Alibaba Cloud ApsaraDB for MongoDB

User Guide

Issue: 20190910

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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

Legal disclaimer..... I
Generic conventions..... I
1 Preface.....1
2 Quick start..... 2
3 Logon and logoff.....3
Log on to the ApsaraDB for MongoDB console.....3
Log off from the ApsaraDB for MongoDB console.....3
4 Billing management.....5
4.1 Change the billing method of instances from pay-as-you-go to subscription.....5
4.2 Manually renew a subscription-based instance.....6
4.3 Automatically renew a subscription-based instance.....7
5 Instance connection.....10
5.1 Connect to an ApsaraDB for MongoDB instance through a cross-zone intranet..... 10
5.2 Connect to ApsaraDB for MongoDB instances through a public network.... 11
5.3 Connect the local client to ApsaraDB for MongoDB instances through an SSL VPN tunnel..... 13
6 Account management..... 19
6.1 Reset the password..... 19
7 Instance management.....21
7.1 Specify a maintenance period..... 21
7.2 Change the configuration.....22
7.3 Change the number of nodes in replica set instances..... 27
7.4 Migrate instances across zones.....29
7.5 Export the list of instances.....33
7.6 Manage the minor database version..... 34
7.7 Upgrade the database version.....36
7.8 Release an instance.....38
7.9 Restart an instance..... 39
8 Network connection management..... 41
8.1 Modify the connection information..... 41
8.2 Switch the network type of an instance..... 43
8.3 Configure a VPC for a new instance.....46
8.4 Configure a hybrid access solution to smoothly switch from a classic network to a VPC.....48
8.5 Apply for a public address..... 50
8.6 Release a public address..... 50

8.7 Modify the expiration time for the classic network....52
9 Data security.... 54
9.1 Configure a whitelist.... 54
9.2 Configure log auditing....56
9.3 Configure SSL encryption....62
9.4 SSL connection sample code for MongoDB drivers.... 65
10 Monitoring and alerting.... 70
10.1 View the monitoring information.... 70
10.2 Set the monitoring granularity.... 72
10.3 Set alert rules.... 74
11 Parameter settings.... 77
11.1 Set database parameters....77
11.2 View the parameter modification history.... 79
12 Primary/Secondary failover....81
12.1 Trigger a primary/secondary failover for a replica set instance.... 81
12.2 Trigger a primary/secondary failover for a shard of a sharded cluster instance....82
13 Data backup....85
13.1 Automatically back up ApsaraDB for MongoDB data.... 85
13.2 Manually back up ApsaraDB for MongoDB data....87
14 Data recovery....89
14.1 Restore a database in ApsaraDB for MongoDB.... 89
14.2 Create an instance based on a backup....92
14.3 Create an instance based on a time point.... 94
14.4 Recover backup data in the current instance....96
14.5 Recover logical backup data in a user-created MongoDB instance.... 97
14.6 Recover physical backup data in a user-created MongoDB instance.... 99
14.6.1 Download the physical backup data of a replica set instance....99
14.6.2 Recover ApsaraDB for MongoDB physical backup data in a user-created MongoDB instance.... 101
15 CloudDBA....105
15.1 Optimize indexes....105
16 Zone-disaster restoration solution....108
16.1 Create a multi-zone replica set instance.... 108
16.2 Create a multi-zone sharded cluster instance....110

1 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

You can manage ApsaraDB for MongoDB instances in the ApsaraDB for MongoDB console, for example, create or connect to an instance. This topic describes how to log on to and log off from the ApsaraDB for MongoDB console.

Prerequisites

Before logging on to the ApsaraDB for MongoDB console, you need to purchase ApsaraDB for MongoDB instances. For more information about how to purchase an instance, see Create an instance. For more information about the billing standards, see ApsaraDB for MongoDB Pricing.

This topic uses a replica set instance as an example to describe how to log on to and log off from the ApsaraDB for MongoDB console. The procedures for logging on to and logging off from the ApsaraDB for MongoDB console for a sharded cluster instance are similar to those for a replica set instance. For more information, see ApsaraDB for MongoDB console in the Sharded Cluster Instance Quick Start.

Log on to the ApsaraDB for MongoDB console

Procedure

1. Use the Alibaba Cloud account that you have used to purchase ApsaraDB for MongoDB instances to log on to the ApsaraDB for MongoDB console.
2. On the page that appears, select the region where the target instance is located to list ApsaraDB for MongoDB instances of the region.
3. Click the target instance ID or choose > Manage in the Operation column to go to the Basic Information page of the target instance. On this page, you can manage accounts, configure a whitelist, and set parameters for the target instance.

Log off from the ApsaraDB for MongoDB console

Context

You can use either of the following methods to log off from the ApsaraDB for MongoDB console:

- **(Recommended) Move the pointer over the avatar in the upper-right corner. On the pop-up menu, click Sign out.**
- **Close your browser.**

4 Billing management

4.1 Change the billing method of instances from pay-as-you-go to subscription

You can change the billing method of an ApsaraDB for MongoDB instance from pay-as-you-go to subscription. Changing the billing method of the instance has no impact on the running of the instance.

Precautions

- The billing method of subscription-based instances cannot be changed to pay-as-you-go. To prevent resource waste, determine whether you need to change the billing method of your resources.
- You cannot release subscription-based instances.
-
- You cannot update the specifications of the instance if you have an unpaid subscription-based instance. You need to cancel this order on the Orders page and change the billing method of the corresponding instance to subscription.

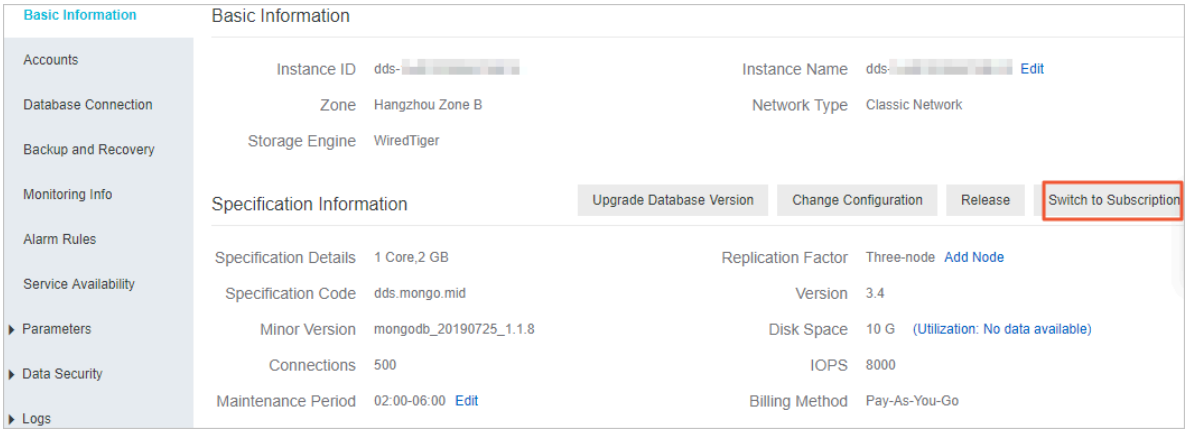
Prerequisites

- The instance is in the running state.
- The billing method of the instance is pay-as-you-go.
- You do not have an unpaid subscription-based instance.
- The instance type cannot be phased-out types (that is, instance types are no longer available for sale). For more information about phased-out instance types, see the Instance types displayed before July 10, 2017. If you need to change the billing method of a phased-out instance type to subscription, change the instance type first. For specific operations, see Change the configuration.

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. Locate the instance and click the instance ID.

5. On the Basic Information page, click Switch to Subscription.



6. On the Confirm Order page, select the Purchase Duration of the instance.
7. Select ApsaraDB for MongoDB Agreement of Service, and click Activate.

> **Note:**
> The system will generate an order for you to switch to the subscription billing method. If this order is not paid or canceled, you cannot purchase new instances or switch the instance billing method to subscription. You can pay for or cancel this order on the Orders page.

8. Select a payment method and click Confirm Payment.

4.2 Manually renew a subscription-based instance

Context

When a subscription-based instance expires, you need to renew the subscription to this instance within seven days or re-create the instance within 8 to 15 days. After the grace period, the instance is released and its data is permanently deleted. When renewing the instance, you can change its configuration. The new configuration takes effect in the new billing period. 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. Select the region where the target instance is located.
3. On the list of instances, choose > Renew in the Operation column corresponding to the target instance to go to its Renew page.

4. Specify Duration. You can select Auto Renew to enable the automatic renewal of the subscription-based instance. This ensures that the instance can be automatically renewed to avoid business interruption.
5. Select ApsaraDB for MongoDB Agreement of Service and click Pay. Follow the payment process to complete the renewal.

4.3 Automatically renew a subscription-based instance

Context

You can enable the automatic renewal of a subscription-based instance. In this case, you do not need to manually renew the instance on a regular basis. This feature also ensures that the instance can be automatically renewed to avoid business interruption.

If you did not enable automatic renewal when purchasing a subscription-based instance, you can also enable this feature in the Alibaba Cloud console. After this feature is enabled, the instance can be automatically renewed based on the selected renewal period. For example, if you select a three-month renewal period, the instance is automatically renewed and billed for another three months each time.

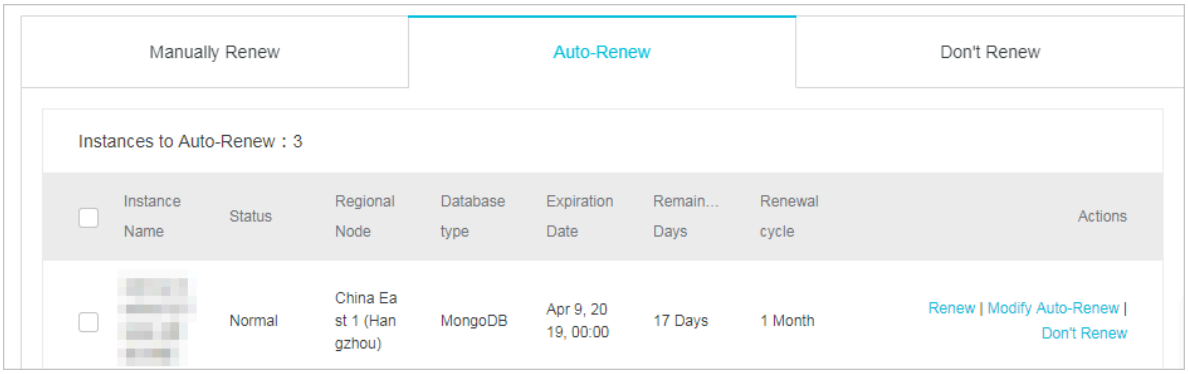
 Note:
When purchasing a subscription-based instance, you can select Auto Renew while specifying Duration.

- Subscription period on a monthly basis: The automatic renewal period is a month.
- Subscription period on a yearly basis: The automatic renewal period is a year.

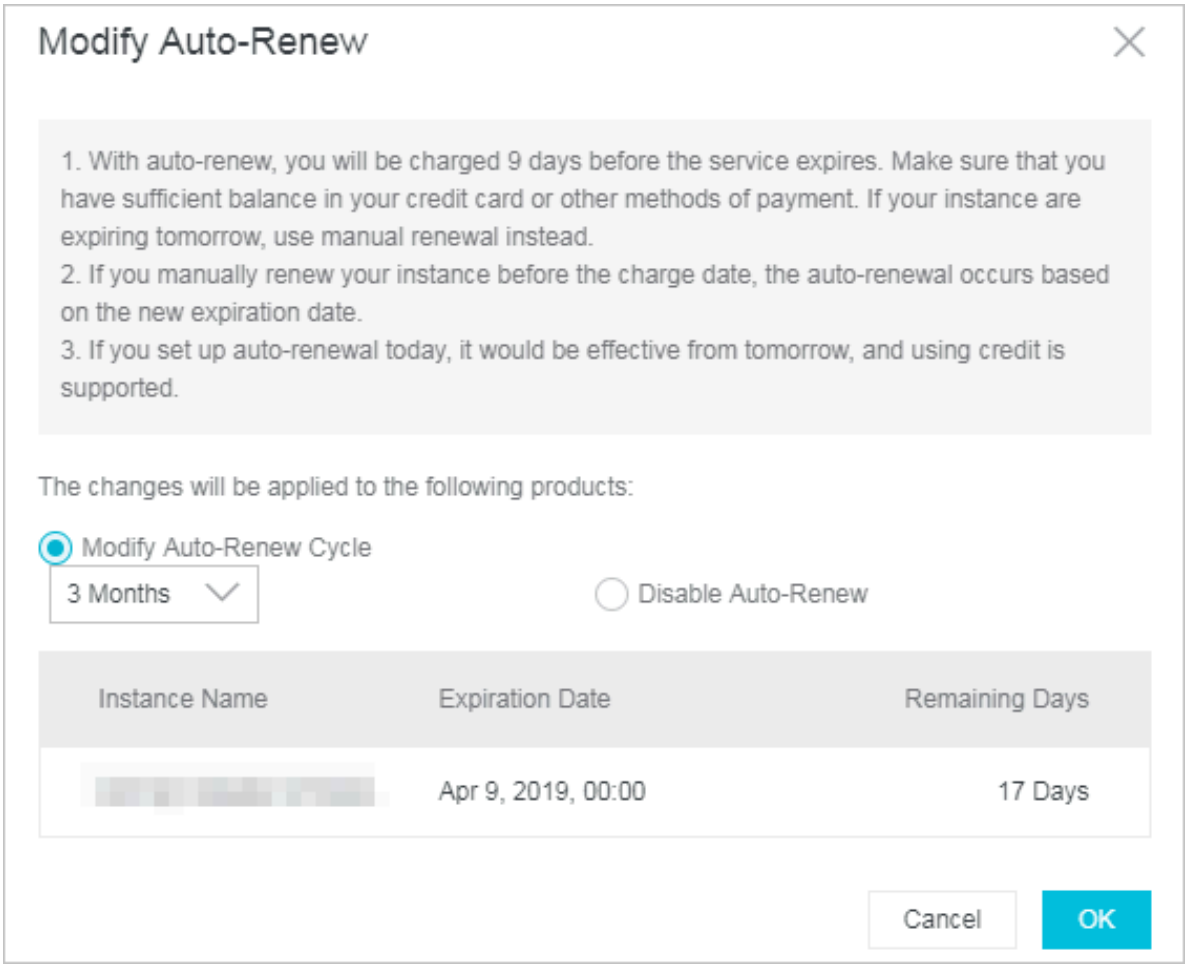
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the top navigation bar, choose Billing Management > Renew to go to the Renew page.
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.



