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, consequential, 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>![danger]</td>
<td>A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.</td>
<td>![danger] Danger: Resetting will result in the loss of user configuration data.</td>
</tr>
<tr>
<td>![warning]</td>
<td>A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.</td>
<td>![warning] Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.</td>
</tr>
<tr>
<td>![caution]</td>
<td>A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.</td>
<td>![caution] Notice: If the weight is set to 0, the server no longer receives new requests.</td>
</tr>
<tr>
<td>![note]</td>
<td>A note indicates supplemental instructions, best practices, tips, and other content.</td>
<td>![note] 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><strong>Italic</strong></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

Legal disclaimer ........................................................................................................ 1
Document conventions ............................................................................................ 1
1 Preface .................................................................................................................. 1
2 Quick start ............................................................................................................ 2
3 Logon and logoff ................................................................................................. 3
4 Billing management ............................................................................................ 5
   4.1 Change the billing method of an ApsaraDB for MongoDB instance from pay-as-you-go to subscription ................................................................. 5
   4.2 Manually renew an ApsaraDB for MongoDB subscription instance .......... 6
   4.3 Enable and disable auto-renewal for an ApsaraDB for MongoDB subscription instance ........................................................................................................... 8
5 Instance connection ........................................................................................... 11
   5.1 Connect to an ApsaraDB for MongoDB instance through a cross-zone intranet .... 11
   5.2 Connect to an ApsaraDB for MongoDB instance ............................................ 12
   5.3 How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different ......................................................... 13
   5.4 How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions .................................................. 16
   5.5 How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same Alibaba Cloud account .................. 18
   5.6 Connect a local client to an ApsaraDB for MongoDB instance over the Internet .... 20
   5.7 Connect a local client to an ApsaraDB for MongoDB instance through an SSL VPN tunnel ............................................................................................. 23
6 Account management ........................................................................................ 29
   6.1 Reset the password for an ApsaraDB for MongoDB instance ....................... 29
   6.2 Manage MongoDB users through DMS .......................................................... 31
7 Instance management ....................................................................................... 36
   7.1 Specify a maintenance period ...................................................................... 36
   7.2 Configuration change overview .................................................................... 37
   7.3 Change the configuration of a standalone or replica set instance ............... 42
   7.4 Change the number of nodes for a replica set instance ............................... 46
   7.5 Change the configuration of a sharded cluster instance ............................. 48
   7.6 View zones of nodes ..................................................................................... 51
   7.7 Migrate an ApsaraDB for MongoDB instance across zones in the same region .... 53
   7.8 Switch node roles ....................................................................................... 57
   7.9 Export the list of instances ........................................................................... 60
   7.10 Upgrade the minor version of an ApsaraDB for MongoDB instance ......... 61
   7.11 Upgrade the database version of an ApsaraDB for MongoDB instance .... 63
7.12 Release an ApsaraDB for MongoDB instance..........................65
7.13 Restart an ApsaraDB for MongoDB instance..........................66

8 Tag management..................................................................69
  8.1 Create a tag........................................................................69
  8.3 Filter instances by tag.........................................................71
  8.4 Unbind or delete a tag........................................................72

9 Network connection management.......................................74
  9.1 Apply for a connection string of a shard or Configserver node...74
  9.2 Release the connection string of a shard or Configserver node....77
  9.3 Enable or disable password-free access for an ApsaraDB for MongoDB instance........................................78
  9.4 Modify a public or internal endpoint of an ApsaraDB for MongoDB instance..........................80
  9.5 Switch the network type of an ApsaraDB for MongoDB instance..........................82
  9.6 Configure a VPC for a new instance....................................85
  9.7 Configure a hybrid access solution to switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC..........................87
  9.8 Change the retention period of internal endpoints on the classic network of an ApsaraDB for MongoDB instance..........................89
  9.9 Apply for a public endpoint for an ApsaraDB for MongoDB instance..........................90
  9.10 Release a public connection string.....................................90

10 Data security...................................................................93
  10.1 Configure a whitelist for an ApsaraDB for MongoDB instance..........................93
  10.2 Configure audit logging for an ApsaraDB for MongoDB instance..........................99
  10.3 Configure SSL encryption for an ApsaraDB for MongoDB instance..........................102
  10.4 Configure TDE....................................................................104
  10.5 Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode.......................................................107
  10.6 SSL connection sample code for MongoDB drivers..........................109

11 Monitoring and alerting.................................................115
  11.1 View monitoring information.............................................115
  11.2 Set the monitoring granularity..........................................118
  11.3 Set alert rules..................................................................120

12 Parameter settings.......................................................122
  12.1 Configure database parameters for an ApsaraDB for MongoDB instance..........................122
  12.2 View the parameter modification history................................124

13 Primary/Secondary failover..........................................126
  13.1 Trigger a primary/secondary failover for a replica set instance..........................126
  13.2 Trigger a primary/secondary failover for a shard of a sharded cluster instance..........................127

14 Log management.......................................................130
  14.1 View slow query logs........................................................130
  14.2 View error logs..................................................................131
  14.3 View operation logs........................................................132
15 Data backup

15.1 Configure automatic backup for an ApsaraDB for MongoDB instance

15.2 Manually back up an ApsaraDB for MongoDB instance

16 Data recovery

16.1 Restoration solution overview

16.2 Restore data to a new ApsaraDB for MongoDB instance from databases

16.3 Restore data to a new ApsaraDB for MongoDB instance by backup set

16.4 Restore data to a new ApsaraDB for MongoDB instance by point in time

16.5 Restore data to your current ApsaraDB for MongoDB instance

16.6 Restore data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using logical backup

16.7 Recover physical backup data in a user-created MongoDB instance

17 CloudDBA

17.1 Index optimization of ApsaraDB for MongoDB

18 Zone-disaster restoration solution

18.1 Create a multi-zone replica set instance

18.2 Create a multi-zone sharded cluster instance
Preface

This document describes how to use ApsaraDB for MongoDB in the ApsaraDB for MongoDB console to help you deeply understand the features of ApsaraDB for MongoDB.

Overview

ApsaraDB for MongoDB is a stable, reliable, and scalable database service that fully complies with the MongoDB protocols. The service provides a complete database solution for disaster recovery, data backup, data recovery, monitoring, and alarms.

To contact technical support, you can log on to the ApsaraDB for MongoDB console and choose More > Support > Open a new ticket or click here to submit a ticket.

For more information about the features and pricing of Alibaba Cloud ApsaraDB for MongoDB, visit the product page of ApsaraDB for MongoDB.

Disclaimer

Some product features or services described in this document may not be included in the scope that you can purchase or use. Your actual business contract and terms shall prevail. This document provides only guidance. No content in this document shall be deemed as explicit or implicit guarantees. Due to product version upgrades or other reasons, the content of this document may be occasionally updated. When using this document, you need to ensure that the document version is consistent with the corresponding software version.
2 Quick start

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

• Get started with standalone instances
• Get started with replica set instances
• Get started with sharded cluster instances
3 Logon and logoff

This topic describes how to log on to and log off from the ApsaraDB for MongoDB console. In the console, you can manage ApsaraDB for MongoDB instances including creating and connecting to instances.

Prerequisites

- You have an Alibaba Cloud account. If you do not have one, create a Alibaba Cloud account.
- Before logging on to the ApsaraDB for MongoDB console, you must purchase an ApsaraDB for MongoDB instance by using your Alibaba Cloud account. For more information about how to create an instance, see #unique_8, #unique_9, and #unique_10.

Note:
For more information about the billing information, see ApsaraDB for MongoDB Pricing.

Log on to the ApsaraDB for MongoDB console

1. Log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account.
2. In the upper-left corner of the page, select the region where the ApsaraDB for MongoDB instance resides.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.

4. Find the target instance and click its instance ID.

5. On the page that appears, manage the instance as needed.

---

**Log off from the ApsaraDB for MongoDB console**

Move the pointer over your avatar in the upper-right corner of the page and choose **Sign out**.
4 Billing management

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

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

Prerequisites

• The instance is in the running state.
• The billing method of the instance is pay-as-you-go.
• The instance has no unpaid subscription orders.
• The type of the instance is available for purchase. For more information about phased-out instance types, see the "Historical instance specifications" section in Instance specifications. If you need to change the billing method of an instance with a phased-out instance type to subscription, change the instance type first. For more information, see Configuration change overview.

Precautions

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

Procedure

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

6. On the **Confirm Order** page, specify **Purchase Cycle** of the instance.

7. Select ApsaraDB for MongoDB Agreement of Service and click **Activate**.

   **Note:**

   The system generates an order to switch the billing method to subscription. You cannot purchase a new instance or change the billing method to subscription until you pay for this order or cancel it. You can pay for or cancel this order on the **Billing Management** page.

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

### 4.2 Manually renew an ApsaraDB for MongoDB subscription instance

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

**Context**

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

**Procedure**

1. Log on to the **ApsaraDB for MongoDB console**.

2. In the upper-left corner of the page, select the region where the target ApsaraDB for MongoDB subscription instance resides.

3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the target instance and click its instance ID.

5. In the **Basic Information** section, click **Renew**.

6. **Specify Duration.**

   ![Renew](image)

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

7. Select **ApsaraDB for MongoDB Agreement of Service** and click **Pay**. Complete the payment as instructed.
4.3 Enable and disable auto-renewal for an ApsaraDB for MongoDB subscription instance

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

Context

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

Note:

When purchasing a subscription instance, you can select Auto Renew next to Duration.

- Subscription on a monthly basis: The auto-renewal cycle is a month.
- Subscription on a yearly basis: The auto-renewal cycle is a year.

Procedure

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

![Auto-Renew tab](image)

**Note:**

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

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

![Modify Auto-Renew dialog box](image)

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

The changes will be applied to the following products:

- **Modify Auto-Renew Cycle**
  - 3 Months
- **Disable Auto-Renew**

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Expiration Date</th>
<th>Remaining Days</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apr 9, 2019, 00:00</td>
<td>17 Days</td>
</tr>
</tbody>
</table>
6. Select an auto-renewal cycle and click OK.
5 Instance connection

5.1 Connect to an ApsaraDB for MongoDB instance through a cross-zone intranet

Currently, Alibaba Cloud intranets are classified into classic networks and VPCs. Cloud products, such as an ECS instance and an ApsaraDB for MongoDB instance, in different zones of the same region can be interconnected through an intranet.

This topic describes two scenarios.

Connect an ECS instance to a new ApsaraDB for MongoDB instance

- If the network type of the ECS instance is VPC and you purchase an ApsaraDB for MongoDB instance in a different zone of the same region, you need to ensure that the two instances have the same VPC ID. In addition, you need to create a VSwitch in the same zone as the ApsaraDB for MongoDB instance. In this way, the two instances can be interconnected properly through an intranet.

- If the network type of the ECS instance is classic network and you purchase an ApsaraDB for MongoDB instance in a different zone of the same region, you need to ensure that the two instances are on the same classic network. In this way, they can be interconnected through a cross-zone intranet.

Connect an ECS instance to an existing ApsaraDB for MongoDB instance

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

- If the two instances are configured with the same network type (either classic network or VPC with the same VPC ID), they can be interconnected through an intranet.

- If the two instances are configured with different network types, you can switch the network type of the ApsaraDB for MongoDB instance to be the same as that of the ECS instance before their interconnection.

Note:
You cannot switch the network type for standalone instances.
5.2 Connect to an ApsaraDB for MongoDB instance

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

Connection methods

<table>
<thead>
<tr>
<th>Category</th>
<th>Connection method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone instance</td>
<td>• #unique_22&lt;br&gt;• #unique_23&lt;br&gt; • Connection sample code for MongoDB drivers</td>
</tr>
<tr>
<td>Replica set instance</td>
<td>• #unique_25&lt;br&gt; • #unique_26&lt;br&gt; • Connection sample code for MongoDB drivers</td>
</tr>
<tr>
<td>Sharded cluster instance</td>
<td>• #unique_27&lt;br&gt; • #unique_28&lt;br&gt; • Connect to an ApsaraDB for MongoDB instance through the program code</td>
</tr>
</tbody>
</table>

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

Troubleshooting

- How to troubleshoot logon issues for the mongo shell
- #unique_35
- Troubleshoot the high CPU usage of ApsaraDB for MongoDB
- How to query and limit the number of connections
5.3 How to connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different

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

Prerequisites

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

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

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

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

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

![Note:](image)

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

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

To enable ClassicLink, perform the following steps:

1. Log on to the VPC console.
2. Select the region of the VPC and click the ID of the VPC.
3. On the VPC Details page, click Enable ClassicLink. In the dialog box that appears, click OK.
4. Log on to the ECS console.
5. In the left-side navigation pane, click Instances.
6. In the upper-left corner of the page, select the region where the instance resides.
7. In the Operation column corresponding to the ECS instance in a classic network, choose More > Network and Security Group > Set classic link.
8. In the dialog box that appears, select the VPC to which the ApsaraDB for MongoDB instance belongs and click OK.
9. In the Connect to VPC dialog box that appears, click Go to the instance security group list and add ClassicLink rules.
10. Click Add ClassicLink Rule. Configure the following parameters and then click OK.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic Security Group</td>
<td>The name of the classic network security group.</td>
</tr>
<tr>
<td>Select VPC Security Group</td>
<td>Select a VPC security group.</td>
</tr>
</tbody>
</table>
ApsaraDB for MongoDB

User Guide / 5 Instance connection

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

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

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

Note:

- You cannot change the network type of standalone instances.
- Switching network types will cause a transient disconnection of the ApsaraDB for MongoDB instance. Perform this operation during off-peak hours or ensure that your application has a reconnection mechanism to prevent negative impacts on your business.
5.4 How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions

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

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

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

1. Create an ApsaraDB for MongoDB instance in the region where the ECS instance is located. For more information, see Create an instance. Skip this step if you have already created an ApsaraDB for MongoDB instance.
2. Migrate the MongoDB database from the instance in the source region to the instance in the destination region. For more information, see #unique_39.
3. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist.

