**Related operations**

- #unique_34/unique_34_Connect_42_section_fwu_oit_4dc
- #unique_34/unique_34_Connect_42_section_1si_mlru_q72
Common connection scenarios

- #unique_35
- #unique_36
- #unique_37
- #unique_38
7 Apply for a public endpoint for a sharded cluster instance

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

Context

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

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intranet Connection - VPC</strong></td>
<td>• A VPC is an isolated network with higher security and performance than a classic network.</td>
</tr>
<tr>
<td></td>
<td>• By default, ApsaraDB for MongoDB provides endpoints on a VPC.</td>
</tr>
<tr>
<td><strong>Intranet Connection - Classic Network</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Public IP Connection</strong></td>
<td>• Connecting to a replica set instance over the Internet is risky. Therefore, ApsaraDB for MongoDB does not provide public endpoints.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click Database Connection.
6. In the upper-right corner of the Public IP Connection section, click Apply for Public Connection String.

7. In the dialog box that appears, specify Node Type and Node ID, and click OK.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Type</td>
<td>shard</td>
<td>A shard. Before you apply for a public endpoint for a shard, you must apply for an internal endpoint for it. For more information, see #unique_39. If you want to read the oplog data of a shard over the Internet when you perform certain operations such as data synchronization between clusters, you must apply for a public endpoint for the shard.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>cs</td>
<td>The config server. Before you apply for a public endpoint for the config server, you must apply for an internal endpoint for it. For more information, see #unique_39. If you want to read the configuration information of the config server over the Internet when you perform certain operations such as data synchronization between clusters, you must apply for a public endpoint for the config server.</td>
<td></td>
</tr>
<tr>
<td>mongos</td>
<td>A mongos. This is the default option because your application is connected to a mongos in most cases.</td>
<td></td>
</tr>
<tr>
<td>Node ID</td>
<td>The ID of the component for which you want to apply for a public endpoint.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:**
- For more information about component types, see #unique_40.
- To apply for a public endpoint for other mongos, repeat this step. You can only apply for a new public endpoint after the current one is created.

**Results**

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

**References**
- #unique_35.
- To ensure data security, we recommend that you release a public endpoint if you no longer need it. For more information, see #unique_41.

**What's next**
Before you connect to the sharded cluster instance by using one of the obtained public endpoints, you must add the public IP address of your client to a whitelist of the instance. For more information, see Configure a whitelist for a sharded cluster instance.
8 Connect to an instance

8.1 Overview of sharded cluster instance connections

This topic provides an overview of sharded cluster instance connections. ApsaraDB for MongoDB supports both connection strings and connection string URIs. You can use a connection string to connect to one mongos, and use a connection string URI to connect to more than one mongos. For high availability, we recommend that you use connection string URIs to connect your application to more than one mongos.

View connection addresses

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, click Database Connection to view connection addresses.
### Introduction to connection addresses

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Address type** | • 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.  
  • 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.  
  • Public IP Connection: Connecting to a sharded cluster instance over the Internet is risky. Therefore, ApsaraDB for MongoDB does not provide public endpoints. If you want to connect to a sharded cluster instance from a device outside Alibaba Cloud (for example, a local client), you must apply for a public endpoint. For more information, see [Apply for a public endpoint for a sharded cluster instance](#). |
| mongos ID    | The connection string of a mongos is in the following format:                                                                                                                                                |
|              |  
  <host>:<port>                                                                                                                                                                                            |
|              |  
  • <host>: the endpoint you use to connect to the sharded cluster instance.  
  • <port>: the port you use to connect to the sharded cluster instance.  
  
  **Note:**  
  During regular tests, you can use a connection string to directly connect to a mongos.  
  

A connection string URI is in the following format:

```
mongodb://[username:password@]host1[:port1][,host2[:port2 ],...[,hostN[:portN]]][/][database][? options]
```

- `mongodb://`: the prefix, which indicates that the connection address is a connection string URI.
- `username:password@`: the username and password you use to log on to a database of the sharded cluster instance. You must separate them with a colon (:).
- `hostX:portX`: the endpoint and port of a mongos in the sharded cluster instance.
- `/database`: the name of the authentication database. It is the database where the database user is created.
- `? options`: additional connection options.