5. Click Enable Auto-Renew in the Actions column corresponding to an instance. The Enable Auto-Renew dialog box appears.



6. Select an automatic renewal period and click OK.
7. On the Auto-Renew tab page, click Renew in the Actions column corresponding to an instance. In the Renew dialog box that appears, renew this instance.

8. On the Auto-Renew tab page, click Don't Renew in the Actions column corresponding to an instance. In the Don't Renew dialog box that appears, click Don't Renew.

Note:

We recommend that you enable the automatic renewal of an instance to ensure that the instance can be automatically renewed to avoid business interruption.

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 ApsaraDB for MongoDB instances through a public network

When you need to connect a local server to an ApsaraDB for MongoDB instance by using the IP address of a public network, you can use the methods described in this topic.

Precautions

This topic is only applicable when you connect a local server to an ApsaraDB for MongoDB instance. If you need to connect to ApsaraDB for MongoDB instances through ECS, you can view the IP addresses of the public network and the internal network on the ECS instance details page.

If you connect to ApsaraDB for MongoDB instances through a public network, security risks may arise. We recommend that you connect to ApsaraDB for MongoDB instances through ECS instances.

Method 1 Locate the public IP address of the local server in an IP address library and connect to the instance

Procedure

1. Access the IP address library of Taobao to query your public IP address.
2. Add the public IP address to the whitelist of the ApsaraDB for MongoDB instance. For more information, see #unique_22.
3. On the local server, log on to the ApsaraDB for MongoDB instance through the mongo shell. For more information, see #unique_23.

> Note:
> You can also log on to ApsaraDB for MongoDB instances through other client tools.

In a case you have added the public IP address of the local server to the whitelist of the ApsaraDB for MongoDB instance but you still fail to connect to the instance. However, if you set the IP address in the whitelist to 0.0.0.0/0, you can connect to the instance. In this case, we recommend you locate the public IP address in connection information. For more information, see Method 2 Locate the public IP address in connection information.

Method 2 Locate the public IP address in connection information

Procedure

1. Add the IP address 0.0.0.0/0 to the whitelist of the ApsaraDB for MongoDB instance. For more information, see #unique_22.

   Note:
   If you add 0.0.0.0/0 to the whitelist, any server can access the ApsaraDB for MongoDB instance. This may pose security risks. Exercise caution when you add 0.0.0.0/0 to the whitelist. Remove 0.0.0.0/0 as soon as you no longer need it.

2. On the local server, log on to the ApsaraDB for MongoDB instance through the mongo shell. For more information, see #unique_23.

   Note:
   When you log on to an ApsaraDB for MongoDB instance through the mongo shell, log on by using the public connection address.

3. Run the following command to query information about the client to which you log on through the mongo shell:

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

   Output examples

   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 the clients:

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

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

More information

If your public IP address is not static and changes frequently, you can use either of the following methods to connect to an ApsaraDB for MongoDB instance:

- Connect to ApsaraDB for MongoDB instances through ECS. For more information, see #unique_23.
- Connect to ApsaraDB for MongoDB instances through a VPN. For more information, see #unique_24.

5.3 Connect the local client to ApsaraDB for MongoDB instances through an SSL VPN tunnel

You can build an SSL VPN tunnel between the management terminal and the VPC in which ApsaraDB for MongoDB instances are deployed. This enables a safe and secure connection to ApsaraDB for MongoDB instances.

Scenario

- The environment in which the client that manages the ApsaraDB for MongoDB database has no static public IP address. In this case, you are required to frequently adjust the IP addresses in the whitelist through the ApsaraDB for MongoDB console. If you fail to delete expired IP addresses, security risks may arise.
- The environment in which the client runs requires high network security. A higher-level of security is required when you connect to AsparaDB for MongoDB instances through a public network.
- When database administrators use a public network to log on to the ApsaraDB for MongoDB database through ECS, they threaten to expose management permissions. Therefore, you are required to separate ECS management permissions from ApsaraDB for MongoDB database permissions.

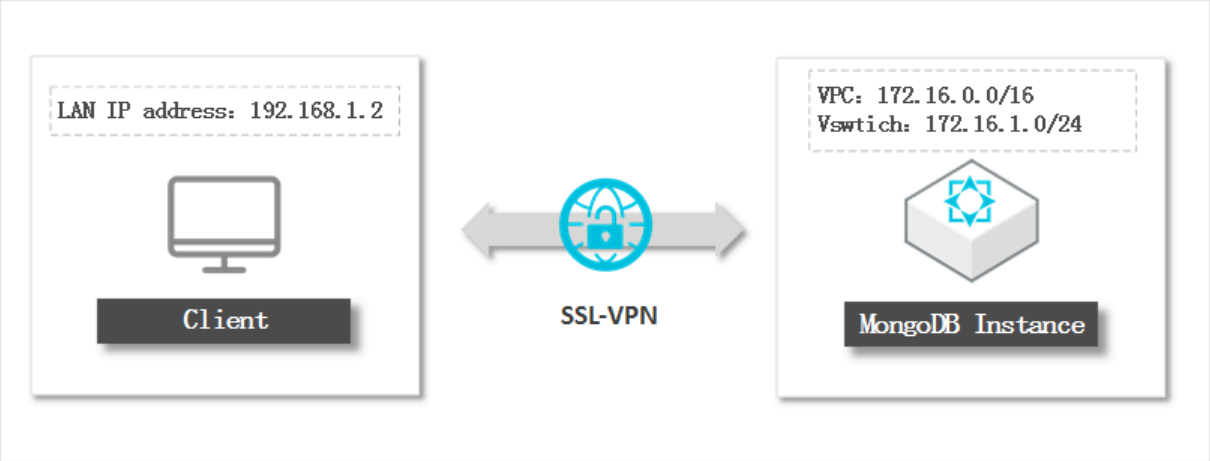
Billing information

Certain fees occur when you create a VPN gateway. For more information, see Pricing.

Prerequisites

- The network type of the ApsaraDB for MongoDB instances is VPC. For more information about how to switch from the classic network to the VPC, see #unique_20/unique_20_Connect_42_section_tp1_1sl_2fb.
- Ensure that the IP address range of the local client and that of an ApsaraDB for MongoDB instance is different.
- Ensure that the local client can access the public network.

Example of environment preparation



Step 1 Create a VPN gateway

1. Log on to the VPC console.
2. In the upper-left corner of the homepage, select the 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 specifications of the VPN gateway as needed.

Parameter	Description
Region	The region in which the VPN gateway is located. Select the region that is the same as that of the ApsaraDB for MongoDB instance.
VPC	Select the VPC to which the ApsaraDB for MongoDB instance belongs.
Peak Bandwidth	Select the bandwidth specifications for the VPN gateway. The bandwidth is that of the public network which is used for the VPN gateway.

Parameter	Description
IPSec-VPN	Specify whether to enable the IPSec-VPN. You can enable this function as needed. You can access the VPN through the terminal. Therefore, select Disable IPSec-VPN. The IPSec-VPN function provides site-to-site connections. You can create an IPSec tunnel to connect an on-premises IDC to a VPC, or connect two VPCs.
SSL VPN	Specify whether to enable the SSL VPN function. You can enable this function as needed. In this case, you need to access the VPN through the terminal. Therefore, select Enable SSL VPN. SSL VPN provides point-to-site VPN connections. You do not need to configure the client gateway. You can access the VPN directly through the terminal.
Billing Cycle	Select the duration of subscription-based instances. The subscription duration can be one to nine months on a monthly basis or one to three years on a yearly basis.
Auto Renewal	Specify whether to enable auto renewal for the instance. · By Month: The auto renewal period is one month. · By Year: The auto renewal period is one year.

6. Click Purchase Now, and follow the instructions to complete the payment.

Step 2 Create an SSL server

1. Log on to the VPC console.
2. In the upper-left corner of the homepage, select the 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 parameters of the SSL server.

Parameter	Description
Name	The name of 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 (-).
VPN Gateway	The associated VPN gateway. Select the VPN gateway created in Step 1 Create a VPN gateway.
Local Network Segment	The local network segment which is the address range to be accessed by the client through the SSL VPN. It can be the network segment of a VPC, a VSwitch, or the network segment of an IDC that is connected with a VPC through a leased line, or a cloud service such as RDS or OSS. The network segment you enter is the network segment address of the VSwitch in a VPC to which an ApsaraDB for MongoDB instance belongs: 172.16.1.0/24. Note: The subnet mask of the local network segment must be 16-bit to 29-bit.
Client Network Segment	The client network segment is the address segment that assigns access addresses to the virtual NICs of a client. When the client accesses the local network through an SSL VPN connection, a VPN gateway assigns an IP address from the specified client network segment to the client. In this case, enter 192.168.100.0/24. Note: Make sure that the client network segment and the Local Network Segment 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 homepage, select the region.
3. In the left-side navigation pane, choose VPN > SSL Clients.
4. On the SSL Clients page, click SSL Client Certificate.

Parameter	Description
Name	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 (-).
SSL Server	Select the SSL server created in Step 2 Create an SSL server.

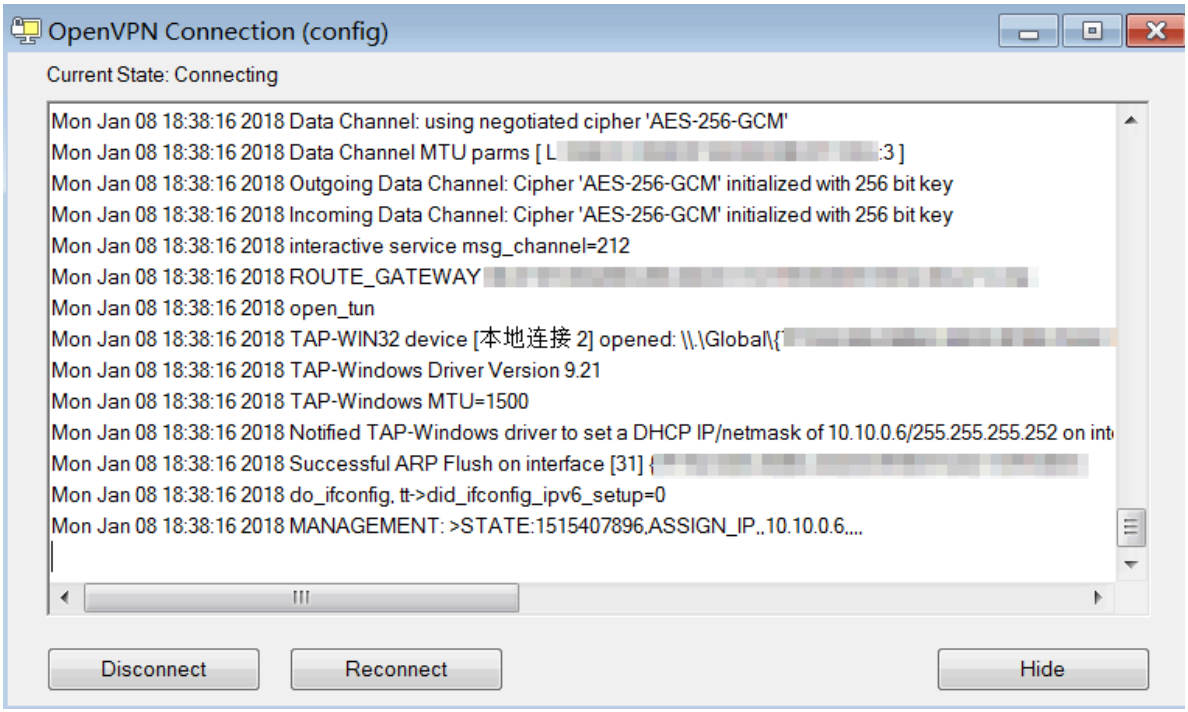
5. Click OK.

Log on to the ApsaraDB for MongoDB database on the client through an SSL VPN tunnel

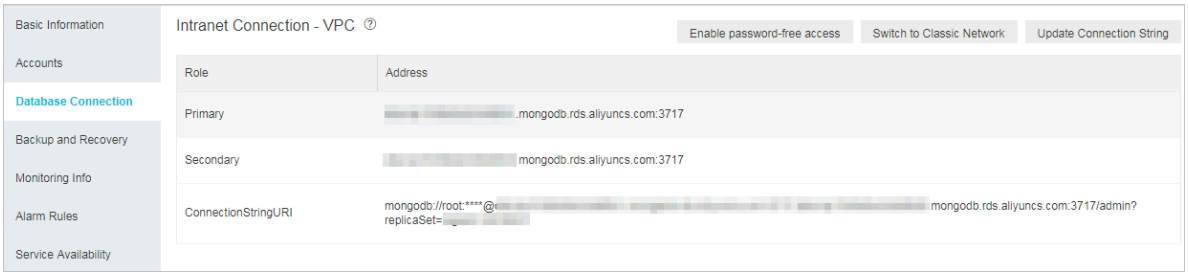
This topic uses Windows as an example. For more information about other operating systems, see the following topics: Connect SSL VPN in the Linux systemand Connect SSL VPN in the Mac system.

1. Log on to the VPC console.
2. In the upper-left corner of the homepage, select the region.
3. In the left-side navigation pane, choose VPN > SSL Clients.
4. To the right of the SSL client instance you have created, click Download to download the generated client certificate.
5. Download and install an OpenVPN client on the client to which you need to connect through the SSL VPN tunnel.
6. Decompress the client certificate that you downloaded in the preceding step and copy it to the config folder of the OpenVPN installation directory.