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

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

You can use the custom image feature or the migration tool to migrate the ECS instance data from the original region to the region where the ApsaraDB for MongoDB instance is located. For example, you can migrate the ECS instance from the China (Qingdao) region to the China (Hangzhou) region.
• Create a custom image from the ECS instance and then create an ECS instance in the region where the ApsaraDB for MongoDB instance is located from the custom image (this method is recommended).

1. **Create a custom image from the ECS instance.**

2. Copy the created custom image to the region where the ApsaraDB for MongoDB instance is located. For more information, see **Copy an image.**

3. **Create an ECS instance from the custom image.**

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

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

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

• Use the migration tool to migrate the ECS instance to the region where the ApsaraDB for MongoDB instance is located.

1. Migrate the ECS instance to the region where the ApsaraDB for MongoDB instance is located. For more information, see **Migrate ECS instances.**

2. Add the private IP address of the ECS instance to the whitelist of the ApsaraDB for MongoDB instance. For more information, see **Configure a whitelist.**
5.5 How to connect an ECS instance to an ApsaraDB for MongoDB instance when they do not belong to the same Alibaba Cloud account

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

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

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

Procedure

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

   **Note:**
   
   When creating the ApsaraDB for MongoDB instance, select the same region, zone, and VPC as the ECS instance.

2. Migrate the MongoDB database from the instance that belongs to the source Alibaba Cloud account to the instance that belongs to destination Alibaba Cloud account. For more information, see #unique_40.

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

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

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

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

The ECS instance and ApsaraDB for MongoDB instance must be in the same region because images cannot be shared across different regions.

Procedure

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

   Note:
   When creating the ECS instance, select the same VPC as the ApsaraDB for MongoDB instance.

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

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

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

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

   Note:
   Ensure that the CIDR blocks of the VPCs or VSwitches involved do not conflict with each other.

Procedure

1. Switch the network type of the ApsaraDB for MongoDB instance to VPC. For more information, see Switch from a classic network to a VPC. If the network type is VPC, skip this step.
2. Switch the network type of the ECS instance to VPC. If the network type is VPC, skip this step.
3. Based on the running environment, select one of the following CEN-based connections over the internal network. For more information, see

- Connect instances that are in the same regions and belong to different accounts.
- Connect instances that are in different regions and belong to different accounts.

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

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

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

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

**Prerequisites**

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

- Apply for a public endpoint for a standalone instance
- Apply for a public endpoint for a replica set instance
- Apply for a public endpoint for a sharded cluster instance

**Precautions**

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

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

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

1. Query the public IP address of your local client.
2. Add the public IP address to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

3. Log on to the ApsaraDB for MongoDB instance by using the mongo shell from your local client. For more information, see #unique_26.

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

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

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

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

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

2. Log on to the ApsaraDB for MongoDB instance by using the mongo shell from your local client. For more information, see #unique_26.
3. Run the following command to query information about the client where you log on:

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

The following figure shows an example.

![Image of db.currentOp output]

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

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

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

References

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

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

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

Scenarios

- The public IP address of the local client changes dynamically. As a result, you must frequently update the whitelist that contains the public IP address of the local client on the ApsaraDB for MongoDB console. If you do not delete expired IP addresses at the earliest opportunity, security risks may arise.
- A higher level of security is required when you connect to an ApsaraDB for MongoDB instance over the Internet.
- You need to log on to the ApsaraDB for MongoDB instance from an ECS instance over the Internet. This may cause security risks. Therefore, you must separate ECS management permissions from ApsaraDB for MongoDB database permissions.

Billing

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

Prerequisites

- VPC is the network type of the ApsaraDB for MongoDB instance. For more information about how to switch the network type from Classic Network to VPC, see Switch from Classic Network to VPC.
- The Classless Inter-Domain Routing (CIDR) block of the local client is different from that of the ApsaraDB for MongoDB instance.
- The local client can access the Internet.
Step 1 Create a VPN gateway

1. Log on to the VPC console.
2. In the upper-left corner of the page, select a region.
3. In the left-side navigation pane, choose VPN > VPN Gateways.
4. On the VPN Gateways page, click Create VPN Gateway.
5. Configure the VPN gateway as needed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Optional. Enter the name of the VPN gateway that you want to create.</td>
</tr>
<tr>
<td>Region</td>
<td>Select the region where the ApsaraDB for MongoDB instance resides.</td>
</tr>
<tr>
<td>VPC</td>
<td>Select the VPC to which the ApsaraDB for MongoDB instance belongs.</td>
</tr>
<tr>
<td>Peak Bandwidth</td>
<td>Select the peak Internet bandwidth of the VPN gateway.</td>
</tr>
<tr>
<td>IPsec-VPN</td>
<td>Enable or disable the IPsec-VPN function as needed. For this example, click Disable for IPsec-VPN because the local client will access the VPN directly. The IPsec-VPN function provides site-to-site connections. You can create an IPsec tunnel to connect a on-premises data center to an VPC, or connect two VPCs.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SSL-VPN</td>
<td>Enable or disable the SSL-VPN function as needed. For this example, click <strong>Enable</strong> for SSL-VPN because the local client will access the VPN directly. The SSL-VPN function provides point-to-site connections. You do not need to configure a gateway for the local client because the client can access the VPN directly.</td>
</tr>
<tr>
<td>Billing Cycle</td>
<td>Select a subscription duration for the VPN gateway. The duration can be one to nine months on a monthly basis or one to three years on a yearly basis. You can also choose whether to enable auto-renewal. The auto-renewal cycle is one month for monthly subscription and is one year for yearly subscription.</td>
</tr>
</tbody>
</table>

6. Click **Buy Now** and complete the payment as instructed.

**Step 2 Create an SSL server**

1. Log on to the VPC console.
2. In the upper-left corner of the page, select a region.
3. In the left-side navigation pane, choose **VPN > SSL Servers**.
4. On the **SSL Servers** page, click **Create SSL Server**.
5. In the **Create SSL Server** dialog box, configure the SSL server.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the SSL server. The name must be 2 to 128 characters in length and must start with a letter. It can contain letters, digits, underscores (_), and hyphens (-).</td>
</tr>
<tr>
<td>VPN Gateway</td>
<td>Select the VPN gateway created in <strong>Step 1 Create a VPN gateway</strong>.</td>
</tr>
</tbody>
</table>
### Parameter Description

**Local Network**
Enter the network segment to be accessed by the local client through the SSL VPN tunnel. It can be the network segment of a VPC, a VSwitch, an IDC that is connected with a VPC by using a leased line, or an Alibaba Cloud service such as ApsaraDB for RDS and Object Storage Service (OSS).

For this example, enter the network segment of the VSwitch in the VPC to which the ApsaraDB for MongoDB instance belongs: 172.16.1.0/24.

**Note:**
The subnet mask of the local network segment must be bit 16 to bit 29.

**Client Subnet**
Enter the network segment where access addresses will be assigned to the virtual network interface cards (NICs) of the local client. When the local client accesses the instance through an SSL VPN tunnel, the VPN gateway assigns an IP address from the specified client network segment to the client.

For this example, enter 192.168.100.0/24.

**Note:**
Make sure that Client Subnet and **Local Network** are different.

6. Click **OK**.

### Step 3 Create an SSL client

1. Log on to the **VPC console**.
2. In the upper-left corner of the page, select a region.
3. In the left-side navigation pane, choose **VPN > SSL Clients**.
4. On the **SSL Clients** page, click **SSL Client Certificate**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of the SSL client certificate. The name must be 2 to 128 characters in length and must start with a letter. It can contain letters, digits, underscores (_), and hyphens (-).</td>
</tr>
<tr>
<td>SSL Server</td>
<td>Select the SSL server created in <strong>Step 2 Create an SSL server</strong>.</td>
</tr>
</tbody>
</table>
5. Click OK.

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

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

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

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

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

**Note:**
Log on using an internal endpoint of the ApsaraDB for MongoDB instance.
6 Account management

6.1 Reset the password for an ApsaraDB for MongoDB instance

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

Limits

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

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target ApsaraDB for MongoDB subscription instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its instance ID.
5. In the left-side navigation pane, click Accounts.
6. Perform one of the following operations as needed.

- Reset the password of the root user.

Find the root user and click **Reset Password** in the **Operation** column.

- Reset the password of an account for a shard or config server in a sharded cluster instance.

  **Note:**

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

Find the account created when you apply for an endpoint for a shard or config server, and click **Reset Password** in the **Operation** column. For this example, find shardaccount. For more information, see **Apply for a connection string of a shard or Configserver node**.

7. In the dialog box that appears, enter a new password and confirm it.

  **Note:**

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

### 6.2 Manage MongoDB users though 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. After you connect to an ApsaraDB for MongoDB instance through DMS, you can easily create, modify, and delete MongoDB users.

**Create a user**

1. Log on to the DMS console.
2. In the top navigation bar, choose **Create > User**.
3. On the **Create User** page, configure user-related parameters.

### Parameter | Description
--- | ---
**Destination Database** | Select the database that you want to save from the drop-down list.

**Note:**
- If you select a destination database other than the admin database, you will create a common user.
- If you select the **admin** database, you will create a privilege user.

**Username** | The name of the user.
- The name cannot contain Chinese characters.
- It can contains letters, digits, and special characters.
- Special characters include: `! # $ % ^ & * ( ) _ + - =`
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>The password of the user. To ensure data security, the password must contain characters from at least three of the following categories: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! # $ % ^ &amp; * ( ) _ + - =</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Enter the password again.</td>
</tr>
</tbody>
</table>

4. On the **Create User** page, configure permission-related parameters.

![Create User page](image)

**Privileges on Current Database**
- **Common Operation Role**
  - read
  - readWrite
- **Database Administrator**
  - dbAdmin
  - except database read and write permissions
  - userAdmin
database

**Note:**
• If you select a destination database other than the admin database,

On the **Current Library Permissions** tab, you can only set permissions for **Current Library ordinary operation role** and **Current Library administrator operation role**.

No permissions can be set on the **Other library Permissions** tab.

• If you select the admin database,

On the **Current Library Permissions** tab, you can set permissions for more roles.

You can set permissions on the **Other library Permissions** tab. You can add a database name and set a role for the database on this tab.

5. Click **Confirm**.

**Edit a user**

1. Log on to the DMS console.

2. On the **Object List** item in the left-side navigation pane, expand the **database** for which you want to edit a user.

3. Expand the **user**. Right-click it and select **Edit user**.

4. In the **Edit user** dialog box that appears, set the password or permission information for the user.

5. Click **Confirm**.
Delete a user

**Note:**
To guarantee instance stability, you cannot delete the root user which is reserved by the system.

1. Log on to the DMS console.
2. On the **Object List** item in the left-side navigation pane, expand the **database** for which you want to delete a user.
3. Expand the **user** and right-click it.
4. Select **Delete user**.
5. In the **message** that appear, click **OK**. The user is deleted.
7 Instance management

7.1 Specify a maintenance period

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

Context

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

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

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Locate the target instance and click its instance ID.
5. In the **Specification Information** area, click **Edit** to the right of **Maintenance Period**.

![Specification Information](image)

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

### 7.2 Configuration change overview

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

For more information about instance specifications, see [#unique_13](#).

For more information about correspondence and restrictions between versions and storage, see [#unique_52](#).

**Note:**

For more information about precautions and procedure for version upgrade, see [Upgrade the database version of an ApsaraDB for MongoDB instance](#).