**Note:**

If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, your client can automatically distribute your requests to multiple mongos to balance loads. When a mongos fails, your client automatically redirects requests to other mongos in the normal state.

Log on to a database of the sharded cluster instance

1. Obtain the preceding connection addresses and the following information:

   - The username you use to log on to the database. The initial username is root.

   **Note:**

   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 based on your needs. For more information, see [Manage MongoDB users through DMS](#).

   - The password of the database user. 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](#).

   - The name of the authentication database. It is the database where the database user is created. If the username is root, enter admin.
2. Log on to the database.

   • Connect to a sharded cluster instance by using the mongo shell
   • Connect to an ApsaraDB for MongoDB instance through the program code

Common connection scenarios

   • #unique_35
   • #unique_36
   • #unique_37
   • #unique_38

FAQ

   • How to troubleshoot logon issues for the mongo shell
   • #unique_48
   • How to troubleshoot the high CPU utilization of ApsaraDB for MongoDB
   • How to query and limit the number of connections

8.2 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 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 sharded cluster instance. For more information, see Configure a whitelist for a sharded cluster instance.

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 region where the target instance resides.

3. In the left-side navigation pane, click Sharding 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 mongos.

![Connection addresses of a mongos](image)

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>
   ```

**Note:**

- `<mongos_host>`: the connection string of a mongos in the sharded cluster instance.
- `<username>`: the username you use 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 users and grant permissions to them as needed. For more information, see [#unique_43](#).
• `<database>`: the name of the authentication database. It is the database where the database user is created. 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.

Note:
The password you enter is not displayed.

Common connection scenarios

• #unique_35
• #unique_36
• #unique_37
• #unique_38

FAQ

• How to troubleshoot logon issues for the mongo shell
• #unique_48
• How to troubleshoot the high CPU utilization of ApsaraDB for MongoDB
• How to query and limit the number of connections

8.3 Connect to an ApsaraDB for MongoDB instance through the program code

Related links

• MongoDB Drivers
• Connection String URI Format

Note:
The connection sample code in this topic applies when you use internal IP addresses provided by Alibaba Cloud to connect to ApsaraDB for MongoDB.
For more information about how to obtain connection strings of ApsaraDB for MongoDB, see Connect to an ApsaraDB for MongoDB instance.

Node.js

Related links: MongoDB Node.js Driver

1. Initialize a project.

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

2. Install the driver package and toolkit.

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

3. Obtain connection strings of ApsaraDB for MongoDB instances.

4. Use the following Node.js sample code.

```javascript
'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, replSetName);
console.info("url:", url);
// Obtain the MongoClient.
mongoClient.connect(url, function(err, db) {
  if(err) {
    console.error("connect err:", err);
    return 1;
  }
  // Authenticate. Here, the username is for authentication of 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 docName = "NODE:" + uuid.v1();
    var doc = {"DEMO": docName, "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();
  });
});
```
$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 connection strings of ApsaraDB for MongoDB instances.
2. Use the following Java sample code.

    Maven configuration

    ```
    <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.rdstest.aliyun-inc.com", 27017);
        public static ServerAddress seed2 = new ServerAddress("demotest-2.mongodb.tbc3.newtest.rdstest.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";
        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. The default parameter settings are sufficient for most scenarios.
    MongoClientOptions options = MongoClientOptions.builder()
        .requiredReplicaSetName(ReplSetName).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]]][/[/database][? options]]
    MongoClientURI connectionString = new MongoClientURI("mongodb://" + username + ":" + password + "@" + 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());
        }
    }
}
```python
} finally {
    // Close the client and release resources.
    client.close();
}
return ;
}

Python

Related links:

- Pymongo download
- Official documentation

1. Install PyMongo.

```python
pip install pymongo
```n
2. Obtain the connection strings of ApsaraDB for MongoDB instances.
3. Use the following Python sample code.

```python
import uuid
from pymongo import MongoClient

# Specify two addresses used to connect 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. Here, the username is for authentication of 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 based 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
```
9 Migrate data

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

This topic describes how to migrate each shard of user-created MongoDB databases to Alibaba Cloud by using Data Transmission Service (DTS). DTS allows you to fully and incrementally migrate data without interruptions to your applications.