7. Click Connect to initiate a connection.



8. Add the IP address of the VPC to which the ApsaraDB for MongoDB instance belongs to the whitelist of the ApsaraDB for MongoDB instance. The IP address 172.16.1.0/24 is added to the whitelist of the ApsaraDB for MongoDB instance.
9. Log on to the ApsaraDB for MongoDB console.
10. Obtain the private IP address 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 database.

> **Note:**
> Log on using the private IP address of the ApsaraDB for MongoDB instance.

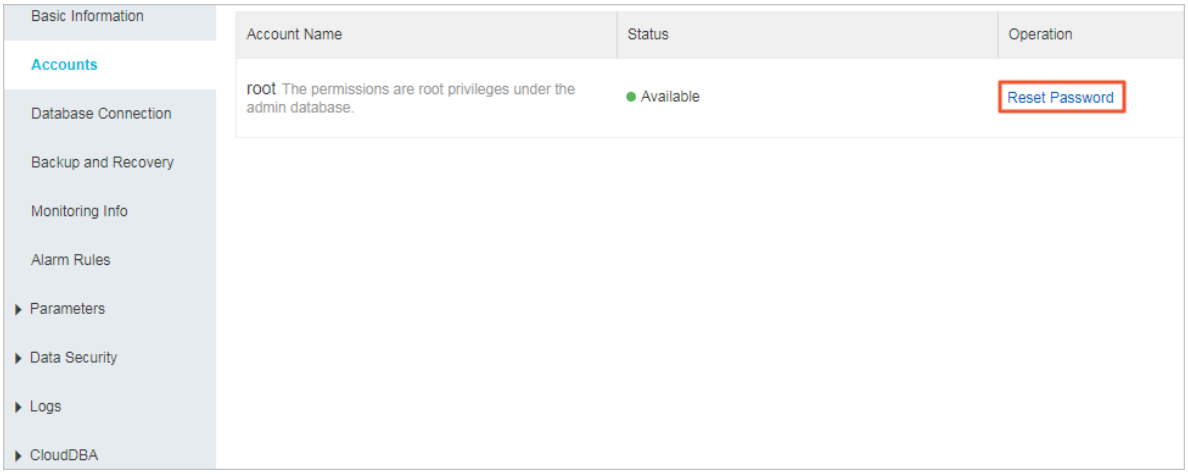
6 Account management

6.1 Reset the password

If you forget your password, need to change the old password, or did not set a password when creating an instance, you can reset the password for the instance.

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 Accounts.



6. Click Reset Password.
7. In the Reset Password dialog box that appears, enter a new password, confirm your password, and click OK.

Note:

- The password must consist of any three types of characters, including uppercase letters, lowercase letters, digits, and special characters. Special characters include exclamation points (!), number signs (#), dollar signs ($),

percent signs (%), carets (^), ampersands (&), asterisks (*), parentheses (()), underscores (_), plus signs (+), hyphens (-), and equal signs (=).

- The password must be 8–32 characters in length.

8. Click OK.

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.

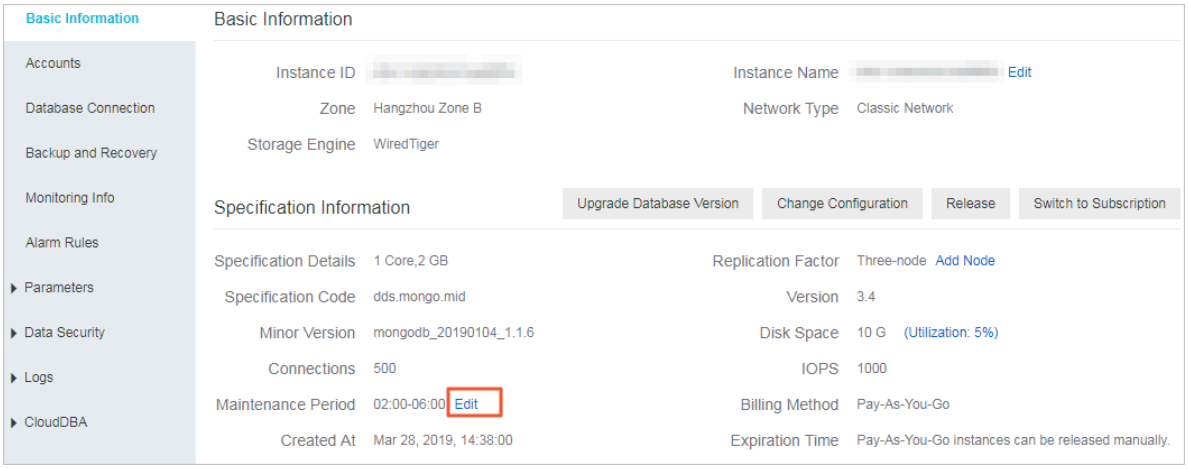
On the day of maintenance, instances enter the Instance being maintained status ahead of the specified maintenance period to guarantee the stability of the maintenance process. You can still connect to instances in this status. In the ApsaraDB for MongoDB console, you cannot change these instances, for 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.



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

7.2 Change the configuration

If the configuration of an instance cannot meet the performance requirements of your applications or is higher than required, you can change the configuration for this instance.

Constraints

Due to the differences among the standalone, replica set, and sharded cluster architectures, you cannot change the architecture of an instance.

Fees

You can upgrade or downgrade the configuration for all ApsaraDB for MongoDB instances. The fees of an instance may change if its configuration is changed. For more information, see Billing items and pricing.

Effective time

- Standalone or replica set instance: When changing the configuration, you can set the effective time for the new configuration.
  - Immediately after data migration: After a configuration change process, the instance immediately enters the Changing Configuration status. The

configuration is successfully changed when the instance status changes to Running.

During some configuration upgrades, the target instance may be disconnected for less than 30s once or twice. You can set the effective time for the configurat ion change as required to avoid an impact on business.

- During the maintenance period: You can set the effective time for the configuration change within a specified period. For more information, see #unique_32.

  Note:
  If an instance is not disconnected during the configuration change, the configuration change can immediately take effect regardless of whether you have set the effective time.

· Sharded cluster instance: You cannot set the effective time for the configuration change. After a configuration change process, the instance immediately enters the Changing Configuration status. The configuration is successfully changed when the instance status changes to Running.

Note:
When an instance is in the Changing Configuration status, you cannot perform most database, account, and network operations for this instance. The completion time of the configuration change depends on various factors such as the network, task queue, and data amount. We recommend that you change the configuration of an instance during off-peak hours or ensure that your applications can automatically re-establish a connection.

Change the configuration of a standalone or replica set instance

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. If the target instance is a Pay-As-You-Go instance:

   Choose  > Change Configuration in the Operation column corresponding to the target instance.

   You can also click the target instance ID or choose  > Manage in the Operation column corresponding to the target instance. On the Basic Information page that appears, click Change Configuration.
4. If the target instance is a subscription-based instance:

   a. Click the target instance ID or choose  > Manage in the Operation column corresponding to the target instance.

   b. On the Basic Information page that appears, click Upgrade or Downgrade.

      You can also choose  > Upgrade in the Operation column corresponding to the target instance.
5. On the Update page, specify Specification and Storage Space for the target instance.

   > Note:
   > - You cannot downgrade the storage space for a standalone instance whose billing method is Pay-As-You-Go.
   > - You cannot downgrade the storage space for a standalone or replica set instance whose billing method is subscription.

   For more information about the specifications and storage space for instances, see #unique_13.

   On the Update page, you can also set the effective time for the configuration change.
6. Select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the configuration change process.

Add nodes to change the configuration of a sharded cluster instance

When adding a mongos node for a sharded cluster instance, you can specify Specification for the mongos node to change the configuration of the instance.

When adding a shard for a sharded cluster instance, you can specify Specification and Storage Space for the shard to change the configuration of the instance.

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. Click the target sharded cluster instance ID.

   You can also choose  > Manage in the Operation column corresponding to the target instance.
4. To add a mongos node, click Add Mongos on the Basic Information page.
5. On the Add Mongos page that appears, specify Specification for the new mongos node.
6. To add a shard, click Add Shard on the Basic Information page.
7. On the Add Shard page that appears, specify Specification and Storage Space for the new shard.

   For more information about the specifications and storage space for instances, see #unique_13.
8. Select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the configuration change process.

Change the configuration of existing nodes to change the configuration of a sharded cluster instance

You can change the specifications of existing mongos nodes or the specifications and storage space of existing shards to change the configuration of a sharded cluster instance.

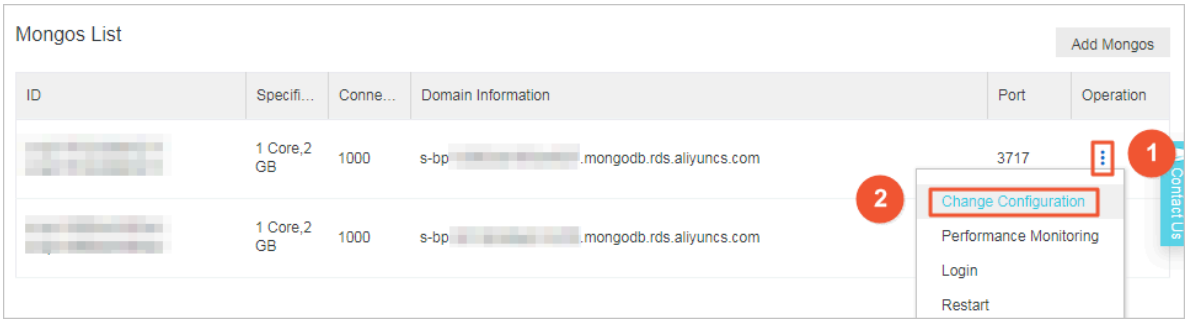
> Note:
> When changing the configuration of a sharded cluster instance, you can only add nodes or change the specifications and storage space of existing nodes. You cannot delete nodes or change configuration items other than the specifications and storage space of existing nodes.

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.

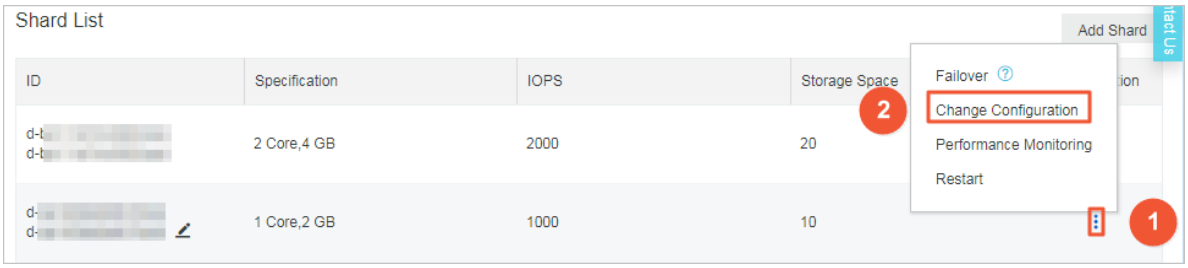
3. Click the target sharded cluster instance ID.

   You can also choose > Manage in the Operation column corresponding to the target instance.

4. To change the configuration of an existing mongos node, do as follows: In the Mongos List area of the Basic Information page, choose > Change Configuration in the Operation column corresponding to the target mongos node.



5. On the Change Configuration Mongos page that appears, specify Specification for the mongos node.
6. To change the configuration of an existing shard, do as follows: In the Shard List area of the Basic Information page, choose > Change Configuration in the Operation column corresponding to the target shard.



7. On the Change Configuration Shard page that appears, specify Specification and Storage Space for the shard.

   Note:

When changing the configuration of an existing shard for a sharded cluster instance whose billing method is subscription, you cannot downgrade the storage space of the shard.

For more information about the specifications and storage space for instances, see #unique_13

8. Select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the configuration change process.

7.3 Change the number of nodes in replica set instances

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. The number of nodes must be kept at least three however you change them. The changes to the node result in a change to the instance billing. 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. Locate the instance and click the instance ID.

5. On the Basic Information page, follow these steps based on the billing method of the instance:
   a) For pay-as-you-go instances, click Upgrade or Downgrade in the Basic Information section.
   b) For subscription-based instances, click Change Configuration in the Basic Information section.
6. On the Change Configuration page, select Replication Factor.

   Note:
   For more information about how to change specifications and storage space, see Change the configuration.
7. Set Migration Time.

   Note:

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

     During some configuration upgrades, the instance may be disconnected for less than 30s once or twice. You can set the effective time for the configuration change as required to avoid negative effects 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 the network of an instance is not disconnected during the configuration change, the configuration change can immediately take effect regardless of whether you have set the migration time.
8. Select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the payment.

What's next

After you add the nodes for the replica set instances, the connection addresses of new nodes (all displayed as Secondary, only different for role IDs) appear in the console. The connection string URI for a high availability connection is also updated. You

can modify the connection address 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.

7.4 Migrate instances across zones

You can migrate instances to other zones within the same region. After the instances are migrated to other zones, the attributes, specifications, and connection addresses of the instances remain unchanged.

Prerequisites

- The instance is a replica set instance.
- The destination zone and the current zone where the instance is located must be in the same region.
- Before you migrate instances in VPCs, ensure that the destination zone has corresponding VSwitches created. For more information about VSwitch creation, see Create a VSwitch.
- Before you migrate instances with a public connection address, release the public connection address. For more information, see #unique_35.

Precautions

- The VPC of instances whose network type is VPC cannot be changed when you migrate the instances to other zones.
- The period for migration is related to several factors such as the network, task queue, and the amount of data to be migrated. We recommend that you modify configurations during off-peak hours.
- Services may be disconnected for 30 seconds during migration across zones. Ensure that your application is configured to reconnect to the database after the application is disconnected.
- Migration across zones causes the changes to virtual IP addresses (VIPs), such as 172.16.88.60. If the application is connected to the VIP of the database, the connection with the database is disconnected. We recommend that you use the following URI: mongodb://root:****@dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717/admin? replicaSet=mgset-132xxx.

This guarantees the high availability of connections. For more information, see #unique_26.