#### Standalone instances

<table>
<thead>
<tr>
<th>Item</th>
<th>Configuration change supported</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifications</td>
<td>Yes</td>
<td>For more information, see <a href="#">Change the configuration of a standalone or replica set instance</a>.</td>
</tr>
<tr>
<td>Item</td>
<td>Configuration change supported</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Storage space</td>
<td>Yes</td>
<td>For more information, see Change the configuration of a standalone or replica set instance. If the billing method is subscription, you cannot downgraded the storage space. You must perform the following operations: 1. Create a pay-as-you-go standalone instance and select the required storage space. For more information, see #unique_8. Note: The storage space of the new instance must be larger than the occupied storage space in the original instance. 2. Use DTS to migrate data from the original instance to the new instance. For more information, see #unique_55. 3. Test and verify the new instance. If it runs normally, switch business to the new instance. Note: If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount. 4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.</td>
</tr>
</tbody>
</table>

Note:
### ApsaraDB for MongoDB

#### User Guide / 7 Instance management

<table>
<thead>
<tr>
<th>Item</th>
<th>Configuration change supported</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>No</td>
<td>You cannot change the number of nodes, architecture, and storage engine for a standalone instance. You must perform the following operations:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Create a pay-as-you-go instance. Select the required number of nodes, architecture, and storage engine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• To increase the number of nodes, you must create a replica set instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The storage space of the new instance must be larger than the occupied storage space in the original instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Use DTS to migrate data from the original instance to the new instance. For more information, see #unique_55.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Test and verify the new instance. If it runs normally, switch business to the new instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.</td>
</tr>
</tbody>
</table>

#### Replica set instances

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Configuration change supported</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td></td>
<td>You cannot change the architecture and storage engine of a replica set instance. You must perform the following operations:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Create a pay-as-you-go instance. Select the required architecture and storage engine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The storage space of the new instance must be larger than the occupied storage space in the original instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Use DTS to migrate data from the original instance to the new instance. For more information, see #unique_55.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Test and verify the new instance. If it runs normally, switch business to the new instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.</td>
</tr>
</tbody>
</table>

### Sharded cluster instances

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

Note:

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mongos node</td>
<td>Specifications</td>
<td>For more information, see Change the configuration of a sharded cluster instance.</td>
</tr>
<tr>
<td>Component</td>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Shard node</td>
<td>Specifications</td>
<td>For more information, see <a href="#">Change the configuration of a sharded cluster instance</a>.</td>
</tr>
<tr>
<td></td>
<td>Storage space</td>
<td>If the billing method is subscription, you cannot downgrade the storage space. You must perform the following operations:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. <strong>Restore data to a new ApsaraDB for MongoDB instance by point in time</strong> Create a pay-as-you-go standalone instance and select the required</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Note: The storage space of the new instance must be larger than the occupied storage space in the original instance.</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Test and verify the new instance. If it runs normally, switch business to the new instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Note: If a long period of use, we recommend that you switch from pay-as-you-go to subscription. This billing method is more cost-effective than pay-as-you-go. The longer the subscription period, the higher the discount.</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. If the original instance is no longer needed, you can manually release the pay-as-you-go instance or submit a ticket to release the subscription instance.</td>
</tr>
<tr>
<td>Configserver node</td>
<td>Specifications and storage space</td>
<td>A Configserver node uses a fixed three-node replica set architecture. By default, 1 core, 2 GB memory, and 20 GB storage space are selected. You cannot change these items.</td>
</tr>
</tbody>
</table>

### 7.3 Change the configuration of a standalone or replica set instance

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

**Precautions**

- When you change the configuration, the new storage space must be larger than the storage space occupied by the current instance.
• If the billing method is subscription, the interval between two configuration downgrades cannot be less than 60 days.

• When the billing method is subscription, you cannot downgrade the storage space. You can use other methods to reduce the storage space. For more information, see Configuration change overview.

• You cannot change the instance type (such as from a standalone instance to a replica set instance) or the storage engine. You can use other methods to change these items. For more information, see Configuration change overview.

Billing rules
For more information, see #unique_61.

Impacts
• Changing configurations does not cause data loss.

• Pre-operations for configuration changes to an instance do not affect the running of the instance. However, when configuration changes are formally executed on the instance, most operations related to databases, accounts, and network cannot be performed. One or two transient disconnections of up to 30 seconds will occur. For more information, see Select switching time.

• The duration of a configuration change depends on various factors such as network conditions, task queues, and data volume. We recommend that you change configurations during off-peak hours and make sure that your applications have automatic reconnection mechanisms.

Select switching time
On the Change Configuration, you can specify switching time. The following table describes details about switching time.
<table>
<thead>
<tr>
<th>Item</th>
<th>Instance status</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Switch Within</strong></td>
<td>The instance immediately enters the Changing Configuration state.</td>
<td>The system performs pre-operations, which do not affect the running of the instance or cause transient disconnections. Configuration changes are formally executed within the maintenance period you set. For example, if the preset maintenance period is 2:00 to 3:00, configuration changes will be performed during this period. Most operations related to databases, accounts, and network cannot be performed. One or two transient disconnections of up to 30 seconds will occur.</td>
</tr>
<tr>
<td><strong>Maintenance Window</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Switch Immediately</strong></td>
<td></td>
<td>Configuration changes are performed immediately. Most operations related to databases, accounts, and network cannot be performed. One or two transient disconnections of up to 30 seconds will occur.</td>
</tr>
<tr>
<td><strong>After Data Migration</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
For more information about how to modify a maintenance period, see Specify a maintenance period.

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances.
4. Change the configuration of the instance.
   If the billing method of the instance is pay-as-you-go, perform the following steps:
   a) Find the instance and click its ID.
   b) In the Basic Information section, click Change Configuration.


   For more information about instance specifications, see #unique_13.

   Note:
   • For more information about the limits on parameters, see Precautions.
   • For more information about the selection and impacts of Migration Time, see Select switching time.

6. Select ApsaraDB for MongoDB Terms of Service and make the payment as prompted.

   Result
   When the instance status changes to Running, the configuration has been changed.
7.4 Change the number of nodes for a replica set instance

To meet data reading performance requirements in various business scenarios, the number of nodes in a replica set instance can be changed in ApsaraDB for MongoDB. Data can be read from added secondary nodes. This method improves the overall read performance of replica set instances.

Context

To meet the high availability of ApsaraDB for MongoDB, the number of nodes in replica set instances can be changed to 3, 5, and 7.

Note:
The nodes of standalone instances cannot be changed.

You can add or remove nodes for a replica set instance, which must have at least three nodes. Changing the number of nodes for a replica set instance results in a change to its bill. For more information, see Billing items and pricing.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances.
4. Find the instance and click its ID.
5. On the Basic Information page, perform the following steps based on the billing method of the instance:
   a) For a pay-as-you-go instance, click Upgrade or Downgrade in the Basic Information section.
   b) For a subscription instance, click Change Configuration in the Basic Information section.
6. On the **Change Configuration** page, specify **Replication Factor** for the instance.

![Configuration Upgrade](image)

**Note:**
For more information about how to change specifications and the storage space, see **Configuration change overview**.

7. Set **Migration Time**.

**Note:**
- **Switch Immediately After Data Migration**: After the configuration change process is complete, the instance immediately enters the **Changing Configuration** state. The configuration has been changed when the status of the instance changes to **Running**.

  During some configuration upgrades, one or two transient disconnections of up to 30 seconds may occur to the instance. You can set the switching time for the configuration change as required to avoid negative impacts on your business.

- **Migrate at Scheduled Time**: You can set the time for the configuration change within a specified period. For more information, see Specify a maintenance period.

  If no transient disconnection is caused during the configuration change, the configuration change can immediately take effect regardless of whether you have set the switching time.

8. Select **ApsaraDB for MongoDB Terms of Service** and make the payment as prompted.

**What's next**
After you add the nodes for the replica set instances, the connection strings of new nodes (all displayed as Secondary but with different role IDs) appear in the console. The connection string URI for a high availability connection is also updated. You can modify the connection string in an application to achieve high availability and read/write splitting connection and improve the overall performance. For more information, see Connect to a replica set instance through the mongo shell.

### 7.5 Change the configuration of a sharded cluster instance

You can change the configuration of a sharded cluster instance if the configuration is excessive or cannot meet the performance requirements of an application.

**Precautions**

- To ensure data security, you can add shard nodes but cannot delete them.
- When you change the configuration of a shard node, the new storage space must be larger than the storage space occupied by the current shard node.
- If the billing method is subscription, you can only upgrade the configuration.

**Note:**

You can use other methods to downgrade the configuration. For more information, see Configuration change overview.

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

**Billing rules**

For more information, see #unique_61.

**Impacts**

- Changing configurations does not cause data loss.
- You cannot specify switching time for the changed configuration of a sharded cluster instance. After you change the configuration in the console, the instance status changes to Changing Configuration immediately. When the instance is in this state, most operations related to databases, accounts, and network cannot be performed. One or two transient disconnections of up to 30 seconds will occur.
- The duration of a configuration change depends on various factors such as network conditions, task queues, and data volume. We recommend that you change configurat...
ions during off-peak hours and make sure that your applications have automatic reconnection mechanisms.

Add a node

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the instance and click its ID.
5. On the Basic Information page, click the corresponding button to add a node.

To add a Mongos node, perform the following steps:
   a) In the Mongos List section, click Add Mongos.
   b) On the Add Mongos page, specify Specifications of the Mongos node.

To add a shard node, perform the following steps:
   a) In the Shard List section, click Add Shard.
   b) On the Add Shard page, specify Specifications and Storage of the shard node.

6. Select ApsaraDB for MongoDB Terms of Service and make the payment as prompted.

Change the configuration of an existing node

For more information about instance specifications, see #unique_13.

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

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

4. Find the instance and click its ID.

5. On the **Basic Information** page, change the configuration of the corresponding node.
   To change the configuration of a Mongos node, perform the following steps:
   a) In the **Mongos List** section, find the Mongos node and choose **Change Configuration**.
   b) On the **Change Mongos Configuration** page, specify **Specifications** of the Mongos node.

   To change the configuration of a shard node, perform the following steps:
   a) In the **Shard List** section, choose **Change Configuration**.
   b) On the **Change Shard Configuration** page, specify **Specifications** and **Storage** of the shard node.

   **Note:**
   If the billing method of the instance is subscription, you cannot downgrade the storage space of the shard node. You can use other methods to reduce the storage space. For more information, see **Configuration change overview**.

6. Select **ApsaraDB for MongoDB Terms of Service** and make the payment as prompted.

**Result**
When the instance status changes to **Running**, the configuration has been changed.

### 7.6 View zones of nodes

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

**Prerequisites**

Replica set or sharded cluster instances must be used.

**Deployment tips**

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

**Note:**

For more information about comparison of node deployment policies for single and multiple zones, see Node deployment policies and Node deployment policy for a sharded cluster instance.

**View zones of nodes**

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

- **Replica set instances**

- **Sharded cluster instances**

**References**

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

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

- **Switch node roles**

You can switch the node roles of an ApsaraDB for MongoDB instance based on your business deployment. This allows your applications to connect to the nearest nodes.
7.7 Migrate an ApsaraDB for MongoDB instance across zones in the same region

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

Prerequisites

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

Precautions

• If the instance is in a VPC, you cannot change the VPC while you migrate the instance across zones.
• The time required varies depending on factors such as the network conditions, task queue status, and data volume. We recommend that you migrate the instance across zones during off-peak hours.
• While you migrate an instance across zones, there is a brief disconnection of 30 seconds. Make sure that your application is configured to reconnect to the instance after it is disconnected.
• Migrating an instance across zones causes changes to virtual IP addresses (VIPs), such as 172.16.88.60. If your application uses the original VIP, it cannot connect to the instance after migration.

Note:

We recommend that you use a connection string URI to connect to the instance, which ensures high availability. For more information, see #unique_45.
Supported migration types and scenarios

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate an ApsaraDB for MongoDB instance from one zone to another</td>
<td>The ApsaraDB for MongoDB instance is migrated to the zone where an ECS instance resides. This way, the ECS instance can connect to the ApsaraDB for MongoDB instance over the internal network with lower network latency.</td>
</tr>
<tr>
<td>Migrate an ApsaraDB for MongoDB instance from one zone to multiple zones</td>
<td>The ApsaraDB for MongoDB instance provides disaster recovery across data centers. The three nodes of a replica set instance are deployed to three different zones in the same region. This enables the instance to tolerate disasters at higher levels. For example, a replica set instance in a single zone can only tolerate server- and rack-level faults, whereas a replica set instance in multiple zones can tolerate server-, rack-, and data center-level faults.</td>
</tr>
<tr>
<td>Migrate an ApsaraDB for MongoDB instance from multiple zones to one zone</td>
<td>Special user requirements are met.</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 of the instance.
3. In the left-side navigation pane, click Replica Set Instances.
4. Find the instance and click its ID.
5. In the Basic Information section, click Change Zone.
6. In the dialog box that appears, configure parameters based on the network type of the instance.

- When the instance is in a VPC or is in hybrid network access mode:
  
  a. Select the destination zone and VSwitch.

![Migrate Instance to Other Zone](image)

![Migrate Instance to Other Zone](image)

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

- When the instance is in a classic network:

  a. Select the destination zone.
b. Specify Migration Time and select the check box of the warning message.

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

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

7. Click **Submit**.

### 7.8 Switch node roles

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

**Typical scenario**

When an ECS instance and an ApsaraDB for MongoDB instance are in the same zone and connected over the internal network, the latency is minimal. If they are connected across different zones, the latency increases and the performance of ApsaraDB for MongoDB instances and your business will be affected.
In this example, the ECS instance to which the application belongs is in Zone 2. If the primary node of the ApsaraDB for MongoDB instance is in Zone 1, the ECS instance needs to connect to the primary node across zones.

To optimize the business deployment architecture, you can switch the roles of the primary and secondary nodes. In this example, you can change the role of the node in Zone 2 to primary and the role of the node in Zone 1 to secondary. Note that only the node roles are changed. ECS and ApsaraDB for MongoDB instances can be connected in the same zone without changing the actual zones and role IDs.

**Prerequisites**

Replica set or sharded cluster instances must be used.

**Precautions**

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

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the instance and click its ID.
5. In the left-side navigation pane, click **Service Availability**.
6. Subsequent steps on the **Service Availability** page vary depending on instance types.
   - Replica set instances
     a. Click **Switch Role** in the upper-right corner of the page.
     b. In the **Switch Role** dialog box that appears, select the nodes.
   - Sharded cluster instances

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

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

b. In the Switch Role dialog box that appears, select the nodes.

7. Click OK.

7.9 Export the list of instances

You can export the list of instances through the ApsaraDB for MongoDB console to manage cloud instances offline.

Procedure

1. Log on to the ApsaraDB for MongoDB console.

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

3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. On the **Instances** page, click **Export**.