To avoid service disruption, we recommend that you use DTS to migrate user-created MongoDB databases to Alibaba Cloud. You can also use the built-in commands of MongoDB in this situation. For more information, see Migrate user-created MongoDB databases to Alibaba Cloud by using the built-in commands of MongoDB.

For more information about data migration and synchronization solutions, see #unique_53.

Prerequisites

- The service port of each shard of the user-created MongoDB databases is accessible over the Internet.
- The version of the user-created MongoDB databases is 3.0, 3.2, 3.4, 3.6, or 4.0.
- Each shard in the destination sharded cluster instance has sufficient storage space.

Note:

For example, the user-created MongoDB databases have three shards, and the second shard occupies the most storage space (500 GB). In this case, the storage space of each shard in the destination sharded cluster instance must be greater than 500 GB.

How it works

DTS migrates each shard of a sharded cluster instance in turn. You must create a data migration task for each shard.

Note:
The distribution of migrated data in the destination sharded cluster instance depends on the shard key you set. For more information, see Configure sharding to maximize the performance of shards.

Precautions

- We recommend that you migrate the user-created MongoDB databases during off-peak hours to avoid business interruptions.
- If the source user-created MongoDB databases and the destination sharded cluster 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 #unique_4.
Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Link configuration fee</th>
<th>Internet traffic fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full data migration</td>
<td>Free of charge</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see Data Transmission Service Pricing.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see Data Transmission Service Pricing.</td>
<td>For more information, see Data Transmission Service Pricing.</td>
</tr>
</tbody>
</table>

Migration types

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

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

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

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

Required database account permissions

<table>
<thead>
<tr>
<th>Data source</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source user-created MongoDB database</td>
<td>Read permissions on the source database</td>
<td>Read permissions on the source database, admin database, and local database</td>
</tr>
<tr>
<td>Destination sharded cluster instance of ApsaraDB for MongoDB</td>
<td>Read/write permissions on the destination database</td>
<td>Read/write permissions on the destination database</td>
</tr>
</tbody>
</table>

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

- For user-created MongoDB databases, visit Create User in MongoDB.
• For a sharded cluster instance of ApsaraDB for MongoDB, see Manage MongoDB users through DMS.

Preparations

1. Disable the balancer of the user-created MongoDB databases. For more information, see Manage the ApsaraDB for MongoDB balancer.

2. Delete orphaned documents generated upon chunk migration failures in the user-created MongoDB databases.

   Note:
   If you do not delete orphaned documents, there may be documents with _id conflicts during the migration process and unwanted data may be migrated.

   a. Download the cleanupOrphaned.js file.

      wget "http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/120562/cn_zh/1564451237979/cleanupOrphaned.js"

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

   Note:
If you want to perform this operation in more than one database, repeat this and the next steps.

```javascript
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);
  });
}
```

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

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

Note:
You must repeat this step on each shard.

Note:
- `<Shardhost>`: the IP address of the shard.
- `<Primaryport>`: the service port of the primary node of the shard.
- `<database>`: the name of the authentication database. It is the database where the database user is created.
- `<username>`: the username you use to log on to a user-created MongoDB database.
• `<password>`: the password of the account with the username mentioned above.

Example:

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

```shell
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.

Note:

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

Procedure

1. Log on to the DTS console.

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

3. In the Migration Tasks section, select the region where the destination sharded cluster instance resides.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>N/A</td>
<td>DTS automatically generates a task name. We recommend that you specify your own task name that helps identify the task. Task names do not need to be unique.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select User-Created Database with Public IP Address.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>If you set Instance Type to User-Created Database with Public IP Address, the system automatically specifies Instance Region.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If the source database has a whitelist, you must click Get IP Address Segment of DTS next to Instance Region to obtain the Classless Inter-Domain Routing (CIDR) block of DTS servers, and add it to the whitelist.</td>
</tr>
<tr>
<td></td>
<td>Database Type</td>
<td>Select MongoDB.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Hostname or IP Address</td>
<td>Enter the endpoint or IP address of a single shard of the source database. For this example, enter the public IP address of the shard.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> DTS migrates each shard of the source database in turn. For this example, enter the endpoint or IP address of the first shard, and then enter that of the second shard in the second migration task. Repeat this until all shards are migrated.</td>
</tr>
<tr>
<td></td>
<td>Port Number</td>
<td>Enter the service port of the shard.</td>
</tr>
<tr>
<td></td>
<td>Database Name</td>
<td>Enter the name of the authentication database. It is the database where the database account is created.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the username of the database account you use to manage the source database. For more information about the account permission requirements, see Required database account permissions.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password of the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After you specify the source database information, click Test Connectivity next to Database Password to check whether the information is correct. If the information is correct, the Passed message is displayed. If the information is incorrect, the Failed message is displayed, and you must click Check next to the Failed message to modify the information as prompted.</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select MongoDB Instance.</td>
</tr>
<tr>
<td>Instance Region</td>
<td></td>
<td>Select the region where the destination sharded cluster instance resides.</td>
</tr>
<tr>
<td>MongoDB Instance ID</td>
<td></td>
<td>Select the ID of the destination sharded cluster instance.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Name</td>
<td>Enter the name of the authentication database. It is the database where the database account is created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="Note:" /> If the database account is root, enter admin.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the username of the database account you use to manage the destination database. For more information about the account permission requirements, see <a href="#">Required database account permissions</a>.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password of the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="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.</td>
</tr>
</tbody>
</table>

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

![Note:](image) The IP addresses of DTS servers are automatically added to a whitelist of the destination sharded cluster instance. This ensures that the DTS servers can connect to this instance. After the migration is complete, you can remove these IP addresses from the whitelist. For more information, see [Configure a whitelist for a sharded cluster instance](#).
7. Configure migration types and migration objects.

Parameter | Description
---|---
Migration Types | If you want to migrate all data, select Full Data Migration.

Note:
To ensure data consistency, do not write data to the user-created MongoDB database while full data migration is in progress.

- If you want to migrate data without interruptions to your business, select both Full Data Migration and Incremental Data Migration.
## Parameter Description

<table>
<thead>
<tr>
<th>Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>• In the Available section, select the objects you want to migrate and</td>
</tr>
<tr>
<td>then click the <img src="image" alt="icon" /> icon to move them to the Selected section.</td>
</tr>
</tbody>
</table>

**Note:**

- Data in the admin database cannot be migrated even if this database is selected.
- The config database is an internal database. Do not migrate data in this database unless otherwise specified.

• A migration object can be a database, collection, or function.
• By default, the name of an object remains unchanged after migration. If you want a different object name after migration, use the object name mapping feature provided by DTS. For more information, see [Object name mapping](#).

8. In the lower-right corner, click Precheck.

**Note:**

• A precheck is performed before the migration task starts. The migration task starts only after the precheck succeeds.
• If the precheck fails, click the ![icon](image) icon for each failed check item to view their details. Perform a precheck again after the failures are fixed.

9. After the precheck succeeds, click Next.


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

12. Repeat Steps 1 to 11 to create migration tasks for the shards that remain.
13. End the migration task.

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

- **Incremental data migration**
  
  An incremental data migration task does not automatically end. You must manually end the task.

**Note:**

Select an appropriate time to manually end a migration task. For example, you can end the migration task during off-peak hours or before you switch over your business to the destination sharded cluster instance.

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

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

14. Switch over your business to the destination sharded cluster instance.

9.2 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 sharded cluster 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 user-
created MongoDB databases to Alibaba Cloud by using DTS*.

For more information about data migration and synchronization solutions, see
#unique_53.

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.

**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 occupied storage space of the user-created MongoDB databases. If the storage
capacity is insufficient, you can upgrade the instance. For more information, see
#unique_5.

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 <database>
   ```

   **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.

   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. 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](image)
   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](image)
   - If you want to connect to a sharded cluster 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 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:

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

   ![Note](image)
   - `<Mongos_host>`: the connection string of a mongos in the destination sharded cluster instance.
   - `<username>`: the username you use to log on to a database of the destination sharded cluster 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 s-bp**********-pub.mongodb.rds.aliyuncs.com:3717
    -u root --authenticationDatabase 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 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.