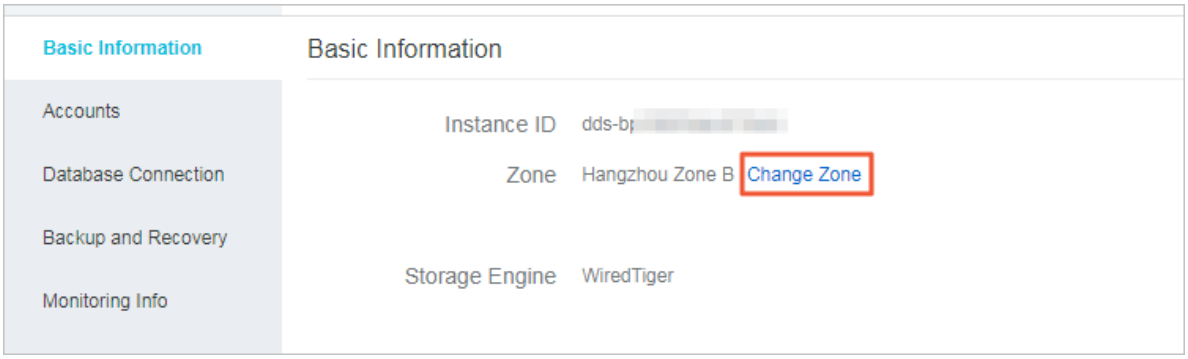
Supported migration types and scenarios

Supported migration type	Scenario
Migrate instances from one zone to another zone	Migrate ApsaraDB for MongoDB instances to the zone to which the ECS instances belong. In the same zone, the ECS instances connect to the ApsaraDB for MongoDB instances through the intranet, which minimizes network latency.
Migrate instances from one zone to multiple zones	Improve the disaster recovery capability of instances to achieve disaster recovery across data centers. Deploy the three nodes of a replica set instance to three zones within the same region. This method helps the instances tolerate disasters at higher levels. For example, instances in a single zone can tolerate server- and rack-related faults, whereas instances in multiple zones can tolerate data center-related faults. Note: For information about the node deployment strategy of replica set instances in multiple zones, see #unique_36/unique_36_Connect_42_section_wjr_qpj_wgb.
Migrate instances from multiple zones to a single zone	Meet the needs of specific functions.

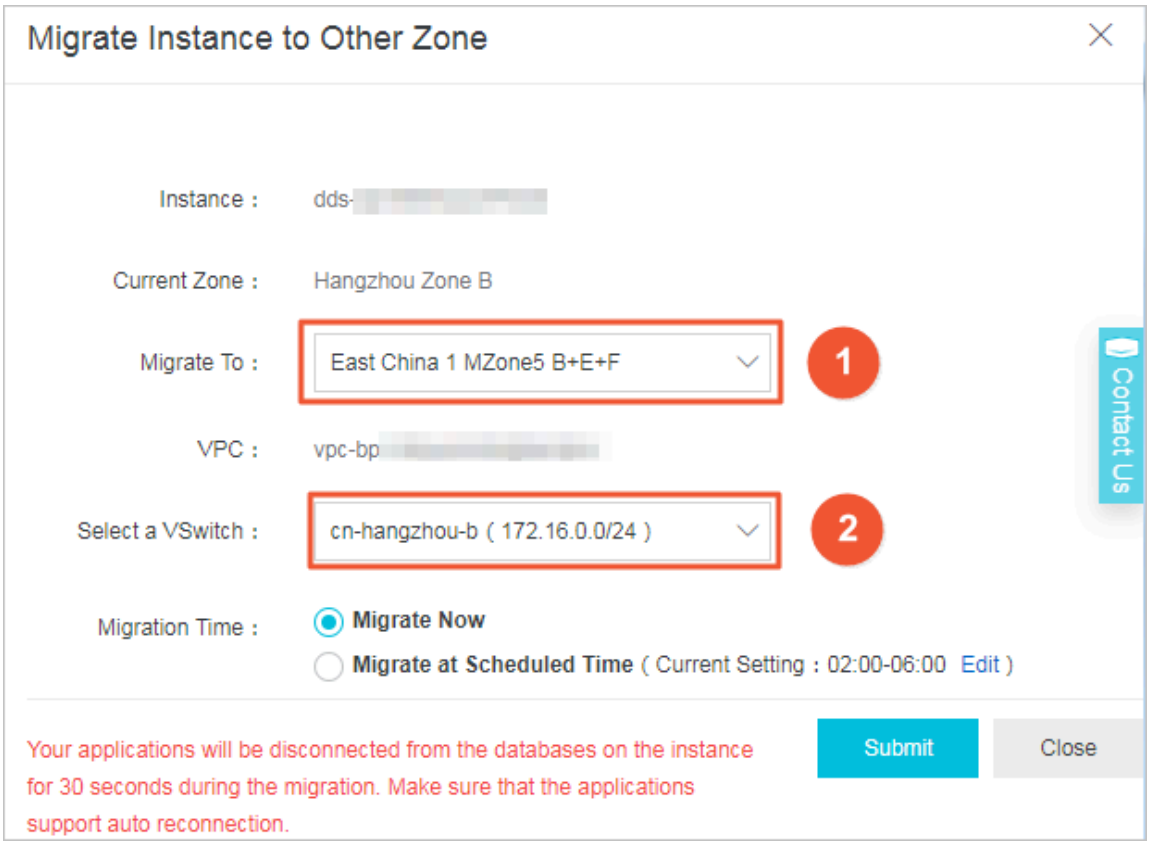
Procedure

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. Locate the instance and click the instance ID.

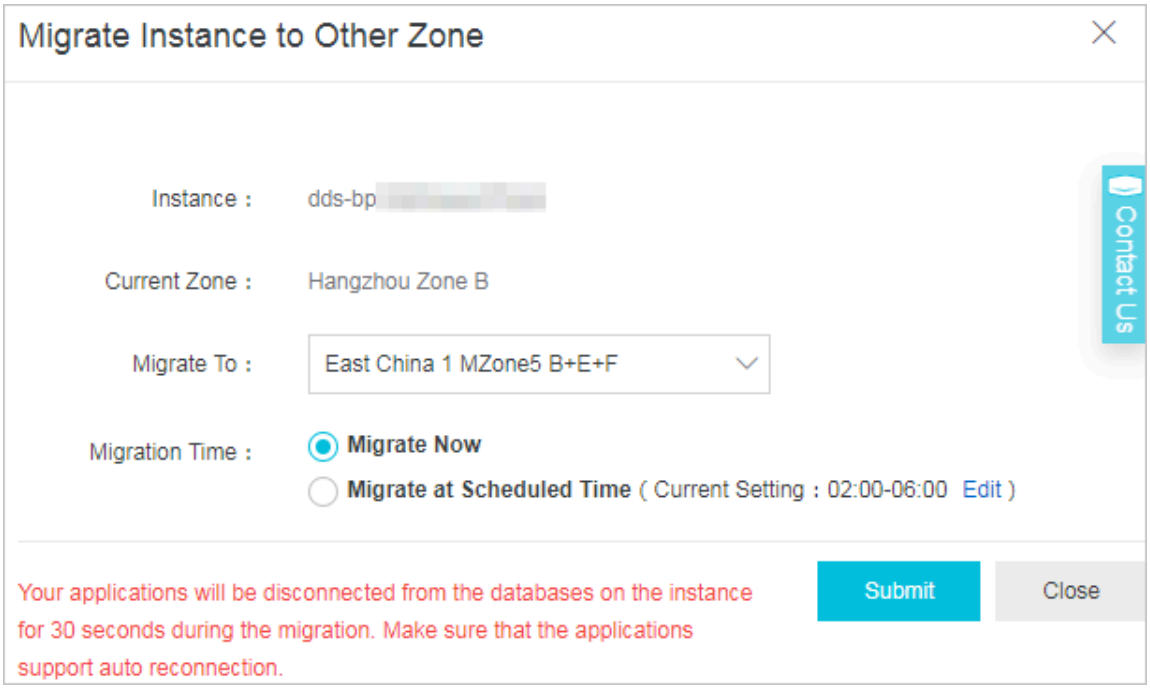
5. On the Basic Information page, locate the Basic Information section, and click Change Zone.



6. In the Migrate Instance to Other Zone dialog box that appears, configure parameters based on the network type of the instance.
   - When the network type of the instance is VPC or is a hybrid network of VPC and classic network:
     a. Select the destination zone for migration.
     b. Select the VSwitch of the destination zone.
     c. Set the time to migrate the instance.



   - When the network type of the instance is classic network:
     a. Select the destination zone for migration.
     b. Set the time to migrate the instance.



Note:

- Migrate Now: The instance is immediately migrated to another zone. The instance is migrated if the instance is in the Running state.
- Migrate at Scheduled Time: Select the period during which the instance can be migrated to another zone. You can click Edit to change the maintenance period.

  The premigration task is performed for the instance and the instance status is changed to Changing Configuration. Perform the migration task within the specified period.

7. Click OK.

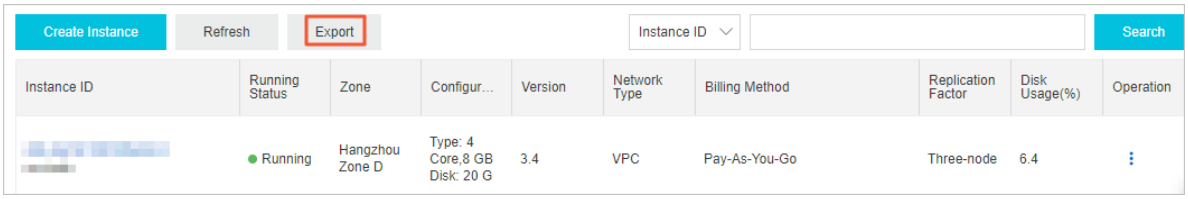
7.5 Export the list of instances

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

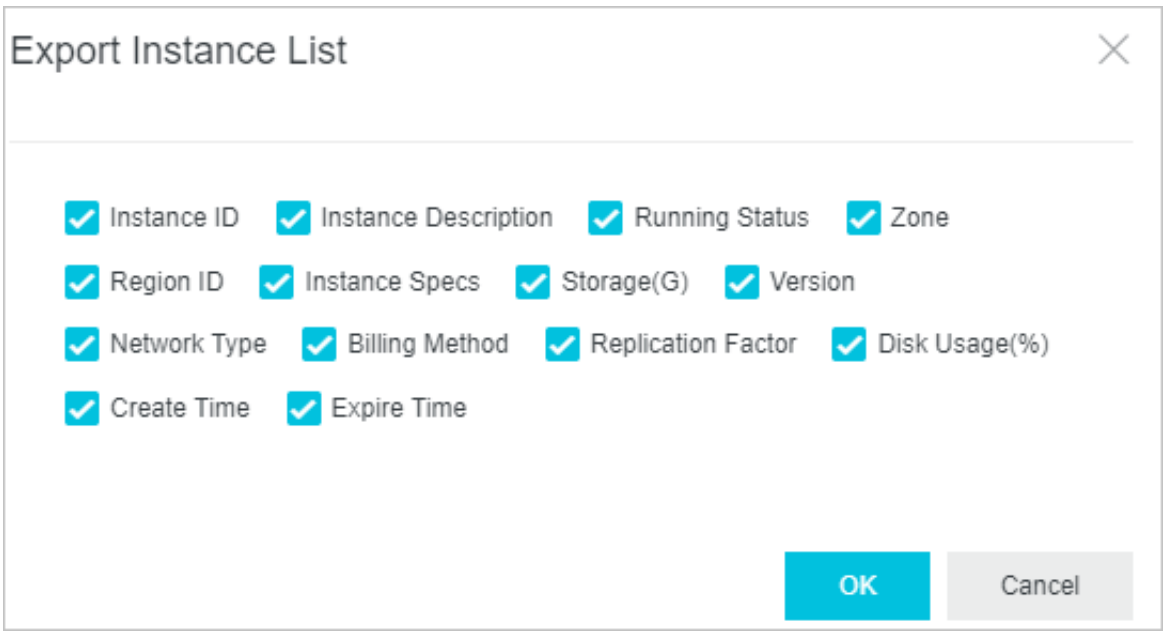
Procedure

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.



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



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.

7.6 Manage the minor database version

When ApsaraDB for MongoDB publishes a minor database version, you can log on to the ApsaraDB for MongoDB console to upgrade your ApsaraDB for MongoDB to the latest minor database version.

Before you start

During an upgrade of the minor database version, the system automatically fixes bugs in the old version. In addition, the latest minor database version also provides you with more new features. You can check the update content in View the publish logs of the latest minor database version.

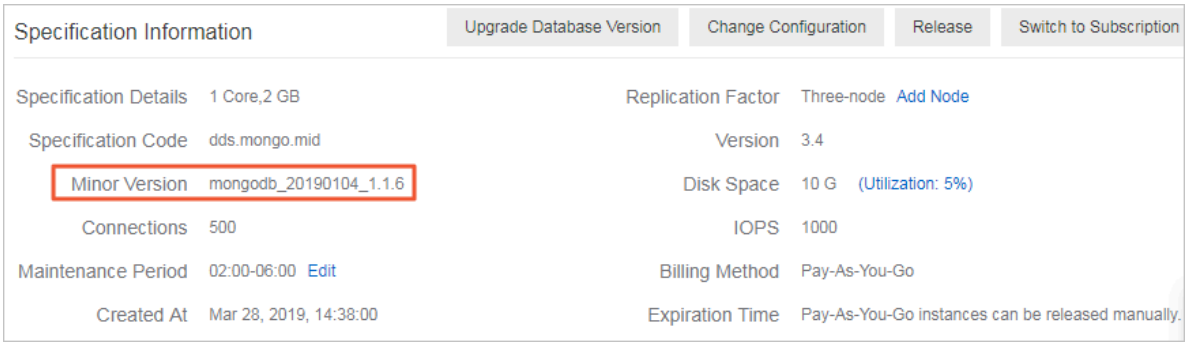
Currently, only replica set and sharded cluster instances support an upgrade of the minor database version. Standalone instances do not support this upgrade.