![Instance Export](image)

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

![Export Instance List](image)

6. Click **OK**.

**Note:**

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

### 7.10 Upgrade the minor version of an ApsaraDB for MongoDB instance

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

**Prerequisites**

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

**Precautions**

- You cannot downgrade an instance after you upgrade it.
- When an instance undergoes a minor version upgrade, it is restarted and has a brief disconnection of less than 30 seconds. We recommend that you perform the upgrade during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected.

**Procedure**

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

![Upgrade Minor Version Button](image)

**Note:**

When you use the latest minor version, the console does not display the Upgrade Minor Version button for the instance.
6. In the Upgrade Minor Version message that appears, view the version release log and determine whether to upgrade the instance.

![Update Minor Version](image)

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

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

7.11 Upgrade the database version of an ApsaraDB for MongoDB instance

This topic describes how to upgrade the database version of an ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console. ApsaraDB for MongoDB now supports 3.2, 3.4, and 4.0.

**Database versions**

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

**Precautions**

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

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

**Note:**
If your application runs in a production environment, we recommend that you use a connection string URI to connect your application to the instance. In this way, the read/write operations of your application remain available even if a node fails as a result of a primary/secondary switchover. For more information, see #unique_45 or #unique_70.

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the instance and click its ID.
5. On the **Basic Information** page, click **Upgrade Database Version** and select the target version.
6. In the **Upgrade Database Version** message that appears, click **OK**.

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

### 7.12 Release an ApsaraDB for MongoDB instance

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

**Prerequisites**

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

**Note:**

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

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where your instance resides.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find your instance and choose > **Release** in the **Operations** column.

Alternatively, you can click the ID of your instance. On the **Basic Information** page that appears, click **Release**.

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

### 7.13 Restart an ApsaraDB for MongoDB instance

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

**Precautions**

- When you restart an instance, all its nodes are restarted in turn and each node has a brief disconnection of about 30 seconds. If the instance houses more than 10,000 collections, the brief disconnections last longer. Therefore, we recommend that you restart an instance during off-peak hours or make sure that your application is configured to reconnect to the instance after it is disconnected.

- When you restart a replica set instance, a primary/secondary switchover may occur and cause the roles of connected nodes to change. We recommend that you use a connection string URI to connect to the instance to avoid impact on the read/write operations of your application. For more information, see #unique_45.

- You can restart a sharded cluster instance, or a mongos or shard in the instance. This kind of nodes are inaccessible until restarted.

**Restart an instance**

1. Log on to the **ApsaraDB for MongoDB console**.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the target instance and choose > **Restart** in the **Operations** column.

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

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

The instance status changes to **Rebooting**. When the instance enters the **Running** state, the restart is complete.

**Restart a node in a sharded cluster instance**

1. Log on to the [ApsaraDB for MongoDB console](#).

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

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

4. Find the instance and click its ID.
5. Follow these steps to restart a node:

- **Restart a mongos.**

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

- **Restart a shard.**

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

6. In the **Restart Node** message that appears, click **OK**.

   The instance status changes to **Rebooting**. When the instance enters the **Running** state, the restart is complete.
8 Tag management

8.1 Create a tag

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

Prerequisites

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

Precautions

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

  Note:

  A key can have zero to multiple values.

• You can edit tags for a maximum of 50 instances at a time.

• You can bind up to 20 tags to an instance.

• You can bind or unbind up to 20 tags at a time.

Procedure

1. Log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account.

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

3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances based on the instance type.
4. Follow these steps to create a tag.

   - Create tags for an instance: Find the instance. In the Operations column, choose Edit tags.

   **Edit tags.**

   ![Edit tags image]

   - Create tags for multiple instances: Select the instances for which you want to create tags and then click *Edit tags* at the bottom of the instance list.

   ![Multiple instances image]

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

   **Note:**
   If you have created tags, click Existing Tags to bind the tags to the instances. For more information, see #unique_74.
6. Set the key and value of the tag and then click **Ok**.

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

**Note:**
After creating tags, you can bind them to other instances.

**Result**
After creating a tag, you can view the tag information of the instance in the instance list.

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

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

**Procedure**

1. Log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account.
2. In the upper-left corner of the page, select the region where the instance resides.
3. In the left-side navigation pane, click or based on the instance type.
4. Click the Label search box, select the tag key or tag value to be filtered, and then click **Search**.

**Note:**

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

**8.4 Unbind or delete a tag**

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

**Prerequisites**

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

**Precautions**

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

**Procedure**

1. Log on to the ApsaraDB for MongoDB console with your Alibaba Cloud account.
2. In the upper-left corner of the page, select the region where the instance resides.
3. In the left-side navigation pane, click or based on the instance type.
4. Find the instance. In the Operations column, choose > Edit tags.

5. In the dialog box that appears, click .

   ![Edit tags dialog box]

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

6. Click OK.
9 Network connection management

9.1 Apply for a connection string of a shard or Configserver node

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

**Prerequisites**
Sharded cluster instances must be used.

**Precautions**
- After you apply for a connection string, two connection strings are allocated to each node, one for the primary node and one for the secondary node.
- The network type of the connection strings must be the same as that of the current Mongos node.
- You cannot modify the connection string of a shard or Configserver node.
- The connection strings allocated here can only be used to access the node over the internal network. If you want to access the node over the Internet, you can [unique_78].

**Introduction to the sharded cluster architecture and nodes**
For more information, see [unique_79].

**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 ApsaraDB for MongoDB subscription instance resides.
3. In the left-side navigation pane, click Sharding Instances.
4. Find the target instance and click its instance ID.
5. In the left-side navigation pane, click Database Connection.
6. Click Apply for Shard \ ConfigServer Connection String.
7. In the dialog box that appears, apply for a connection string for the shard or Configserver node.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Node Type            | • shard: the shard node.  
<pre><code>                    | • cs: the Configserver node.                                               |
</code></pre>
<p>| Select Node ID       | Select a check box corresponding to the ID of the node for which you want to create a connection string. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account</strong></td>
<td>The account name must be 4 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter.</td>
</tr>
</tbody>
</table>
| **Note:**     | • You must set the account and password only when applying for the connection string of a shard or Configserver node for the first time. The account and password are required for all shard and Configserver nodes.  
                   • The permissions of this account are fixed to read-only.                                                                                     |
| **Password**  | • The password must contain at least three 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.                                                                                       |
| **Confirm Password** | Enter the account password again.                                                                                                           |

8. Click OK.

9. Wait until the instance status changes from **Establishing Network Connection** to **Running**.

**Node types**

After applying for the connection string of a shard or Configserver node, you can view the connection string on the **Database Connection** page. The following table describes node types.
<table>
<thead>
<tr>
<th>Node type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db</td>
<td>The shard node.</td>
</tr>
<tr>
<td>cs</td>
<td>The Configserver node.</td>
</tr>
<tr>
<td>mongos</td>
<td>The Mongos node.</td>
</tr>
</tbody>
</table>

References

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

### 9.2 Release the connection string of a shard or Configserver node

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

**Precautions**

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

**Procedure**

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

**Note:** Issue: 20200506
7. In the dialog box that appears, click OK.

8. Wait until the instance status changes from Releasing Network Connection to Running.

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

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

Prerequisites

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

Note:

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

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

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the instance and click its ID.
5. In the left-side navigation pane, click Database Connection.
6. In the upper-right corner of the Intranet Connection - VPC section, click Enable password-free access or Disable password-free access.

- Enable password-free access.

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

- Disable password-free access.

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

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

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

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

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_83</td>
<td>Enables or disables password-free access over a VPC for an ApsaraDB for MongoDB instance.</td>
</tr>
</tbody>
</table>

### 9.4 Modify a public or internal endpoint of an ApsaraDB for MongoDB instance

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

### Limits

<table>
<thead>
<tr>
<th>Instance</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone instances</td>
<td>You can only modify the public and internal endpoints of a primary node.</td>
</tr>
<tr>
<td>Replica set instances</td>
<td>You can modify the public and internal endpoints of both primary and secondary nodes.</td>
</tr>
<tr>
<td>Sharded cluster instances</td>
<td>You can only modify the public and internal endpoints of a mongos.</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 ApsaraDB for MongoDB subscription instance resides.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the target instance and click its instance ID.

5. In the left-side navigation pane, click **Database Connection**.

6. In the **Intranet Connection** or **Public IP Connection** section, click **Update Connection String**.
7. In the dialog box that appears, enter a new endpoint.

![Update Connection String dialog box]

- **Note:**
  - You can only modify the prefix of the endpoint.
  - The new endpoint must be 8 to 64 characters in length and can contain letters and digits. It must start with a lowercase letter.

8. Click **Submit**.

**What's next**

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

**9.5 Switch the network type of an ApsaraDB for MongoDB instance**

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

**Prerequisites**

The instance is a replica set or sharded cluster instance.
Note:
For a standalone instance, its network type is always VPC and cannot be changed.

Precautions

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

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

Internal connection addresses

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

Switch from Classic Network to VPC

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

![Intranet Connection - Classic Network](image1)

8. In the **VPC** dialog box that appears, specify **VPC** and **VSwitch**.

![VPC](image2)

**Note:**

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

9. Click **OK**.
Switch from VPC to Classic Network

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

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

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

9.6 Configure a VPC for a new instance

ApsaraDB for MongoDB supports two network types: classic network and VPC. This topic describes how to configure a VPC for a new ApsaraDB for MongoDB instance.
On the Alibaba Cloud platform, a classic network and a VPC have the following differences:

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

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

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

Procedure

1. Create a VPC. For more information, see #unique_87.
2. Create an ApsaraDB for MongoDB instance in the same region as the VPC.
3. When creating the ApsaraDB for MongoDB instance, select **VPC** as the network type on the instance creation page.
4. Under **VPC**, select the configured VPC and VSwitch for **VPC** and **VSwitch**, respectively, as shown in the following figure.
5. On the instance creation page, specify other configuration items as required. For more information, see the following links.

- Create a standalone instance
- Create a replica set instance
- Create a sharded cluster instance

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

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

Prerequisites

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

Limits

In hybrid network access mode, you cannot switch the network type to Classic Network.

Introduction

While you switch the network type of an ApsaraDB for MongoDB instance from Classic Network to VPC, you can choose to retain the internal endpoints on the classic network for up to 120 days. In hybrid network access mode, the instance supports access from ECS instances in both the classic network and VPC.

In hybrid network access mode, you can switch the network types of ECS instances and other Alibaba Cloud services from Classic Network to VPC until all services are deployed in a VPC.

Procedure

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

5. In the left-side navigation pane, click **Database Connection**.

6. In the **Intranet Connection - Classic Network** section, click **Switch to VPC**.

7. In the **VPC** dialog box that appears, configure related parameters.
   - Specify **VPC** and **VSwitch**.
   - Turn on **Retain the connection address of the classic network**.
   - Set **Expiration Time (Days)**.

8. Click **OK**.
9.8 Change the retention period of internal endpoints on the classic network of an ApsaraDB for MongoDB instance

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

Prerequisites

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

Context

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target ApsaraDB for MongoDB subscription instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its instance ID.
5. In the left-side navigation pane, click Database Connection.
6. In the Retained Classic Network Address section, click Change Expiration Time.
7. In the dialog box that appears, select a retention period.

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

8. Click OK.

9.9 Apply for a public endpoint for an ApsaraDB for MongoDB instance

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

Apply for a public endpoint for a standalone instance
Apply for a public endpoint for a replica set instance
Apply for a public endpoint for a sharded cluster instance

References

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

9.10 Release a public connection string

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

Precautions

- You can release one or more public connection strings of the Mongos, shard, and Configserver nodes for a sharded cluster instance.
- After the public connection string is released for an instance or node, you cannot connect to the instance or node through the original public connection string.
- After the public connection string is released, we recommend that you delete the corresponding public IP address from the whitelist to ensure data security. For more information, see Configure a whitelist.

Standalone and replica set instances

Note:

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

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

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

**Sharded cluster instances**

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

**Note:**

- For more information about node types, see #unique_79.
- After the public connection string of a shard or Configserver node is released, the public connection strings of the primary and secondary nodes are released.

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

<table>
<thead>
<tr>
<th>Node type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>db</strong></td>
<td>The shard node.</td>
</tr>
<tr>
<td><strong>cs</strong></td>
<td>The Configserver node.</td>
</tr>
<tr>
<td><strong>mongos</strong></td>
<td>The Mongos node.</td>
</tr>
</tbody>
</table>

**Note:**

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

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

10.1 Configure a whitelist for an ApsaraDB for MongoDB instance

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

Prerequisites

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

Context

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

Configure an IP address whitelist

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the target instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose Data Security > Whitelist Settings.
6. Configure an IP address whitelist.

To manually modify an IP address whitelist, follow these steps:

a) Find the target IP address whitelist, and choose \textit{Manually Modify} in the \textit{Operation} column.

b) Enter IP addresses or Classless Inter-Domain Routing (CIDR) blocks.

**Note:**

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

c) Click OK.

To load the private IP addresses of ECS instances to an IP address whitelist, follow these steps:

a) Find the target IP address whitelist, and choose > **Import ECS Intranet IP** in the **Operation** column.

![Image of IP address whitelist](image)

b) From the displayed private IP addresses of ECS instances created by the current account, select the target IP addresses and add them to the IP address whitelist.
c) Click OK.

**Note:**
For easy O&M and access control, we recommend that you add an ECS security group. For more information, see Add an ECS security group.