During the upgrade of the minor database version, instances are restarted once. The upgrade is completed when instances are being restarted. We recommend that you upgrade the minor database version for instances during off-peak hours.

> Note:
> If an instance has been upgraded to the latest minor database version, the console does not display Upgrade in the Specification Information area for the instance.

View the minor database version

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 Specification Information area, view the current minor database version of the target instance.



Upgrade the minor database version

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 Specification Information area, click Upgrade to the right of the current minor database version.
6. In the Upgrade Minor Version dialog box that appears, click OK to upgrade the current database to the latest minor database version.

View the publish logs of the latest minor database version

During the upgrade of the minor database version, you can view the update content of the latest minor database version.

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 Specification Information area, click View Publish Log to the right of the current minor database version to view the update content of the latest minor database version.

Note:
If an instance has been upgraded to the latest minor database version, the console does not display View Publish Log in the Specification Information area for the instance.

7.7 Upgrade the database version

ApsaraDB for MongoDB supports MongoDB 3.2, MongoDB 3.4, and MongoDB 4.0. You can upgrade the database version for an instance in the ApsaraDB for MongoDB console.

Database versions

For more information, see Versions and storage engines.

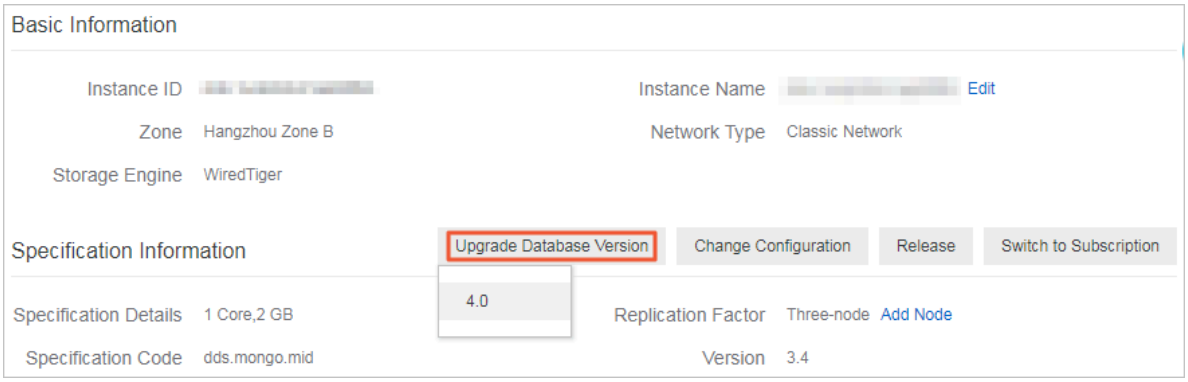
Notes

- Standalone instances support only MongoDB 3.4 and cannot be upgraded to MongoDB 4.0.

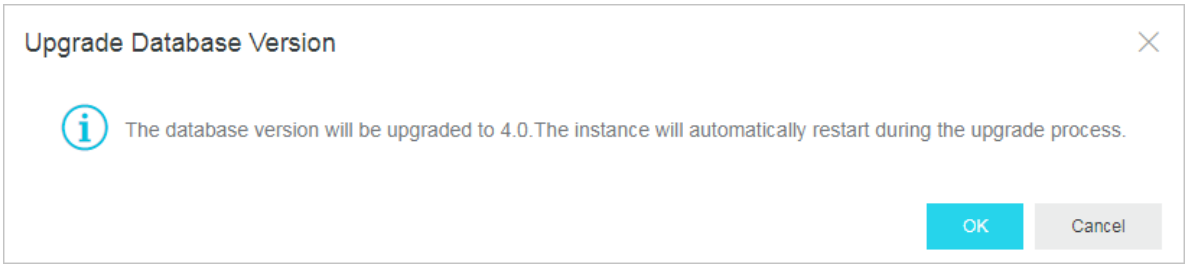
- After upgrading the database version for an instance, you cannot downgrade the upgraded version.
- An upgrade of the database version can last for some time depending on the data size of the database to be upgraded. You need to set the upgrade time in advance based on business requirements.
- Because instances are automatically restarted twice or three times in the upgrade process, you need to upgrade the database version during off-peak hours.
- If you use a connection string URI to connect your applications to an instance, the instance may be disconnected intermittently when it is being restarted. You need to ensure that your applications can automatically re-establish a connection.
- The balancer of a sharded cluster instance is disabled during an upgrade and enabled again after the upgrade.

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 Basic Information area, click Upgrade Database Version to select the target database version.



6. In the Upgrade Database Version dialog box that appears, click OK.



The instance enters the Upgrading Version status. The database version is successfully upgraded when the instance status changes to Running.

7.8 Release an instance

Based on business requirements, you can manually release a Pay-As-You-Go instance.

Prerequisites

The target instance must be a Pay-As-You-Go instance.

> Note:
> You cannot manually release a subscription-based instance, which is automatically released after it expires.

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. Choose > Release in the Operation column corresponding to the target instance.

   You can also click the target instance ID or choose > Manage in the Operation column corresponding to the target instance.

   On the Basic Information page that appears, click Release.
4. In the Release Instance dialog box that appears, click OK.

7.9 Restart an instance

If the connections to an instance exceed the upper limit or the instance encounters any performance problems, you can manually restart the instance.

For a standalone instance or three-node replica set instance, you can log on to the ApsaraDB for MongoDB console to restart the instance.

For a sharded cluster instance, you can restart the instance or restart a node of the instance. When a node is being restarted, this node cannot be accessed. During the restart of a node, you can also restart another node.

 Note:
An instance may be disconnected during a restart. You need to restart it with caution and make arrangements for business interruption.

Restart a standalone or replica set instance

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. Choose > Restart in the Operation column corresponding to the target instance.

   You can also click the target instance ID or choose > Manage in the Operation column corresponding to the target instance. On the Basic Information page that appears, click Restart Instance.
4. In the Restart Instance dialog box that appears, click OK.

   The instance immediately enters the Rebooting status. The instance is successfully restarted when its status changes to Running.

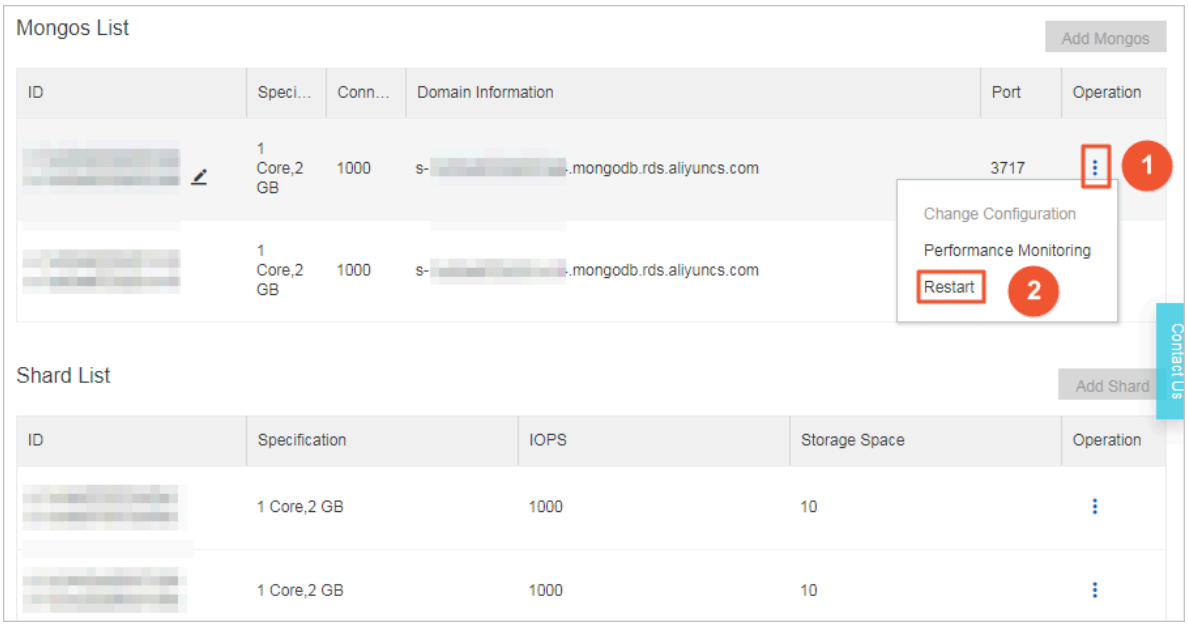
Restart a sharded cluster instance

Restart a sharded cluster instance: The procedure for restarting a sharded cluster instance is the same as that for restarting a standalone or replica set instance. For more information, see Restart a standalone or replica set instance.

Restart a node of a sharded cluster instance

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. Click the target instance ID or choose > Manage in the Operation column corresponding to the target instance.
4. To restart a mongos node, do as follows: In the Mongos List area of the Basic Information page, choose > Restart in the Operation column corresponding to the target mongos node.



Note:
When a node is being restarted, this node cannot be accessed.

5. To restart a shard, do as follows: In the Shard List area of the Basic Information page, choose > Restart in the Operation column corresponding to the target shard.
6. In the Restart Node dialog box that appears, click OK.

   The instance immediately enters the Rebooting status. The node is successfully restarted when the instance status changes to Running.

8 Network connection management

8.1 Modify the connection information

You can log on to the ApsaraDB for MongoDB console to modify the intranet or Internet connection information for an instance.

Context

For a standalone instance, you can modify the intranet or public address for the primary node only.

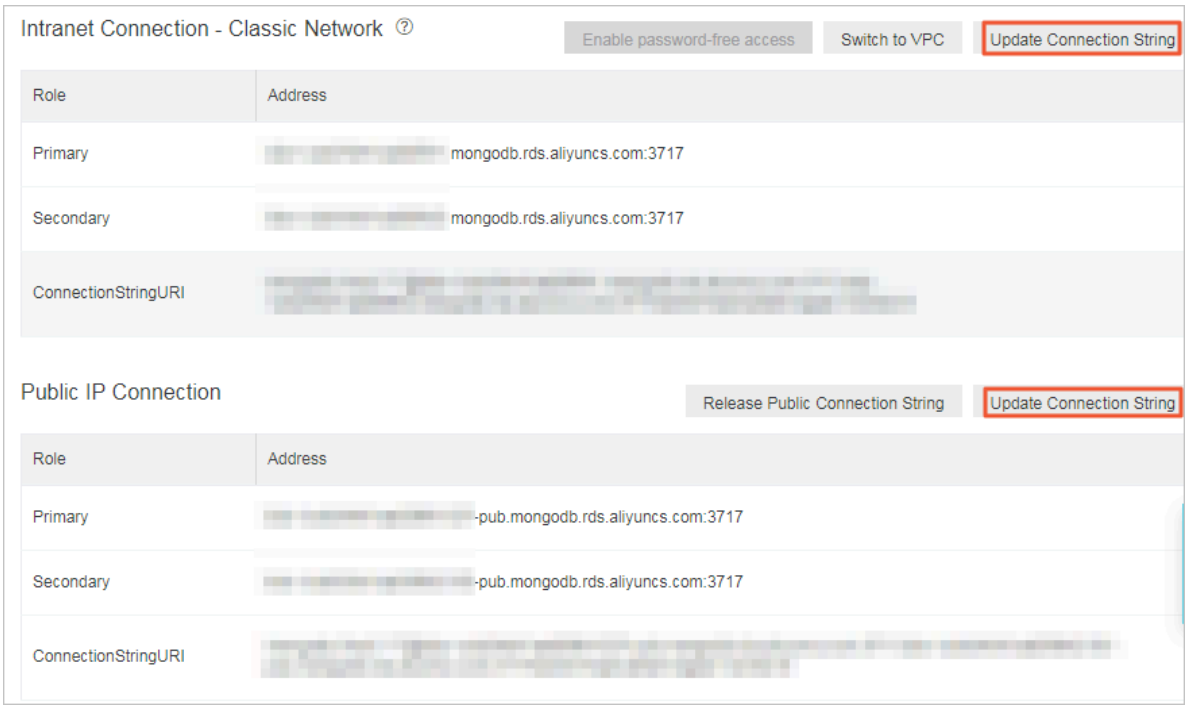
For a replica set instance, you can modify the intranet or public address for the primary and secondary nodes.

For a sharded cluster instance, you can modify the intranet or public address for all mongos nodes.

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 Database Connection.

6. In the Intranet Connection or Public IP Connection area, click Update Connection String, as shown in the following figure.



7. In the Update Connection String dialog box that appears, modify the connection information for the instance.

   You can modify the intranet or public address of the instance. For more information, seeTable 8-1: Parameters of the connection information

Table 8-1: Parameters of the connection information

Instance	Network type	Parameter setting	Description
Standalone	Intranet or Internet	Modify the intranet or public address of the primary node.	You can modify only the prefix of the address. A connection address starts with a lowercase letter and consists of letters and digits. It contains 8–64 characters.
Replica set		Select the primary node or a secondary node and modify its intranet or public address.	
Sharded cluster		Select a mongos node and modify its intranet or public address.	

8. After setting the required parameter, click OK.

What's next

After modifying the intranet or Internet connection information, you need to use the modified address to connect a terminal or application to the instance.

8.2 Switch the network type of an instance

ApsaraDB for MongoDB allows you to create an instance whose network type is classic network or VPC. You can log on to the ApsaraDB for MongoDB console or call the ModifyDBInstanceNetworkType operation to switch between the two network types.

Network types

- On a classic network, instances are not isolated. You can configure a whitelist policy for them to block unauthorized access.
- A VPC is an isolated network environment that is securer and recommended.

  You can customize the routing table, IP address range, and gateway in the VPC. 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.

Notes

You can switch the network type for replica set and sharded cluster instances, but not for standalone instances.

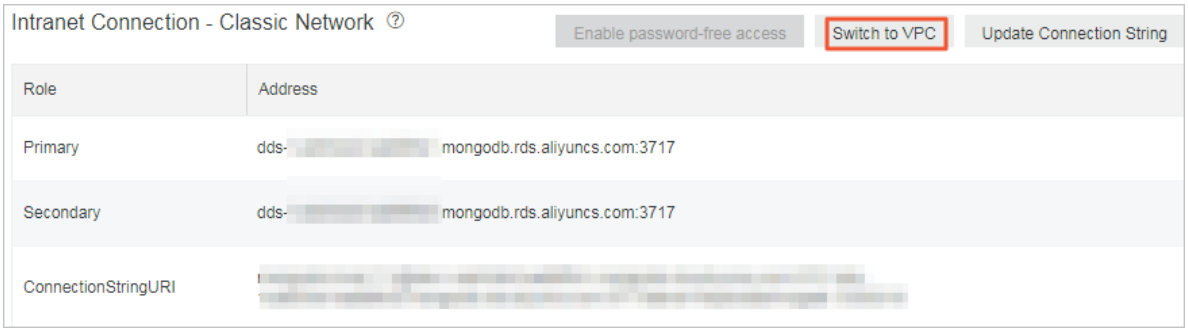
During the switchover, the target instance may be disconnected transiently once. We recommend that you switch the network type during off-peak hours or ensure that your applications can automatically re-establish a connection to avoid an impact of intermittent disconnection.

Switch from a classic network to a VPC

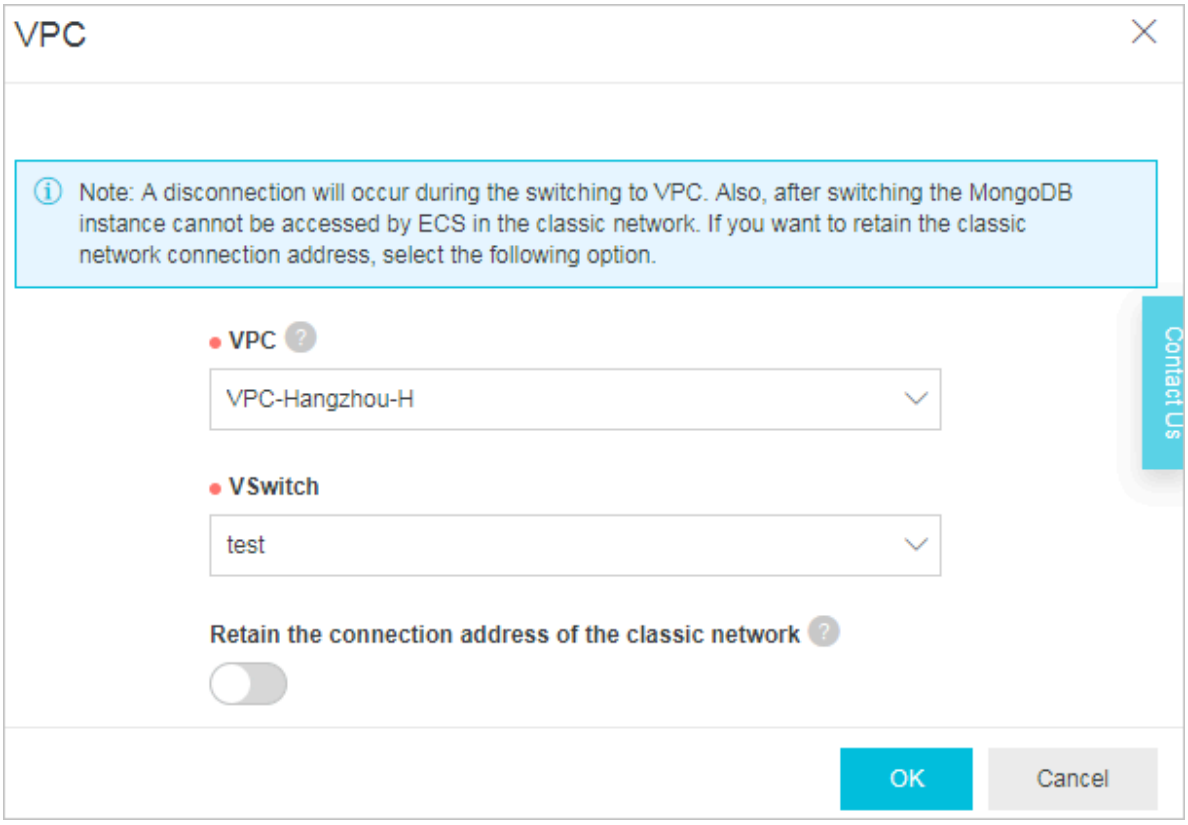
You can choose to keep the intranet addresses on the classic network to smoothly switch the network type without intermittent disconnection. For more information, see #unique_46.

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 home page, select the region where the target instance is located.
4. In the left-side navigation pane, click Replica Set Instances or Sharding Instances based on the architecture of the target instance.
5. Locate the target instance and click its instance ID.
6. In the left-side navigation pane, click Database Connection.
7. In the Intranet Connection - Classic area, click Switch to VPC.



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



Note:

- You can enable Retain the connection address of the classic network to generate new intranet addresses in the VPC and keep the existing intranet

addresses on the classic network within a specified period. When the period expires, the intranet addresses on the classic network are automatically released.

- If you do not enable Retain the connection address of the classic network, the target ApsaraDB for MongoDB instance may be disconnected transiently once when its network type is switched to VPC. Cloud products, such as ECS, on the classic network cannot be connected to this instance. We recommend that you switch the network type during off-peak hours or ensure that your applications can automatically re-establish a connection to avoid an impact of intermittent disconnection.

9. Click OK.

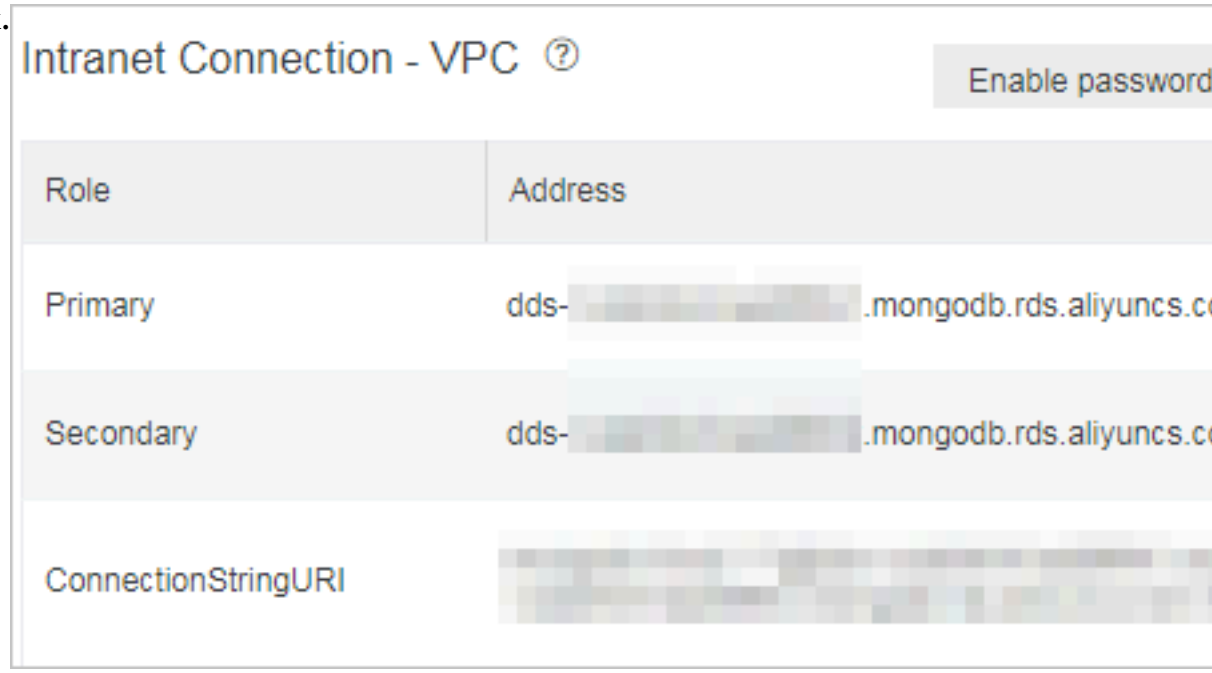
Switch from a VPC to a classic network

After the network type of an ApsaraDB for MongoDB instance is switched to classic network, intranet addresses in the VPC are released and ECS instances in the VPC cannot access this instance through the intranet. ApsaraDB for MongoDB generates intranet addresses on the classic network and remains public addresses unchanged. You need to modify the connection information in your applications.

 Note:
After the network type of an ApsaraDB for MongoDB instance is switched to classic network, ECS instances in the VPC cannot be connected to this instance. During the switchover, the target instance may be disconnected transiently once. We recommend that you switch the network type during off-peak hours or ensure that your applications can automatically re-establish a connection to avoid an impact of intermittent disconnection.

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 Database Connection.

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



7. In the Switch to Classic Network dialog box that appears, click OK.

8.3 Configure a VPC for a new instance

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

Context

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

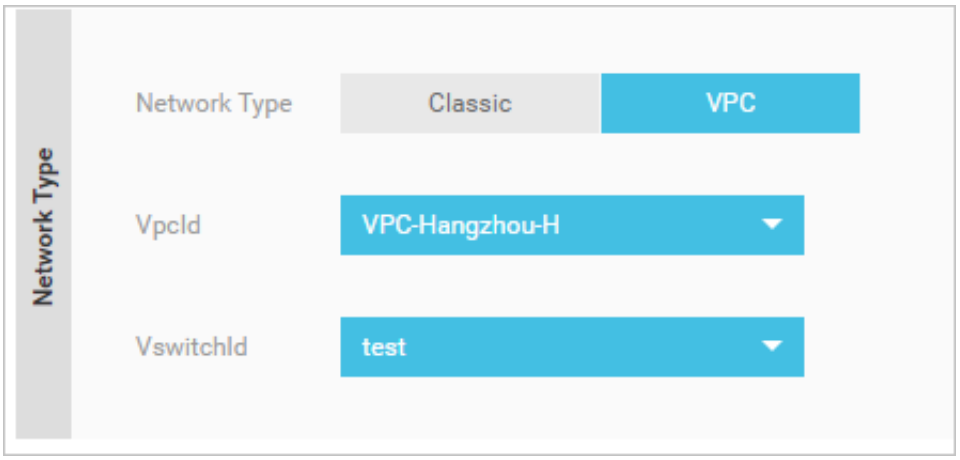
- On the classic network, cloud services are not isolated. You can configure a security group or whitelist policy for them to block unauthorized access.
- A VPC helps you build an isolated network environment in Alibaba Cloud, where you can customize its routing table, IP address range, and gateway.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

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

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

Procedure

1. Create a VPC. For more information, see Create a VPC.
2. Create an ApsaraDB for MongoDB instance in the same region as the VPC.
3. When creating the ApsaraDB for MongoDB instance, select VPC as the network type on the instance creation page.
4. Under VPC, select the configured VPC and VSwitch for VPC and VSwitch, respectively, as shown in the following figure.



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

8.4 Configure a hybrid access solution to smoothly switch from a classic network to a VPC

To meet the increasing network switchover requirements, ApsaraDB for MongoDB provides a hybrid network access feature to help you smoothly switch from a classic network to a VPC without intermittent disconnection or network disconnection.

Prerequisites

The target instance must be a replica set or sharded cluster instance.

Constraints

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

Solution

When you switch the network type of an ApsaraDB for MongoDB instance from classic network to VPC, ApsaraDB for MongoDB immediately releases the intranet addresses on the classic network. In this case, the instance is disconnected for 30s once and cloud products (such as ECS) on the classic network cannot be connected to this instance.

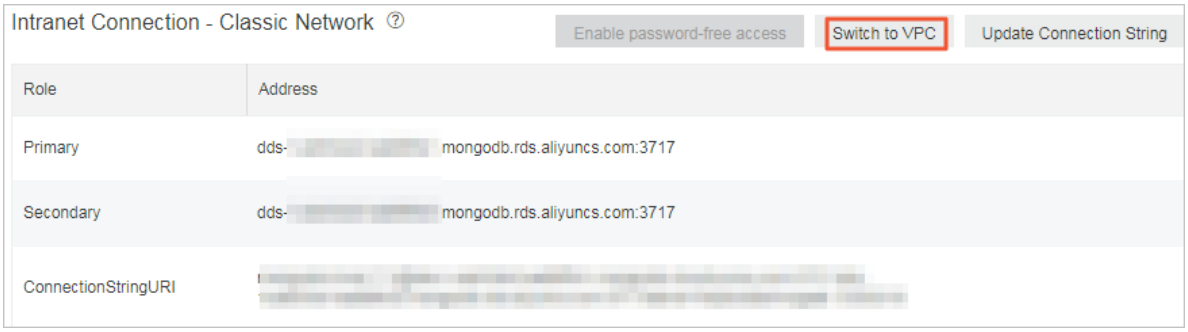
Using the hybrid access solution, you can connect ECS instances on the classic network and in the VPC to the ApsaraDB for MongoDB instance at the same time to smoothly switch its network type. When you switch its network type from classic network to VPC, you can enable ApsaraDB for MongoDB to generate new intranet addresses in the VPC and keep the existing intranet addresses on the classic network within a specified period, a maximum of which is 120 days. Within the specified period, this ApsaraDB for MongoDB instance can be accessed by ECS instances on the classic network and in the VPC.

In hybrid access mode, you can gradually switch the network type or migrate ECS and other cloud products from the classic network to the VPC until all products can be interconnected through the securer VPC on the intranet.