**Add an ECS security group**

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

**Note:**
After you add an ECS security group, all its ECS instances can access the ApsaraDB for MongoDB instance either over an internal network or over the Internet. For access over an internal network, the two types of instances must have the same network type. If the
network type is VPC, they must be in the same VPC. For access over the Internet, you must have applied for a public endpoint for the ApsaraDB for MongoDB instance.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the target instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose Data Security > Whitelist Settings.
7. In the dialog box that appears, select the target ECS security group.

Note:

- You can add only one ECS security group to an ApsaraDB for MongoDB instance. After you add an ECS security group, all its ECS instances can access the ApsaraDB for MongoDB instance either over an internal network or over the Internet. For access over an internal network, the two types of instances must have the same network type. If the network type is VPC, they must be in the same VPC. For access over the
Internet, you must have applied for a public endpoint for the ApsaraDB for MongoDB instance.

- If you move your pointer over an ECS security group, you can view its name and description. If you move your pointer over VPC, you can view the VPC ID. This way, you can quickly find the target ECS security group.

**Delete an IP address whitelist or ECS security group**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the target instance.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > Whitelist Settings**.
6. Delete an IP address whitelist or ECS security group.

   To delete an IP address whitelist, follow these steps:
   a) Find the target IP address whitelist, and choose **Delete Whitelist Group** in the **Operation** column.

**Note:**
You cannot delete the default IP address whitelist.

b) In the message that appears, click **OK**.

To delete an ECS security group, follow these steps:

a) Click **Clear**.

b) In the message that appears, click **OK**.

**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

**10.2 Configure audit logging for an ApsaraDB for MongoDB instance**

This topic describes how to configure audit logging for an ApsaraDB for MongoDB instance. This feature records all operations you have performed on the databases of your instance. With audit logs, you can perform operations such as fault analysis, behavior analysis, and security audit on databases.

**Prerequisites**

The instance is a replica set or sharded cluster instance.
Precautions

- After you enable audit logging, the index optimization feature is also enabled. For more information, see [Index optimization of ApsaraDB for MongoDB](https://example.com).
- If the instance is a sharded cluster instance, you cannot manually select the types of database operations for auditing. In this case, the operation types admin, slow, query, insert, update, and delete are selected by default.
- The default retention period of audit logs is 30 days.

Enable audit logging

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

7. Click **OK**.

Query audit logs

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click **Replica Set Instances** or **Sharding Instances**.
4. Find the instance and click its ID.
5. In the left-side navigation pane, choose **Data Security > Audit Log**.
6. You can use a word or record in a collection (Keyword), or the database name (DB),
database account name (User), start time, and end time to query the audit logs.

Select operation types for auditing

Note:
You can only perform this operation when the instance is a replica set instance.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances.
4. Find the instance and click its ID.
5. In the left-side navigation pane, choose Data Security > Audit Log.
6. Click Audit Log Filter Setting.
7. In the dialog box that appears, select the operation types for auditing.

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

Note:
If audit logging was enabled for ApsaraDB for MongoDB instances before July 2018, the default operation types for auditing are admin, slow, insert, update, delete, and command. If you want to include the query operations, select query in audit settings. For more information, see Select operation types for auditing.

8. Click OK.

Disable audit logging

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the instance and click its ID.
5. In the left-side navigation pane, choose Data Security > Audit Log.
6. Click Disable Audit Log.

Note:
- After you disable audit logging, the index optimization feature is also disabled.
- After you disable audit logging, logs are no longer collected and stored audit logs are deleted.

7. In the message that appears, click OK.

10.3 Configure SSL encryption for an ApsaraDB for MongoDB instance

This topic describes how to configure Secure Sockets Layer (SSL) encryption for an ApsaraDB for MongoDB instance. SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity. After enabling SSL encryption, you must install SSL CA certificates on your application. CA is short for certification authority.

Prerequisites

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

When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your application is configured to reconnect to the instance after it is disconnected.

Note:
When an instance is restarted, all its nodes are restarted in turn and each node has a brief disconnection of about 30 seconds. If the instance houses more than 10,000 collections, the brief disconnections last longer.

Precautions

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

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Replica Set Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose Data Security > SSL.
6. Perform one of the following operations as needed.
When you enable or disable SSL encryption or update SSL CA certificates for an instance, the instance is restarted. Plan your operations in advance and make sure that your application is configured to reconnect to the instance after it is disconnected.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Prerequisite</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable SSL encryption</td>
<td>The SSL encryption status is Disabled.</td>
<td>Turn on SSL Status. In the message that appears, click OK.</td>
</tr>
<tr>
<td>Update an SSL CA certificate</td>
<td>The SSL encryption status is Enabled.</td>
<td>Click Update Certificate. In the message that appears, click OK.</td>
</tr>
<tr>
<td>Download an SSL CA certificate file</td>
<td>The SSL encryption status is Enabled.</td>
<td>Click Download Certificate and download an SSL CA certificate file to your computer.</td>
</tr>
<tr>
<td>Disable SSL encryption</td>
<td>The SSL encryption status is Enabled.</td>
<td>Turn off SSL Status. In the message that appears, click OK.</td>
</tr>
</tbody>
</table>

References

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

10.4 Configure TDE

Transparent data encryption (TDE) encrypts and decrypts data files in real time when they are written or read. It encrypts data files when they are written to disks, and decrypts data files when they are loaded to the memory from disks. TDE does not increase the size of data files. You can use TDE without modifying your application that uses ApsaraDB for MongoDB. To enhance data security, you can enable the TDE feature for an instance in the ApsaraDB for MongoDB console.

Prerequisites

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

Note:

Before enabling TDE, you can create a pay-as-you-go instance of MongoDB 4.0 to test the compatibility between your application and the database version. You can release the instance after the test is completed.
Impact

- When you enable TDE, your instance is restarted, and any application that has connected to the instance will be disconnected from it. We recommend that you enable TDE during off-peak hours and ensure that your application can reconnect to the instance upon disconnection.
- TDE increases the CPU usage of your instance.
- You cannot restore TDE-encrypted collections to a user-created MongoDB database through physical backup files. To restore TDE-encrypted collections to a user-created MongoDB database, use logical backup files.

Note

- You cannot disable TDE after it is enabled.
- Currently, you can enable TDE for an instance and disable encryption for a collection as required.

Note:

In special business scenarios, you can choose not to encrypt a collection when creating it. For more information, see Disable encryption for a collection.

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
3. Find the instance and click the instance ID.
4. In the left-side navigation pane, choose Data Security > TDE.
5. Turn on **TDE Status** to enable TDE.

![TDE Status](image)

6. In the **Restart Instance** dialog box that appears, click **OK**.

The instance status changes to **Modifying TDE**. After the status changes to **Running**, TDE is enabled.

**Disable encryption for a collection**

After you enable TDE, all new collections are encrypted. In special business scenarios, you can choose not to encrypt a collection when creating it. To create a collection with encryption disabled, follow these steps:

1. Connect to your instance through the mongo shell. For more information, see the guide for connecting to a replica set instance or a sharded cluster instance.

2. Run the following command to create a collection with encryption disabled:

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

   **Note:**
10.5 Use the mongo shell to connect to an ApsaraDB for MongoDB database in SSL encryption mode

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

Prerequisites

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

  Note:
  If the database version of the instance is earlier than required versions, you must upgrade the database version. For more information, see Upgrade the database version of an ApsaraDB for MongoDB instance.

- SSL encryption is enabled for the instance. For more information, see Configure SSL encryption for an ApsaraDB for MongoDB instance.

- Mongo shell 3.0 or later is installed on the local server or ECS instance from which you want to connect to the database. For more information about the installation procedure, visit Install MongoDB.

- The IP address of the local server or the ECS instance is added to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist for an ApsaraDB for MongoDB instance.

Precautions

After you enable SSL encryption for an instance, the CPU utilization of this instance is significantly increased. We recommend that you enable it only when necessary. For example, you can enable SSL encryption when you connect to an ApsaraDB for MongoDB instance over the Internet.
**Procedure**

The following example uses a Linux system.

1. Download an SSL CA certificate package. For more information, see [Configure SSL encryption for an ApsaraDB for MongoDB instance](#).

2. Decompress the package and upload the certificate files to the local server or the ECS instance with the mongo shell installed.

   **Note:**
   For this example, upload the .pem file to the `/root/sslcafile/` directory of the local server.

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

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

   **Note:**
   - `<host>`: the connection string (including the port number) of the primary or secondary node in the ApsaraDB for MongoDB instance. For more information, see #unique_45.
   - If you want to connect to a database of the ApsaraDB for MongoDB instance over the Internet, apply for a public endpoint for this instance. For more information, see [Apply for a public endpoint for an ApsaraDB for MongoDB instance](#).
   - If you want to connect to a database of the ApsaraDB for MongoDB instance over an internal network, make sure that the ApsaraDB for MongoDB instance has the same network type as the ECS instance. If the network type is VPC, make sure that the two instances are in the same VPC.
   - `<username>`: the username you use to log on to a database of the ApsaraDB for MongoDB instance. The initial username is root. We recommend that you do not log on to a database as the root user in a production environment. You can create users
and grant permissions to the users as needed. For more information, see Manage MongoDB users through 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.
- `<sslCAFile_path>`: the path of the SSL CA certificate files.

Example:

```
```

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

**Note:**

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

**Common connection scenarios**

- Connect 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

**10.6 SSL connection sample code for MongoDB drivers**

ApsaraDB for MongoDB supports `sslAllowConnectionsWithoutCertificates` to allow you to establish SSL connections to MongoDB clients without a certificate. However, you need to configure the CA to verify the server certificate and ignore host name verification.

For more information about how to configure SSL encryption, see Configure SSL encryption for an ApsaraDB for MongoDB instance.

**Node.js**

**Related link:** MongoDB Node.js Driver
Sample code

Add `/? ssl = true` to the end of the MongoDB client URI, set `sslCA` to the path of the CA certificate, and set `checkServerIndentity` to false to ignore host name verification.

```javascript
var MongoClient = require('mongodb'). MongoClient;
var f = require('util').format,
    fs = require('fs');

// Read the CA certificate.
var ca = [fs.readFileSync(__dirname + '/path/to/ca.pem')];

// Connect to the MongoClient and validate the certificate returned from the server.
MongoClient.connect("mongodb://host01:27017,host02:27017,host03:27017/?
  replicaSet=myreplset&ssl=true", {
  server: {
    sslValidate:true,
    checkServerIndentity:false, # Ignore host name verification.
    sslCA:ca
  }
}, function(err, db) {
  db.close();
});
```

**PHP**

**Related link:** [MongoDB PHP Driver](#)

**Sample code**

Use `MongoDB\Client::__construct` to create a client instance, including three groups of parameters: `$uri`, `$uriOptions`, and `$driverOptions`.

```php
function __construct($uri = 'mongodb://127.0.0.1/', array $uriOptions = [], array $driverOptions = [])
```

In `$uriOptions`, set `ssl` to true to enable SSL connection. In `$driverOptions`, set `ca_file` to the path of the CA certificate. Set `allow_invalid_hostname` to true to ignore host name verification.

```php
<? php
$client = new MongoDB\Client(
  'mongodb://host01:27017,host02:27017,host03:27017',
  [ 'ssl' => true,
    'replicaSet' => 'myReplicaSet'
  ],
  [ "ca_file" => "/path/to/ca.pem",
    "allow_invalid_hostname" => true
  ]
);
```
Java

Related link: MongoDB Java Driver

Sample code

In MongoClientOptions, set sslEnabled to true to enable SSL connection. Set sslInvalidHostNameAllowed to true to ignore host name verification.

```java
import com.mongodb.MongoClientURI;
import com.mongodb.MongoClientOptions;
MongoClientOptions options = MongoClientOptions.builder().sslEnabled(true).sslInvalidHostNameAllowed(true).build();
MongoClient client = new MongoClient("mongodb://host01:27017,host02:27017,host03:27017/? replicaSet=myreplset", options);
```

Run a keytool command to specify the CA certificate.

```bash
keytool -importcert -trustcacerts -file <path to certificate authority file>
   -keystore <path to trust store> -storepass <password>
```

Set Java Virtual Machine (JVM) system properties to specify the correct trust store and key store.

```java
System.setProperty("javax.net.ssl.trustStore","/trust/mongoStore.ts");
System.setProperty("javax.net.ssl.trustStorePassword","StorePass");
```

Python

Related link: MongoDB Python Driver

Sample code

Set ssl to True to enable SSL connection, set ssl_ca_certs to the path of the CA certificate, and set ssl_match_hostname to False to ignore host name verification.

```python
import ssl
from pymongo import MongoClient
uri = "mongodb://host01:27017,host02:27017,host03:27017/? replicaSet=myreplset"
client = MongoClient(uri,
   ssl=True,
   ssl_ca_certs='ca.pem',
   ssl_match_hostname=False)
```

C

Related link: MongoDB C Driver

Sample code
Add /? ssl = true to the end of the MongoDB client URI. Use `mongoc_ssl_opt_t` to set SSL options and set `ca_file` to the path of the CA certificate. Set `allow_invalid_hostname` to false to ignore host name verification.

```c
mongoc_client_t *client = NULL;
client = mongoc_client_new (
    "mongodb://host01:27017,host02:27017,host03:27017/? replicaSet=myreplset&ssl=true");
const mongoc_ssl_opt_t *ssl_default = mongoc_ssl_opt_get_default ();
mongoc_ssl_opt_t ssl_opts = { 0 };

/* Optionally copy a certificate in a custom trust directory or file; otherwise the default is used. */
memcpy (&ssl_opts, ssl_default, sizeof ssl_opts);
ssl_opts.ca_file = "/path/to/ca.pem"
ssl_opts.allow_invalid_hostname = false
mongoc_client_set_ssl_opts (client, &ssl_opts);
```