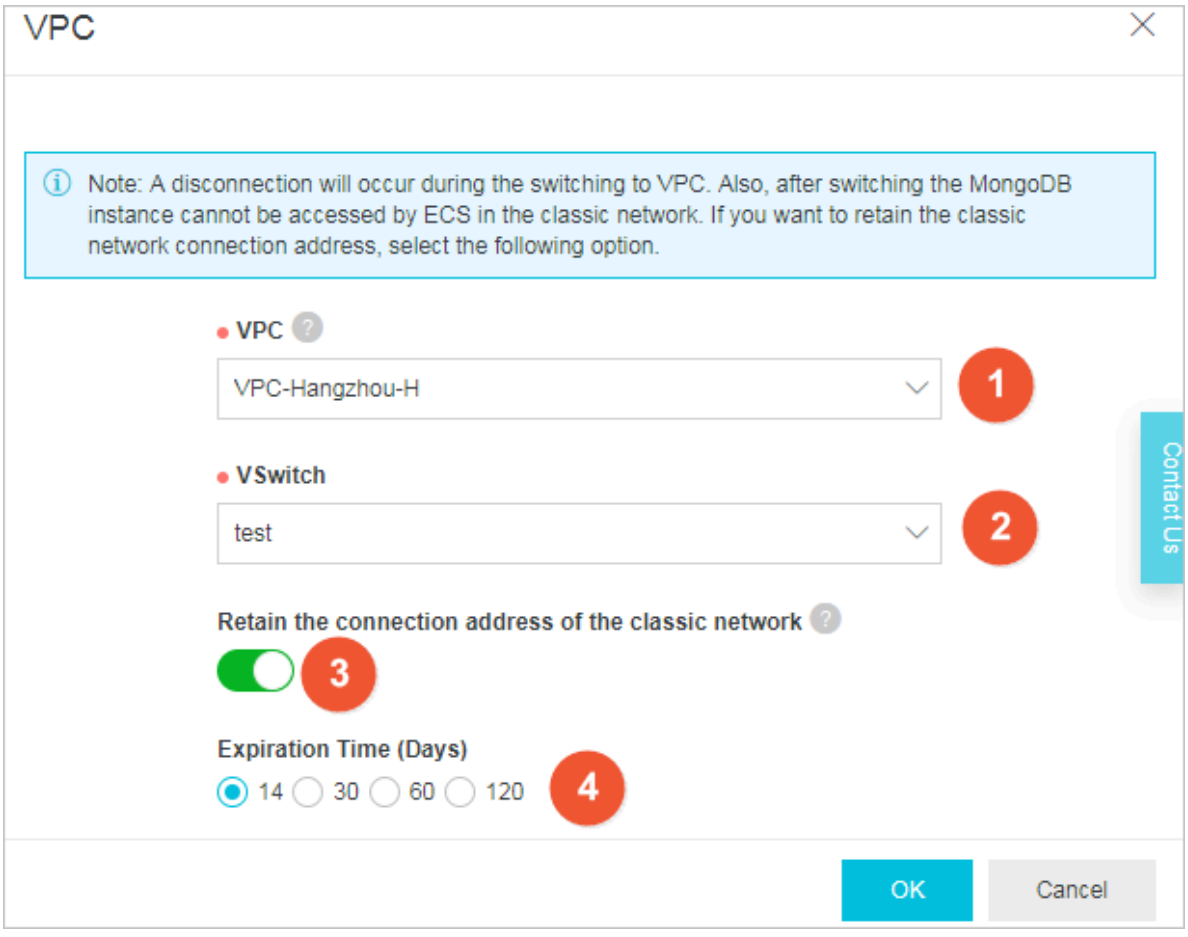
Procedure

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 Database Connection.
6. In the Intranet Connection - Classic area, click Switch to VPC.



7. In the VPC dialog box that appears, set related parameters.



a. Specify VPC and VSwitch.

> **Note:**

For more information about how to create a VPC or VSwitch, see Create a VPC and VSwitch.

b. Turn on the Retain the connection address of the classic network switch.

c. Select a period for Expiration Time (Days).

8. Click OK.

8.5 Apply for a public address

ApsaraDB for MongoDB allows you to apply for a public address to connect to an instance through the Internet.

Apply for a public address for a standalone instance

Apply for a public address for a replica set instance

Apply for a public address for a sharded cluster instance

8.6 Release a public address

After using a public address to connect to an ApsaraDB for MongoDB instance through the Internet, you can log on to the ApsaraDB for MongoDB console or call the ReleasePublicNetworkAddress operation to release this public address.

Notes

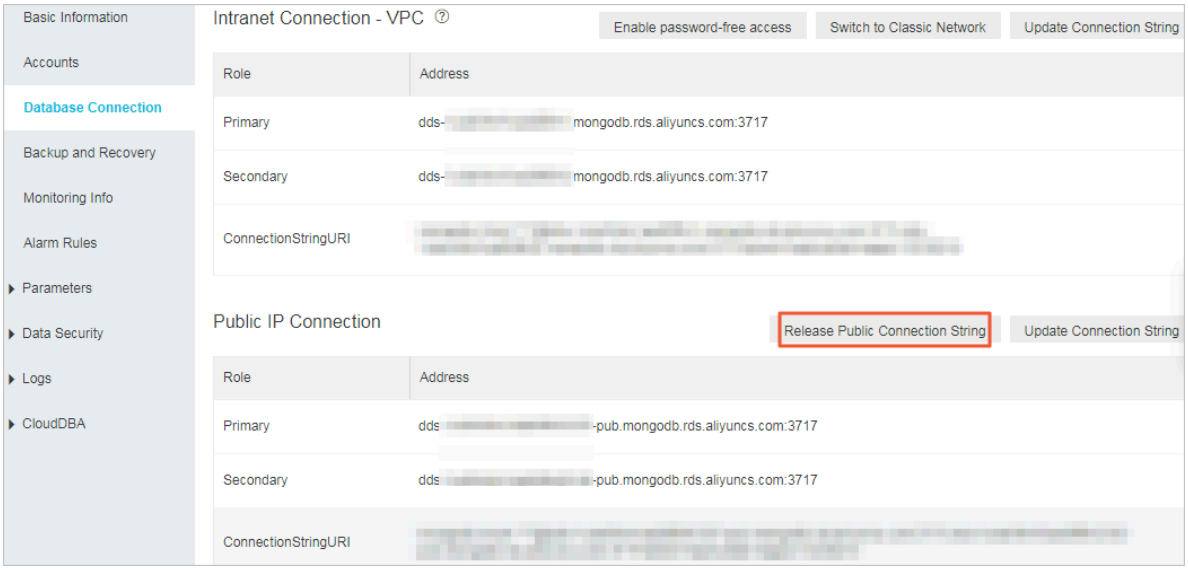
- For a sharded cluster instance, you can release the public address for one or more mongos nodes. You can still use a public address that is not released to connect to the corresponding mongos node.
- After the public address of an instance or mongos node is released, you cannot use this public address to connect to the instance or mongos node.
- After the public address of an instance is released, if you no longer use a public IP address to connect to this instance, we recommend that you delete this public IP address from the whitelist to guarantee data security. For more information, see Configure a whitelist.

Release the public address for a standalone or replica set instance

To release the public address for a replica set instance, you release the public addresses of the primary and secondary nodes.

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 Database Connection.
6. In the Public IP Connection area, click Release Public IP Address.



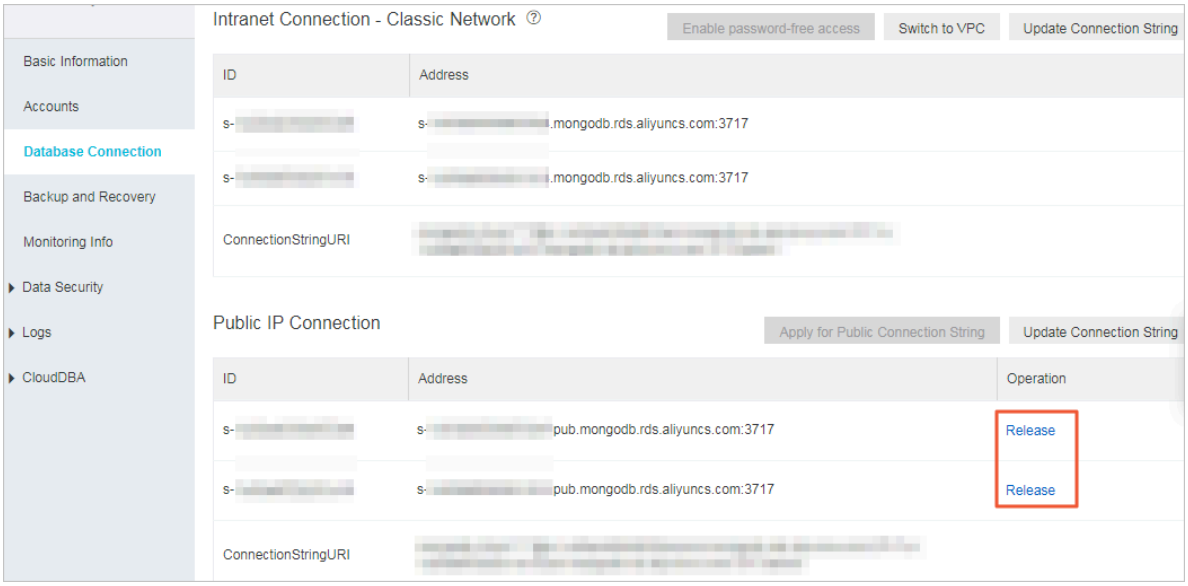
7. In the Release Public IP Address dialog box that appears, click OK.

Release the public address for a sharded cluster instance

You can release the public address for one or more mongos nodes of a sharded cluster instance. Then, you can still use a public address that is not released to connect to the corresponding mongos node.

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 left-side navigation pane, click Database Connection.
6. In the Public IP Connection area, locate the target mongos node whose public address needs to be released.

7. In the Operation column corresponding to the target mongos node, click Release.



> **Note:**
> You can repeat this step to release the public addresses for other mongos nodes based on business requirements. To release the public address for another mongos node of this instance, you need to wait until the last public address is released.

8. In the Release Public IP Address dialog box that appears, click OK.

8.7 Modify the expiration time for the classic network

In hybrid network access mode, you can modify the expiration time for the classic network.

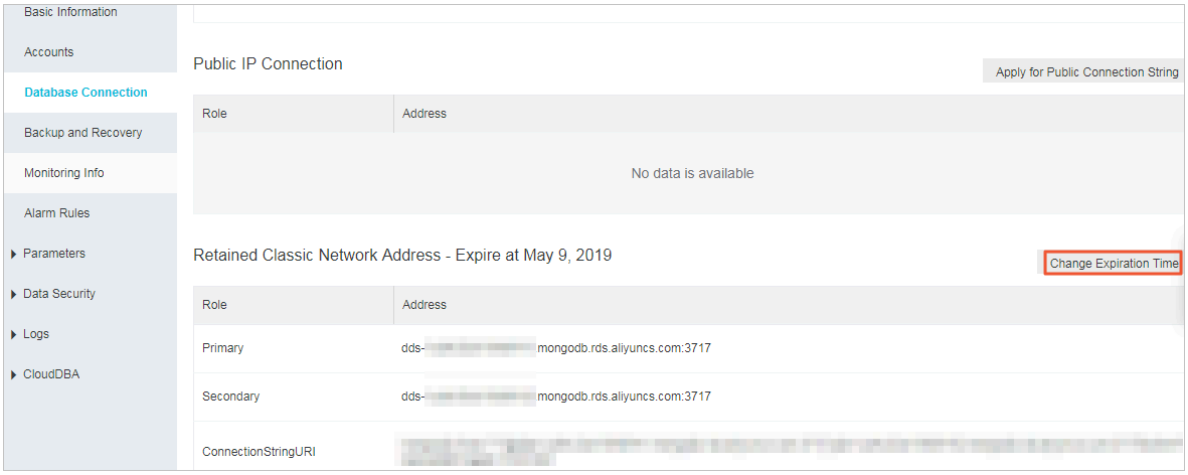
Context

When you switch the network type of an instance from classic network to VPC, you can choose to keep the intranet addresses on the classic network within a specified period. When generating new intranet addresses in the VPC, ApsaraDB for MongoDB can keep the existing intranet addresses on the classic network within the specified period. In this case, you can use the hybrid network access solution to smoothly switch from the classic network to the VPC without intermittent disconnection. When the period expires, the intranet addresses on the classic network are automatically released.

ApsaraDB for MongoDB allows you to modify the expiration time for the classic network within the previously specified period to shorten or prolong the period for keeping the intranet addresses on the classic network.

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 Database Connection.
6. In the Retained Classic Network Address area, click Change Expiration Time.



7. In the Change Expiration Time dialog box that appears, select a period for Expiration Time (Days).

> Note:
> You can set the expiration time to 14 days, 30 days, 60 days, or 120 days for the classic network.

8. Click OK.

9 Data security

9.1 Configure a whitelist

After creating an ApsaraDB for MongoDB instance, you need to configure a whitelist for the instance to allow external devices to access this instance. The default whitelist contains only the IP address 127.0.0.1, indicating that no device is allowed to access this instance. This topic describes how to configure a whitelist in the console.

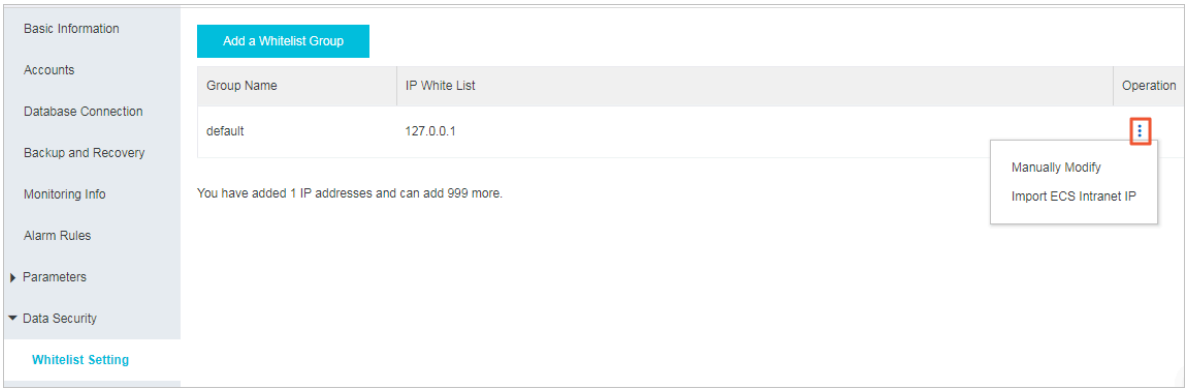
Notes

- Before using the target instance for the first time, you must configure its whitelist. After configuring the whitelist, you can view the connection information about the instance on its Basic Information page and Database Connection page.
- If you use the whitelist correctly, you can guarantee the highest-level security protection for your ApsaraDB for MongoDB instance. We recommend that you maintain the whitelist on a regular basis.