**C++**

**Related link:** MongoDB C++ Driver

**Sample code**

Add /? ssl = true to the end of the MongoDB client URI. Use `mongocxx::options::ssl` to set SSL parameters and set `ca_file` to the path of the CA certificate.

```cpp
#include <mongocxx/client.hpp>
#include <mongocxx/uri.hpp>
#include <mongocxx/options/client.hpp>
#include <mongocxx/options/ssl.hpp>

mongocxx::options::client client_options;
mongocxx::options::ssl ssl_options;

// If the server certificate is not signed by a well-known CA,
// you can set ca_file to a custom CA certificate.
ssl_options.ca_file("/path/to/ca.pem");

client_options.ssl_opts(ssl_options);

auto client = mongocxx::client{
    uri("mongodb://host01:27017,host02:27017,host03:27017/? replicaSet=myreplset&ssl=true"), client_opts};
```

**Scala**

**Related link:** MongoDB Scala Driver

**Sample code**

Currenty, you cannot ignore host name verification for the MongoDB C++ driver.
The MongoDB Scala driver uses the underlying support for SSL provided by Netty to support SSL connections to MongoDB servers. In MongoClientOptions, set sslEnabled to true to enable SSL connection and set sslInvalidHostNameAllowed to true to ignore host name verification.

```scala
import org.mongodb.scala.connection.{ NettyStreamFactoryFactory, SslSettings}

MongoClientSettings.builder()
  .sslSettings(SslSettings.builder()
    .enabled(true)
    .invalidHostNameAllowed(true)
    .build())
  .streamFactoryFactory(NettyStreamFactoryFactory())
  .build()
val client: MongoClient = MongoClient("mongodb://host01:27017,host02:27017,host03:27017/? replicaSet=myreplset")
```

Run a keytool command to specify the CA certificate, which is the same as the method for Java.

```
keytool -importcert -trustcacerts -file <path to certificate authority file>
  -keystore <path to trust store> -storepass <password>
```

Set JVM system properties to specify the correct trust store and key store.

```
System.setProperty("javax.net.ssl.trustStore","/trust/mongoStore.ts");
System.setProperty("javax.net.ssl.trustStorePassword","StorePass");
```

**Golang**

**Related links**: MongoDB Golang Driver and crypto/tls package

**Sample code**

The MongoDB Golang driver uses the underlying support for SSL provided by the crypto/tls package to support SSL connections to MongoDB servers. Use Config to set SSL options. Set RootCAs to specify the CA certificate and set InsecureSkipVerify to true to ignore host name verification.

```go
import (  
  "crypto/tls"  
  "crypto/x509"  
  "gopkg.in/mgo.v2"
)

rootPEM, err := ioutil.ReadFile("path/to/ca.pem")
roots := x509.NewCertPool()
ok := roots.AppendCertsFromPEM([]byte(rootPEM))
tlsConfig := &tls.Config{
  RootCAs: roots,
  InsecureSkipVerify: true
}
```
url := "mongodb://host01:27017,host02:27017,host03:27017/?replicaSet=myreplset&ssl=true"
dialInfo, err := ParseURL(url)
dialInfo.DialServer = func(addr *ServerAddr) (net.Conn, error) {
    return tls.Dial("tcp", addr.String(), tlsConfig)
}

session, err := DialWithInfo(dialInfo)
if err != nil {
    panic(err)
}

session.Close()
11 Monitoring and alerting

11.1 View monitoring information

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

Precautions

If you receive an alert message from Alibaba Cloud, such as a message indicating that your CPU utilization is higher than 80%, you must filter and troubleshoot the nodes of an instance when you view monitoring information of the instance for error diagnosis and analysis.

Procedure

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

   ![Note:](image.png)
By default, the monitoring data of the last day is displayed. You can also select a time range to view historical monitoring data.

- Standalone instances: Only the monitoring data of primary nodes is displayed.
- Replica set instances: You can specify to display the monitoring data of the primary or secondary node at the top of the page.
- Sharded cluster instances: You can specify to display the monitoring data of a Mongos, shard, or Configserver node at the top of the page.

**Note:**
Mongos nodes have IDs prefixed with `s-`. Shard nodes have IDs prefixed with `d-`. Configserver nodes have IDs suffixed with `-cs`.

### Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage</td>
<td>The CPU utilization of the instance.</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>The memory usage of the instance.</td>
</tr>
<tr>
<td>IOPS Usage</td>
<td>The IOPS usage of the instance, including</td>
</tr>
<tr>
<td></td>
<td>- Data disk IOPS usage.</td>
</tr>
<tr>
<td></td>
<td>- Log disk IOPS usage.</td>
</tr>
<tr>
<td>IOPS Usage</td>
<td>The percentage of the IOPS used by the instance to the maximum available IOPS.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Disk Space Usage</td>
<td>The total disk space used by the instance, including</td>
</tr>
<tr>
<td></td>
<td>• Total usage.</td>
</tr>
<tr>
<td></td>
<td>• Data disk usage.</td>
</tr>
<tr>
<td></td>
<td>• Log disk usage.</td>
</tr>
<tr>
<td>Disk Space Usage</td>
<td>The percentage of the total space used by the instance to the maximum available space.</td>
</tr>
<tr>
<td>Opcounters</td>
<td>Operation QPS of the instance, including</td>
</tr>
<tr>
<td></td>
<td>• The number of insert operations.</td>
</tr>
<tr>
<td></td>
<td>• The number of query operations.</td>
</tr>
<tr>
<td></td>
<td>• The number of delete operations.</td>
</tr>
<tr>
<td></td>
<td>• The number of update operations.</td>
</tr>
<tr>
<td></td>
<td>• The number of getmore operations.</td>
</tr>
<tr>
<td></td>
<td>• The number of command operations.</td>
</tr>
<tr>
<td>Connections</td>
<td>The current number of connections to the instance.</td>
</tr>
<tr>
<td>Cursors</td>
<td>The number of cursors used by the instance, including</td>
</tr>
<tr>
<td></td>
<td>• The number of currently used cursors.</td>
</tr>
<tr>
<td></td>
<td>• The number of expired cursors.</td>
</tr>
<tr>
<td>Network</td>
<td>The network traffic of the instance, including</td>
</tr>
<tr>
<td></td>
<td>• The inbound traffic.</td>
</tr>
<tr>
<td></td>
<td>• The outbound traffic.</td>
</tr>
<tr>
<td></td>
<td>• The number of processed requests.</td>
</tr>
<tr>
<td>Global Lock</td>
<td>The queues waiting for the global lock for the instance, including</td>
</tr>
<tr>
<td></td>
<td>• The queues waiting for the global read lock for the instance.</td>
</tr>
<tr>
<td></td>
<td>• The queues waiting for the global write lock for the instance.</td>
</tr>
<tr>
<td></td>
<td>• The total queues waiting for the global lock.</td>
</tr>
<tr>
<td>WiredTiger</td>
<td>The cache metric of the WiredTiger engine for the instance, including</td>
</tr>
<tr>
<td></td>
<td>• The size of data read to the cache.</td>
</tr>
<tr>
<td></td>
<td>• The capacity of the disk with data written from the cache.</td>
</tr>
<tr>
<td></td>
<td>• The configured maximum available disk capacity.</td>
</tr>
</tbody>
</table>
11.2 Set the monitoring granularity

ApsaraDB for MongoDB provides an optional monitoring granularity setting feature for you to set a finer granularity for collecting routine monitoring data and correctly locating O&M problems.

**Notes**

- Standalone instances do not support this feature.
- The database version of ApsaraDB for MongoDB instances must be MongoDB 3.4 (upgraded to the latest minor database version) or MongoDB 4.0.

**Note:**
The monitoring granularity of every second depends on the latest minor database version of ApsaraDB for MongoDB 3.4. The latest minor database version is compatible with all earlier minor database versions.

- ApsaraDB for MongoDB instances whose database version is MongoDB 3.2 do not support the monitoring granularity of every second. You need to upgrade their database version to MongoDB 3.4 to use this feature. For more information, see Upgrade the database version.
- For ApsaraDB for MongoDB instances created after December 5, 2017 with the database version of MongoDB 3.4, you can directly set the monitoring granularity to every second. All metrics take effect immediately.
- For ApsaraDB for MongoDB instances created before December 5, 2017 with the database version of MongoDB 3.4, if they have been restarted once since December 5, 2017, they can be automatically upgraded to the latest minor database version. If they have never been restarted since December 5, 2017, you need to restart them during off-peak hours. All metrics take effect after their restart.
- Currently, ApsaraDB for MongoDB provides the monitoring granularity of every second free of charge.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Every second</th>
<th>Every 300s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space Usage</td>
<td>N/A</td>
<td>Supported in MongoDB 3.2, MongoDB 3.4, and MongoDB 4.0.</td>
</tr>
<tr>
<td>Disk Space Usage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Metric

<table>
<thead>
<tr>
<th>Metric</th>
<th>Every second</th>
<th>Every 300s</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage</td>
<td>Supported in MongoDB 3.4 (upgraded to the latest minor database version) and MongoDB 4.0.</td>
<td></td>
</tr>
<tr>
<td>Memory Usage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOPS Usage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opcounters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cursors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GlobalLock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WiredTiger</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances based on the architecture of the target instance.
4. Locate the target instance and click its instance ID.
5. In the left-side navigation pane, click Monitoring Info.
6. On the Monitoring Info page that appears, click Monitor Granularity Setting.
7. In the **Monitor Granularity Setting** dialog box that appears, select a monitoring granularity.

8. Click **OK**.

11.3 Set alert rules

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

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. Locate the target instance and click its instance ID.
4. In the left-side navigation pane, click **Alarm Rules**.
5. Click **Set Alarm Rule** to jump to the CloudMonitor console.
6. In the upper-right corner of the CloudMonitor console, click **Create Alarm Rule**.
7. On the **Create Alarm Rule** page that appears, specify related resources.

![Related Resource](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Products</td>
<td>The architecture of the instance.</td>
</tr>
<tr>
<td></td>
<td>• ApsaraDB for MongoDB-Instance Copy</td>
</tr>
<tr>
<td></td>
<td>• ApsaraDB for MongoDB-Cluster Instance</td>
</tr>
<tr>
<td></td>
<td>• ApsaraDB for MongoDB-Single node instance</td>
</tr>
</tbody>
</table>

**Note:**
If you select **ApsaraDB for MongoDB-Cluster Instance**, you need to select the mongos nodes and shards to be monitored for **Mongos** and **Shard**, respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Range</td>
<td>• If you select <strong>All Resources</strong>, the alerting service sends an alert notification when any ApsaraDB for MongoDB instances match alert rules.</td>
</tr>
<tr>
<td></td>
<td>• If you select <strong>Instances</strong>, the alerting service sends an alert notification when any selected ApsaraDB for MongoDB instances match alert rules.</td>
</tr>
<tr>
<td>Region</td>
<td>The region where the instance is located.</td>
</tr>
<tr>
<td>Instances</td>
<td>The ID of the instance to be monitored. You can select multiple instance IDs.</td>
</tr>
</tbody>
</table>

8. Set alert rules and configure notification methods. For more information about parameters, see **Manage alert rules**.

**Note:**
If you have not created alert contacts in CloudMonitor, see **Manage alert contacts and alert contact groups**.

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

For more information about metrics, see **ApsaraDB for MongoDB in Cloud service monitoring**.
12 Parameter settings

12.1 Configure database parameters for an ApsaraDB for MongoDB instance

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

Prerequisites

The instance is a standalone or replica set instance.

Precautions

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

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

Procedure

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

### Note:
On the parameter list, you can check whether the instance needs to be restarted after you modify a parameter.
7. In the Modify Parameter dialog box that appears, modify parameters as needed.

![Modify Parameter dialog box]

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

8. Click OK.

### 12.2 View the parameter modification history

You can log on to the ApsaraDB for MongoDB console or call the DescribeParameterModificationHistory operation to view the parameter modification history.

**Prerequisites**

The target instance must be a standalone or replica set instance. Sharded cluster instances do not support this feature.
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. In the left-side navigation pane, click Replica Set Instances.
4. Locate the target instance and click its instance ID.
5. In the left-side navigation pane, choose Parameters > Modification History.

On the Modification History page that appears, modification records of the last 24 hours are displayed by default. You can also specify a time range to query parameter modification records.
13 Primary/Secondary failover

13.1 Trigger a primary/secondary failover for a replica set instance

An ApsaraDB for MongoDB replica set instance consists of three nodes by default. ApsaraDB for MongoDB provides addresses for you to connect to the primary node and a secondary node. The other secondary node is hidden as a backup to guarantee high availability. If a node is faulty, the high availability system of ApsaraDB for MongoDB automatically triggers a primary/secondary failover to guarantee the availability of the instance. In addition, you can manually trigger a primary/secondary failover for an ApsaraDB for MongoDB instance in scenarios such as routine disaster recovery drills.

Context

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

**Note:**

- You can trigger a primary/secondary failover only for replica set and sharded cluster instances, but not for standalone instances due to their single-node architecture.
- After you trigger a primary/secondary failover for an instance, the instance may be disconnected for 30s once. You need to ensure that your applications can automatically re-establish a connection.
- You can trigger a primary/secondary failover only for instances in the normal running status.

Procedure

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

![Node List Table](image)

6. In the **Failover** dialog box that appears, click **OK**.

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

The failover takes about 1 minute. Then, the instance returns to normal.

**Note:**
If you have used the address of the primary node to connect to an instance, you are connecting to a secondary node after a failover and you have no write permission on the instance. In this case, you need to use the address of the new primary node to connect to the instance to obtain read and write permissions. For more information, see **Obtain the replica set instance connection information**.

### 13.2 Trigger a primary/secondary failover for a shard of a sharded cluster instance

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

**Notes**

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

**Note:**

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

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. In the left-side navigation pane, click **Sharding Instances**.
4. Locate the target instance and click its instance ID.
5. In the **Shard List** area, locate the target shard and choose **Failover** in the Operation column.

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

6. In the **Failover** dialog box that appears, click **OK**.
7. The failover takes about 1 minute. You can repeat the preceding procedure to trigger a primary/secondary failover for other shards of the same sharded cluster instance as required.
14 Log management

14.1 View slow query logs

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

Prerequisites

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the instance and click its ID.
5. In the left-side navigation pane, choose Logs > Slow Query Logs.
6. View slow query logs based on instance types:

   - Replica set instances: You can select the database name and time range to query the corresponding slow query logs.

   ![Replica Set Instances Slow Query Logs]

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

   ![Sharded Cluster Instances Slow Query Logs]
14.2 View error logs

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

Prerequisites

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

Procedure

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

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

- Query the error logs of a Mongos node.
  
  You can select the Mongos node ID and time range to query the error logs.

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

14.3 View operation logs

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

Prerequisites

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region of the instance.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the instance and click its ID.
5. In the left-side navigation pane, choose Logs > Running Logs.
6. Perform the following steps based on instance types:

- **Replica set instances**: You can select the node role and time range to query the corresponding operation logs.

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

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

  - Query the operation logs of a Mongos node.

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

  - Query the operation logs of a shard node.

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

15.1 Configure automatic backup for an ApsaraDB for MongoDB instance

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

Precautions

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. Alternatively, you can select the database version 4.0 when you create the instance.

Automatic backup

- ApsaraDB for MongoDB stores backup files in Object Storage Service (OSS) to reduce the storage space usage of ApsaraDB for MongoDB instances.
- Standalone instances can only use snapshot backup, which affects their I/O performance in the backup process.

Note:
Snapshot backup retains the status of disk data at a specific point in time.

- Replica set and sharded cluster instances support physical backup.

Note:
With physical backup, all physical database files in an ApsaraDB for MongoDB instance are backed up. Physical backup runs on the hidden node of an ApsaraDB for MongoDB instance, and does not affect the I/O performance of the primary and secondary nodes. Backing up a large volume of data may take a long time.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its ID.

5. In the left-side navigation pane, click **Backup and Recovery**.

6. Click **Backup Settings**.

7. In the dialog box that appears, configure the following parameters as needed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retention Days</td>
<td>The number of days for which you want to retain backup data. It can only be seven days.</td>
</tr>
<tr>
<td>Backup Time</td>
<td>The hour at which you want to perform the backup task. We recommend that you select an off-peak hour.</td>
</tr>
<tr>
<td>Day of Week</td>
<td>The backup cycle. You can select one or more than one day in a week.</td>
</tr>
</tbody>
</table>

8. Click **OK**.
15.2 Manually back up an ApsaraDB for MongoDB instance

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

Impacts

- ApsaraDB for MongoDB stores its backup files in Object Storage Service (OSS) to reduce the storage space usage of ApsaraDB for MongoDB instances.
- Standalone instances only support snapshot backup, which decreases their I/O performance.
- Physical backup and logical backup run on the hidden nodes of replica set instances, and do not affect the performance of the primary and secondary nodes. Backing up a large volume of data may take a long time.

Backup methods

- Snapshot backup: The status of disk data at a specific point in time is retained.
- Physical backup: Physical database files of an ApsaraDB for MongoDB instance are backed up.
- Logical backup: mongodump is used to logically back up each database.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the source instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the source instance and click its ID.
5. In the upper-right corner of the page that appears, click Backup Instance.
6. In the dialog box that appears, specify Backup Method.

Note:
- If the instance is a standalone instance, you can only select Snapshot Backup.
• If the instance is a replica set or sharded cluster instance, you can select **Logical Backup** or **Physical Backup**.

7. Click **OK**.

References

**Restoration solution overview**
16 Data recovery

16.1 Restoration solution overview

The data restoration feature of ApsaraDB for MongoDB can minimize any losses caused by incorrect operations on databases. ApsaraDB for MongoDB provides many data restoration solutions to meet requirements in different scenarios.

Restore data to ApsaraDB for MongoDB instances

<table>
<thead>
<tr>
<th>Method</th>
<th>Instance type</th>
<th>Scenario</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an instance from a backup</td>
<td>Standalone or replica set instance</td>
<td>Applicable to the scenarios where the entire instance is restored and data timeliness is not a key requirement.</td>
<td>A new instance is created based on the backup data and data is restored to the new instance.</td>
</tr>
<tr>
<td>Create an instance based on a point in time</td>
<td>Replica set or sharded cluster instance</td>
<td>Applicable to the scenarios where multiple databases or the entire instance is restored. The data at a specified point in time will be restored.</td>
<td>Note: The created instance will be billed. For more information, see Billing items and pricing.</td>
</tr>
<tr>
<td>Restore data to a new ApsaraDB for MongoDB instance from databases</td>
<td>Replica set instance</td>
<td>Applicable to the scenarios where one or more databases are quickly restored. For example, a data set or document is deleted by mistake.</td>
<td></td>
</tr>
</tbody>
</table>

Note:
The created instance will be billed. For more information, see Billing items and pricing.
### Restore data to user-created databases

You can download backup files of ApsaraDB for MongoDB to your server and recover the data to a user-created database. This feature is applicable to scenarios such as business testing or data analysis.

<table>
<thead>
<tr>
<th>Method</th>
<th>Instance type</th>
<th>Scenario</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore logical backup files of ApsaraDB for MongoDB to user-created databases</td>
<td>Replica set instance</td>
<td>N/A</td>
<td>Restoring data directly to the current instance poses high risks. We recommend that you restore the data by using the following feature: Restore data to a new ApsaraDB for MongoDB instance by point in time or Create an instance from a backup. After the data is validated, migrate the data back to the original instance through DTS.</td>
</tr>
<tr>
<td>Restore physical backup files of ApsaraDB for MongoDB to user-created databases</td>
<td>Replica set instance</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

### 16.2 Restore data to a new ApsaraDB for MongoDB instance from databases

This topic describes how to restore one or more databases of an ApsaraDB for MongoDB instance to a new ApsaraDB for MongoDB instance by using a backup created at a specific point in time. This method is ideal for quick data restoration.

#### Background information

Instances created after March 26, 2019 support the restoration of one or more databases. For information about when this feature will be available to instances created before March 26, 2019, follow the official website.

#### Prerequisites

- The instance is created after March 26, 2019.
• The instance is located in the region China (Qingdao), China (Beijing), China (Zhangjiakou-Beijing Winter Olympics), China (Hohhot), China (Hangzhou), China (Shanghai), China (Shenzhen), or Singapore.

• The instance is a replica set instance.

• The database version of the instance is 3.4 or 4.0.

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

• The storage engine of the instance is WiredTiger.

• The backup file list of the instance contains the backup files of the databases you want to restore.

**Precautions**

• You can only restore databases from physical backups.

• The time required varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the new instance changes to **Running**, the restoration is complete.

**Billing**

While you restore one or more databases, the system creates an instance and you are charged for the new instance. For more information, see #unique_16.

**Note:**
If you want to restore the databases to a pay-as-you-go instance, make sure that your account has sufficient balance.

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the source instance resides.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the source instance and click its ID.
5. In the left-side navigation pane, click **Backup and Recovery**.
6. On the **Backup and Recovery** page, click **Create Instance By Time Point**.
7. In the dialog box that appears, configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select recovery time point</td>
<td>Select a point in time from which you want to restore data. You can select any time from the last seven days.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>The time you select must be earlier than the current time and later than the time when the source instance was created.</td>
</tr>
<tr>
<td>Select databases to recover</td>
<td>• <strong>All Databases:</strong> If you select this option, all databases in the source instance are restored.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Select Databases:</strong> If you select this option, only selected databases are restored.</td>
</tr>
<tr>
<td></td>
<td>You can directly select the databases you want to restore, or click <strong>Enter Databases</strong> to enter the names of the databases.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>If you want to restore more than one database, separate the database names with commas (,) when you enter them.</td>
</tr>
</tbody>
</table>

8. Click **OK**.

9. On the Instance Purchase page that appears, select a billing method for the new instance.

**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.

10. Configure the new instance. For more information, see #unique_9.

Note:
- You cannot change Region, Database Version, Storage Engine, and Replication Factor for the new instance.
- To make sure that the new instance has sufficient space to store the data restored from the source instance, we recommend that you select a storage capacity greater than or equal to that of the source instance.

11. Click Buy Now.
12. On the Confirm Order page, read and select ApsaraDB for MongoDB Agreement of Service, and complete the payment as prompted.

16.3 Restore data to a new ApsaraDB for MongoDB instance by backup set

This topic describes how to restore data to a new ApsaraDB for MongoDB instance by backup set. This method is ideal for data restoration and verification.

Prerequisites
- The source instance is a standalone or replica set instance.
- You can only select a backup set from the last seven days.

Billing

This method creates an instance and you are charged for the new instance. For more information, see Billing items and pricing.

Note:
If you want to restore data to a pay-as-you-go instance, make sure that your account has sufficient balance.
Procedure

1. Log on to the ApsaraDB for MongoDB console.

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

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

4. Find the source instance and click its ID.

5. In the left-side navigation pane, click Backup and Recovery.

6. On the Backup and Recovery page that appears, find the target backup set and choose > Create Instance from Backup Point in the Operation column.

7. On the Instance Purchase page that appears, select a billing method for the new instance.

   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 the 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.

8. Configure the new instance. For more information, see #unique_9.

Note:
• You cannot change Region, Database Version, Storage Engine, and Replication Factor for the new instance.
• To make sure that the new instance has sufficient space for restoration, we recommend that you set the storage capacity greater than or equal to that of the source instance.

9. Click Buy Now.

10. Read and select ApsaraDB for MongoDB Agreement of Service, and complete the payment as prompted.

Note:
The time required to restore data to a new instance by backup set varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the new instance changes to Running, the restoration is complete.

16.4 Restore data to a new ApsaraDB for MongoDB instance by point in time

This topic describes how to restore data to a new ApsaraDB for MongoDB instance by point in time. This method is ideal for data restoration and verification.

Prerequisites
• The source instance is a replica set or sharded cluster instance.
• You can only select a point in time from the last seven days.

Billing
This method creates an instance and you are charged for the new instance. For more information, see Billing items and pricing.

Note:
If you want to restore data to a pay-as-you-go instance, make sure that your account has sufficient balance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the source instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the source instance and click its ID.

5. In the left-side navigation pane, click **Backup and Recovery**.

6. On the **Backup and Recovery** page, click **Create Instance By Time Point**.

7. In the **Create Instance By Time Point** dialog box that appears, select the target point in time and click **OK**.

   **Note:**
   To ensure data integrity and accuracy, do not select the latest point in time (usually the latest hour) if the instance is a sharded cluster instance. If you do so, restoration fails.

8. On the **Instance Purchase** page that appears, select a billing method for the new instance.

   **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 the 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.

9. Configure the new instance. For more information, see #unique_9 or #unique_10.
10. Click **Buy Now**.

11. Read and select **ApsaraDB for MongoDB Agreement of Service**, and complete the payment as prompted.

**Note:**
The time required to restore data to a new instance by point in time varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the new instance changes to **Running**, the restoration is complete.

### 16.5 Restore data to your current ApsaraDB for MongoDB instance

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

**Prerequisites**

The instance is a replica set instance with three nodes.

**Impacts**

If you restore data to your current instance, all existing data is overwritten and cannot be restored.

**Warning:**

This operation is risky. We recommend that you restore data to a new ApsaraDB for MongoDB instance by point in time or backup set. Then, verify the data, and migrate the data back to the source instance by using Data Transmission Service (DTS). For more information, see **Restore data to a new ApsaraDB for MongoDB instance by point in time** or **Restore data to a new ApsaraDB for MongoDB instance by backup set**.
Precautions

The time required to restore data to your current instance varies depending on factors such as the data volume, task queue status, and network conditions. When the status of the instance changes to Running, the restoration is complete.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the source instance resides.
3. In the left-side navigation pane, click Replica Set Instances.
4. Find the source instance and click its ID.
5. In the left-side navigation pane, click Backup and Recovery.
6. On the Backup and Recovery page that appears, find the target backup set and choose > Data Recovery in the Operation column.

7. In the Recover Backup Instance message that appears, click OK.

Note:
The instance status becomes Restoring from Backup. You can click Refresh to update the instance status. When the instance status changes to Running, the restoration is complete.

16.6 Restore data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using logical backup

This topic describes how to restore the data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using logical backup. Data restoration uses the
mongorestore command. You must have created a logical backup and downloaded the logical backup file to the server where you plan to run the mongorestore command.

**Prerequisites**

- The ApsaraDB for MongoDB instance is a replica set instance.
- The ApsaraDB for MongoDB instance and the user-created MongoDB databases run the same database version.

**Context**

Full logical backup uses the mongodump command to back up a database. During the backup process, you can still perform read/write operations on the database.

**Note:**

Full logical backup runs on the hidden node of the ApsaraDB for MongoDB instance, which does not affect the read/write performance of the primary and secondary nodes. Backing up a large volume of data may take a long time.

**Procedure**

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the source instance resides.
3. In the left-side navigation pane, click **Replica Set Instances**.
4. Find the source instance and click its ID.
5. In the upper-right corner of the page, click **Backup Instance**.
6. In the **Backup Instance** dialog box that appears, select the **Logical Backup** method.
7. Click **OK**. Then, wait until the backup is complete.
8. On the **Backup and Recovery** page, find the target logical backup file and choose **Download** in the Operation column.
9. Copy the downloaded file to the server where you plan to run the mongorestore command, and run the following command to import the file to user-created MongoDB databases:

```
mongorestore -h <hostname> --port <server port> -u <username> -p <password> --drop --gzip --archive=<backupfile> -vvvv --stopOnError
```

Parameter description:

- `<hostname>`: the address of the server where the user-created MongoDB databases reside. If you also run the mongorestore command on this server, enter 127.0.0.1.
- `<server port>`: the port number of the user-created MongoDB databases.
- `<username>`: the username you use to log on to the user-created MongoDB databases.
- `<password>`: the password of the preceding account.
- `<backupfile>`: the name of the logical backup file you downloaded.

Example:

```
mongorestore -h 127.0.0.1 --port 27017 -u root -p xxxxxxxx --drop --gzip --archive=hins1111_data_20190710.ar -vvvv --stopOnError
```

16.7 Recover physical backup data in a user-created MongoDB instance

16.7.1 Download the physical backup data of a replica set instance

You can download the physical backup data of a replica set instance based on the backup time and recover the downloaded backup data in the user-created MongoDB instance.

Prerequisites

Standalone instances do not support this feature. You can create an instance from a specified backup to recover data. For more information, see Restore data to a new ApsaraDB for MongoDB instance by backup set.

Context

After setting an automatic backup policy, you can back up data for replica set instances in physical backup mode.
A physical backup is carried out on the hidden secondary node of an ApsaraDB for MongoDB instance. Therefore, it does not affect the I/O performance of the primary and secondary nodes. It may take a long time to back up a large amount of data. You need to wait patiently.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.
3. In the left-side navigation pane, click Replica Set Instances.
4. Locate the target instance and click its instance ID.
5. In the left-side navigation pane, click Backup and Recovery.
6. On the Backup and Recovery page that appears, locate the target physical backup and choose > Download.

Note:
After downloading the backup file, you can follow the instructions in Recover ApsaraDB for MongoDB physical backup data in a user-created MongoDB instance to recover data.

16.7.2 Restore data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using physical backup

This topic describes how to restore the data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using physical backup. Before you start the data
Data recovery, you must download physical backup files of the ApsaraDB for MongoDB instance in the ApsaraDB for MongoDB console.

Prerequisites

- The ApsaraDB for MongoDB instance is a replica set instance.
- The TDE feature is disabled for the ApsaraDB for MongoDB instance.
- The storage engine of the ApsaraDB for MongoDB instance is WiredTiger or RocksDB. If the storage engine is TerarkDB, use logical backup to restore data to user-created MongoDB databases. For more information, see Restore data of an ApsaraDB for MongoDB instance to user-created MongoDB databases by using logical backup.

Note:
If the storage engine is RocksDB, you must compile and install a MongoDB program for user-created MongoDB databases to support the RocksDB storage engine.

Database version requirements

<table>
<thead>
<tr>
<th>ApsaraDB for MongoDB instance</th>
<th>User-created MongoDB database</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>3.2 or 3.4</td>
</tr>
<tr>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Formats of physical backup files

<table>
<thead>
<tr>
<th>Physical backup file format</th>
<th>File extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tar compressed package</td>
<td>.tar.gz</td>
<td>ApsaraDB for MongoDB instances that were created before March 26, 2019 have physical backup files in .tar format.</td>
</tr>
<tr>
<td>xbstream file package</td>
<td>_qp.xb</td>
<td>ApsaraDB for MongoDB instances that were created on or after March 26, 2019 have physical backup files in .xb format.</td>
</tr>
</tbody>
</table>

Note:
You must decompress the two types of packages in different ways. For more information, see Download and decompress a physical backup file.
Environment preparation

The following procedure uses an Alibaba Cloud ECS instance created from an image that contains Ubuntu 16.04 (64-bit). For more information, see Create an ECS instance.

**Note:**

- MongoDB of the required version is installed on the ECS instance. For more information about the installation procedure, see official MongoDB documentation.
- The /path/to/mongo/data directory of the ECS instance is used for the user-created MongoDB databases. This directory is empty.

**Download and decompress a physical backup file**

1. Download the physical backup file of an ApsaraDB for MongoDB instance. You can also use the wget command to download it.
2. Copy the downloaded physical backup file to the /path/to/mongo/data/ directory.
3. Decompress the physical backup file.

   - If the extension of the physical backup file is .tar.gz, for example, the file name is hins20190412.tar.gz, run the following commands to decompress the file:

   ```bash
cd /path/to/mongo/data/
tar xzvf hins20190412.tar.gz
```

   **Figure 16-1: Decompression result**

   ![Decompression result](image)

   - If the extension of the physical backup file is _qp.xb, for example, the file name is hins20190412_qp.xb, follow these steps:

     a. Install the percona-xtrabackup tool.

     ```bash
     apt-get update
     apt install percona-xtrabackup
     ```

     b. Visit QuickLZ to download the qpress tool package.

     c. Decompress the qpress tool package and install the tool.

     ```bash
     tar xvf qpress-11-linux-x64.tar
     chmod 775 qpress
     ```
d. Decompress the physical backup file. In this example, the file is `hins20190412_qp.xb`.

```shell
cd /path/to/mongo/data/
cat hins20190412_qp.xb | xbstream -x -v
innobackupex --decompress --remove-original /path/to/mongo/data
```

Figure 16-2: Decompression result

Restore data to user-created MongoDB databases in standalone mode

1. Create a mongod.conf configuration file in the /path/to/mongo directory.

   ```shell
touch mongod.conf
```

2. Customize the mongod.conf configuration file to meet the startup requirements of the MongoDB program.

   You can select a configuration template based on the storage engine of the ApsaraDB for MongoDB instance, and copy the selected configuration template to the mongod.conf configuration file. In this file, enable standalone startup and authorization.

   • WiredTiger

   ```
systemLog:
   destination: file
   path: /path/to/mongo/mongod.log
   logAppend: true
security:
   authorization: enabled
storage:
   dbPath: /path/to/mongo/data
directoryPerDB: true
net:
   port: 27017
unixDomainSocket:
   enabled: false
processManagement:
   fork: true
   pidFilePath: /path/to/mongo/mongod.pid
```

Note:
By default, ApsaraDB for MongoDB uses the WiredTiger storage engine, with the directoryPerDB option enabled. Therefore, this option is set to true in the preceding configuration.

- RocksDB

```yaml
systemLog:
  destination: file
  path: /path/to/mongo/logs/mongod.log
  logAppend: true
security:
  authorization: enabled
storage:
  dbPath: /path/to/mongo/data
  engine: rocksdb
net:
  port: 27017
unixDomainSocket:
  enabled: false
processManagement:
  fork: true
  pidFilePath: /path/to/mongo/logs/mongod.pid
```

3. Use the mongod.conf configuration file to start the MongoDB program.

```
/usr/bin/mongod -f /path/to/mongo/mongod.conf
```

4. In the ECS instance, log on to a user-created MongoDB database by using the mongo shell.

```
mongo --host 127.0.0.1 -u <username> -p <password> --authenticationDatabase admin
```

Parameter description:

- `<username>`: a username of databases in the ApsaraDB for MongoDB instance. The initial username is root.
- `<password>`: the password of the preceding account.

**Start the MongoDB program in replica set mode**

By default, the physical backup file of an ApsaraDB for MongoDB instance contains the replica set configuration of this instance. You must start the MongoDB program in standalone mode. Otherwise, user-created MongoDB databases may be inaccessible.

If you want to start the MongoDB program in replica set mode, you must restore data to user-created MongoDB databases in standalone mode first, and then follow these steps:

1. In the ECS instance, log on to a user-created MongoDB database by using the mongo shell.
2. Delete the original replica set configuration.

   use local
db.system.replset.remove({})

3. Shut down the MongoDB program.

   use admin
db.shutdownServer()

4. Add replication-related configurations to the mongod.conf configuration file in the /path/to/mongo/ directory. For more information about the commands used, visit Deploy a Replica Set.

5. Use the mongod.conf configuration file to start the MongoDB program.

   /usr/bin/mongod -f /path/to/mongo/mongod.conf

6. Add nodes to the replica set and initialize the replica set.

   Note:
   This step uses an rs.initiate() command. For more information about the command, visit rs.initiate().
17 CloudDBA

17.1 Index optimization of ApsaraDB for MongoDB

This topic introduces the index optimization feature of ApsaraDB for MongoDB. If you forget to add indexes or use incorrect indexes, query statements may run slowly or time out. This increases the CPU utilization of your ApsaraDB for MongoDB instance. This feature detects slow queries caused by the index issues, and provides the optimal indexes for these slow queries to improve the performance of ApsaraDB for MongoDB.

Prerequisites

- The instance is a replica set or sharded cluster instance.
- The region of the instance is China (Hangzhou), China (Shanghai), China (Shenzhen), China (Qingdao), or China (Beijing).
- The audit logging feature is enabled for the instance.

Rules to generate an index optimization report

ApsaraDB for MongoDB automatically generates index optimization reports for statements executed from 00:00:00 to 24:00:00 every day. You can also select a time range to generate an index optimization report by using the Custom Analysis function in the ApsaraDB for MongoDB console.

Note:

- Query statements whose execution time exceeds 100 ms are defined as slow queries.
- Index optimization reports are retained for seven days and are automatically deleted after the seventh day.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the page, select the region where the target instance resides.
3. In the left-side navigation pane, click Replica Set Instances or Sharding Instances.
4. Find the target instance and click its ID.
5. In the left-side navigation pane, choose CloudDBA > Index Optimization.
6. Click Custom Analysis.
7. In the Custom Analysis dialog box that appears, select a time range and click **OK**.

![Custom Analysis dialog box](image)

8. Find the index optimization report you want to view, and click **View Detail** in the **Operation** column.

![Index optimization report](image)
9. In the dialog box that appears, view the index diagnosis details.

10. Add or optimize indexes based on the details.
18 Zone-disaster restoration solution

18.1 Create a multi-zone replica set instance

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

Precautions

- You can only create a multi-zone replica set instance in China (Hangzhou), China (Beijing), China (Shenzhen) and Singapore.
- When you create a multi-zone replica set instance, you must set **Replication Factor** to **Three Nodes Replicaset**.

**Note:**

If you need more nodes, you can reset the number of nodes after you create the instance.

**Node deployment policies**

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-zone deployment</td>
<td>The system deploys the primary, secondary, and hidden nodes in one zone.</td>
</tr>
</tbody>
</table>

![Diagram of single-zone deployment](image)
## Zone-disaster restoration solution

### Deployment

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-zone deployment</td>
<td>The system deploys the primary, secondary, and hidden nodes in three different zones.</td>
</tr>
</tbody>
</table>

### Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click Replica Set Instances.
3. On the Replica Set Instances page, click Create Instance.
4. On the Create Instance page, set Region to China (Hangzhou), China (Beijing), China (Shenzhen) or Singapore, and select a multi-zone from the Zone drop-down list.
5. Configure other parameters. For more information, see #unique_9.

6. Click Buy Now. The Confirm Order page appears.

7. Read and select ApsaraDB for MongoDB Agreement of Service, and complete the payment as prompted.

References

You can use the Service Availability function to view the distribution of nodes in a replica set instance across zones. You can also switch the node roles of the instance based on your business deployment. This way, your applications can connect to the nodes closest to them. For more information, see Switch node roles.

18.2 Create a multi-zone sharded cluster instance

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

Precautions

You can only create a multi-zone sharded cluster instance in China (Hangzhou), China (Beijing), China (Shenzhen) and Singapore.

Node deployment policies

If you use the single-zone deployment solution, the system deploys all components of the sharded cluster instance to one zone. If you use the multi-zone deployment solution, the system deploys all components to three different zones.

• mongos is evenly deployed across all data centers. At least two mongos are deployed at a time, with each to one zone. When you add a third mongos, the system deploys it to the third zone. Each new mongos added later is deployed to one of the three zones in turn.
The primary, secondary, and hidden shards in each shard are not deployed to the three zones in sequence. The deployment of these shards may change when manual switchover or HA failover between primary and secondary shards is triggered.

Figure 18-1: Deployment policy for the components in a multi-zone sharded cluster instance

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the left-side navigation pane, click Sharding Instances.
3. On the Sharding Instances page, click Create Instance.
4. On the **Create Instance** page, set **Region** to **China (Hangzhou), China (Beijing), China (Shenzhen)** or **Singapore**, and select a multi-zone from the Zone drop-down list.

5. Configure other parameters. For more information, see #unique_10.

6. Click **Buy Now**. The **Confirm Order** page appears.

7. Read and select **ApsaraDB for MongoDB Agreement of Service**, and complete the payment as prompted.

**References**

You can use the Service Availability function to view the distribution of components in a sharded cluster instance across zones. You can also switch the component roles of the instance based on your business deployment. This way, your applications can connect to the components closest to them. For more information, see **Switch node roles**.