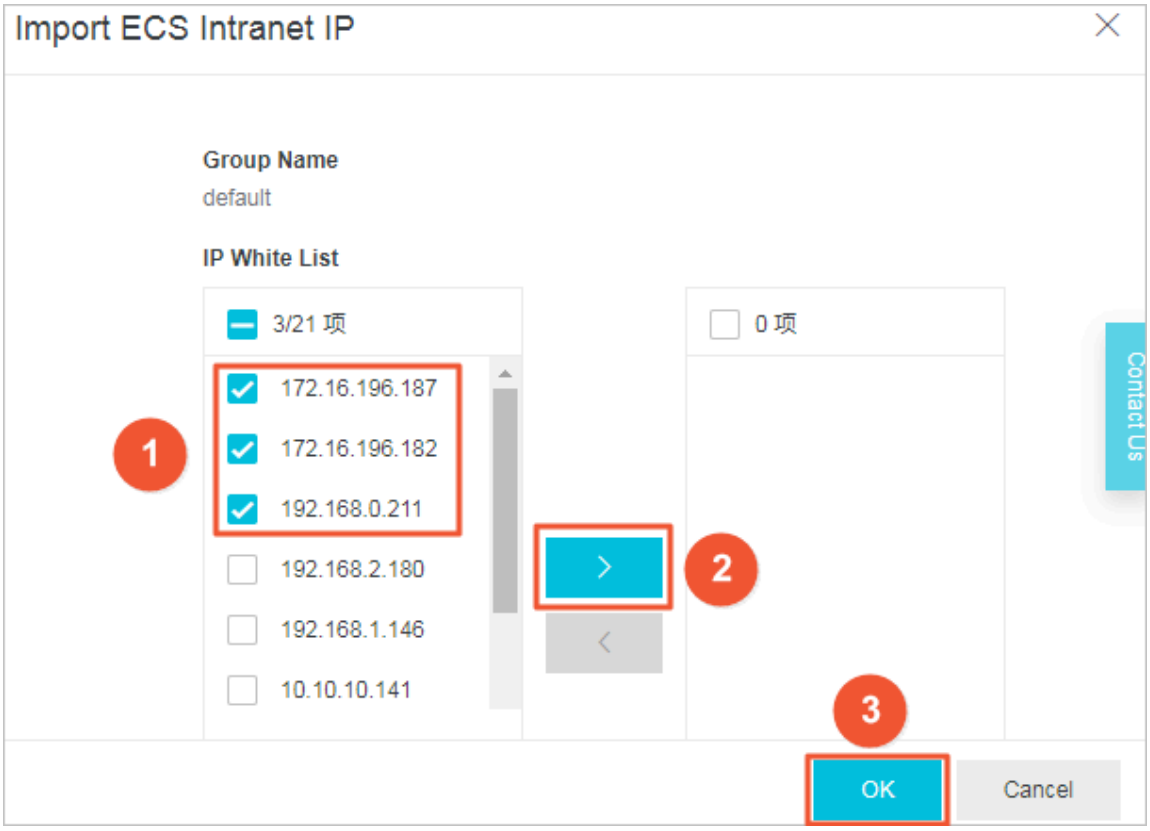
Procedure

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, choose Data Security > Whitelist Setting.

6. Select Manually Modify or Import ECS Intranet IP to configure the IP address whitelist.



- Select Manually Modify. On the page that appears, enter IP addresses or CIDR blocks and click OK.
- Select Import ECS Intranet IP. The system displays all ECS intranet IP addresses under your account. You can select ECS intranet IP addresses, add them to the whitelist, and click OK.



Note:

- You need to separate IP addresses with commas (,) and ensure that they are different from one another. You can add a maximum of 1,000 IP addresses.

Supported formats include 0.0.0.0/0, 10.23.12.24, and 10.23.12.24/24. 10.23.12.24 is an IP address, and 10.23.12.24/24 is a CIDR notation, in which the suffix /24 indicates the number of bits for the prefix of the IP address. The suffix ranges from 1 to 32.

- 0.0.0.0/0 and empty indicate that your ApsaraDB for MongoDB instance can be accessed by all IP addresses. In this case, the database is at high security risk. We recommend that you add only the public IP addresses or CIDR blocks of your web servers to the whitelist.

Delete a whitelist group

You can delete whitelist groups other than the default group.

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, choose Data Security > Whitelist Setting.
6. Locate the whitelist group to be deleted. Choose > Delete Whitelist Group in the Operation column.
7. In the Delete Whitelist Group dialog box that appears, click OK to delete the whitelist group.

9.2 Configure log auditing

ApsaraDB for MongoDB provides log auditing to record all database operations that you have performed. Based on log auditing, you can conduct fault analysis, behavior analysis, and security audits for ApsaraDB for MongoDB. This feature can effectively help you obtain the information about data operations.

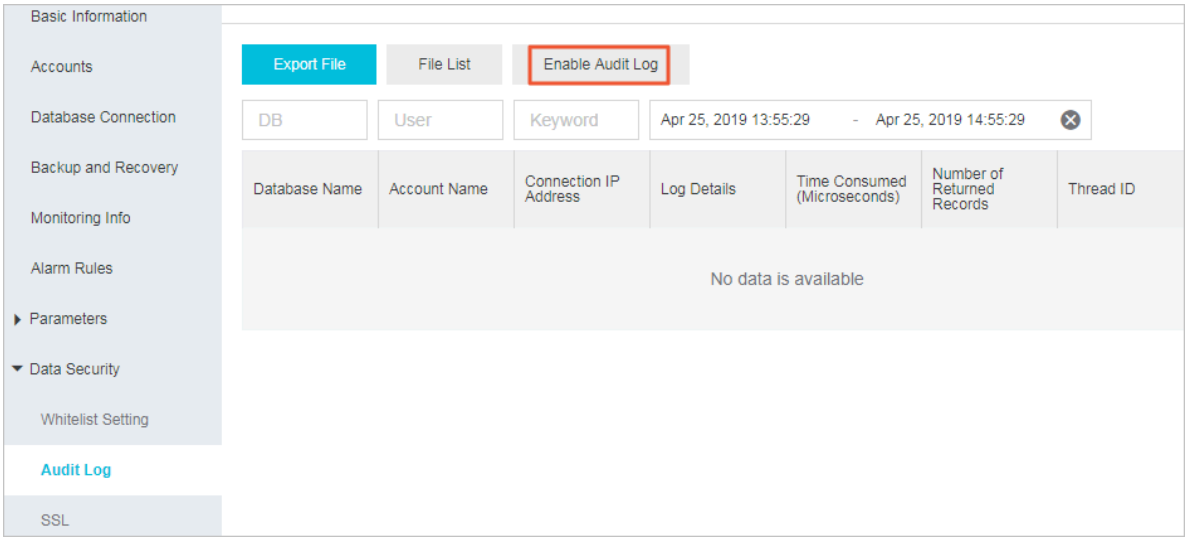
Before you start

- Replica set and sharded cluster instances support log auditing. Standalone instances do not support this feature.
- You can specify the types of database operations to be audited for replica set instances.

- You cannot specify the types of database operations to be audited for sharded cluster instances. When log auditing is enabled, the system automatically audits admin, slow, query, insert, update, and delete operations.
- After log auditing is enabled, ApsaraDB for MongoDB stores the audit data for 30 days by default.
- You can enable and disable log auditing only in the console. For more information, see Enable log auditing and Disable log auditing.
- To query audited logs, you can log on to the ApsaraDB for MongoDB console or call the DescribeAuditRecords operation.

Enable log auditing

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, choose Data Security > Audit Log.
6. Click Enable Audit Log.



> **Note:**
> When you enable log auditing, the CloudDBA index optimization feature is also enabled. For more information about CloudDBA index optimization, see#unique_60.

7. Click OK.

Query and download audited logs

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, choose Data Security > Audit Log.
6. You can query, export, and download audited logs.

   - Query: You can enter the database name (DB), username used to log on to the database (User), and a word or record in a collection (Keyword), and select or enter the start time and end time to query audited logs by condition.

     Table 9-1: Parameters of audited logs

Name	Description
Database Name	The name of the queried database. If you specify the database name, audited logs of the specified database are displayed for the target instance. If you do not specify the database name, audited logs of all databases are displayed for the target instance.
Account Name	The username used to log on to the queried database. If you specify the username, audited logs of the database that is logged on to by using the specified username are displayed for the target instance. If you do not specify the username, audited logs of all databases are displayed for the target instance.
Connection IP Address	The IP address of the client used to log on to the queried database. If you specify the client IP address, audited logs of the database that is logged on to on the specified client are displayed for the target instance. If you do not specify the client IP address, audited logs of all databases are displayed for the target instance.

Name	Description
Log Details	The statement that was run and recorded in the audited logs. If you specify the keyword, audited logs that contain the specified keyword are displayed for the target instance. If you do not specify the keyword, audited logs of all databases are displayed for the target instance.
Time Consumed (Microseconds)	The execution time of the statement.
Number of Returned Records	The number of records returned after the statement was run.
Thread ID	None
Execution Time	The time when the statement was run.

- Export File: You can export a file of audited logs.

  Note:
  If the number of statements in audited logs that meet the filtering conditions exceeds 1 million, only 1 million statements can be exported. Statements are exported at the speed of 900 rows per second. It takes about 20 minutes to export 1 million statements.

- File List: You can view a list of exported files of audited logs. Table 9-2: Parameters of exported files of audited logs describes the parameters of the list.

Table 9-2: Parameters of exported files of audited logs

Name	Description
File ID	The ID automatically generated by the system for the file of audited logs.

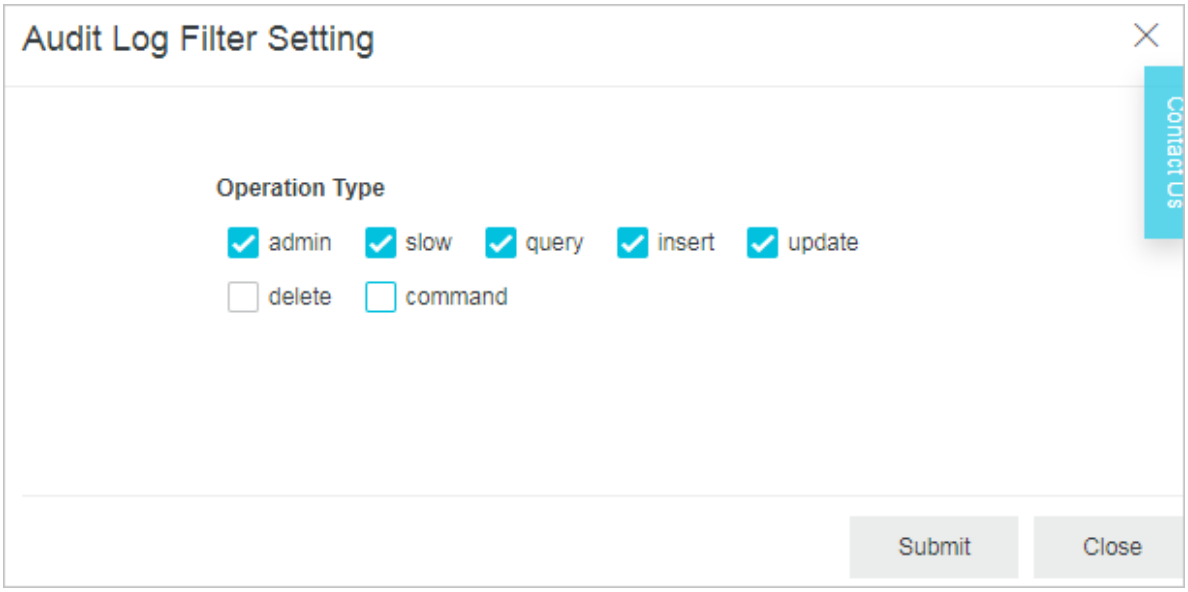
Name	Description
Archiving Status	The archiving status of the file of audited logs, including: - Initializing: indicates that the system has not started to export or is exporting the file of audited logs. - Success: indicates that the system has successfully exported the file of audited logs. Note: You can download files in the Success status only.
Audit Start Time	The start time for exporting the file of audited logs.
Audit End Time	The end time for exporting the file of audited logs.
Download	The button that you click to download the file of audited logs to a local device.
Log Size	The size of the file of audited logs.

Specify audit settings

After log auditing is enabled for a replica set instance, you can specify the types of database operations to be audited.

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 Data Security > Audit Log.
6. Click Audit Log Filter Setting.

7. In the Audit Log Filter Setting dialog box that appears, select the types of database operations to be audited.



You can select the following database operations:

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

Note:

- If log auditing was enabled for instances before July 2018, the default types of database operations to be audited are admin, slow, insert, update, delete, and command. If you need to audit the query operation, you can select query in audit settings.
- If log auditing is enabled for instances after July 2018, the default types of database operations to be audited are admin, slow, query, insert, update, delete, and command.

8. Click OK.

Disable log auditing

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, choose Data Security > Audit Log.
6. Click Disable Audit Log.

   Note:

   - After you disable log auditing, the CloudDBA index optimization feature is also disabled.
   - After you disable log auditing, logs are no longer collected, subsequent database operations cannot be audited, and stored audited logs are also deleted .
7. In the Disable Audit Log dialog box that appears, click OK.

9.3 Configure SSL encryption

To enhance the security of data links, you can enable SSL encryption and install an SSL certificate issued by the CA in your application. The SSL encryption feature encrypts network connections at the transport layer to improve data security and guarantee data integrity during communication. This topic describes how to view the details of SSL encryption, enable and disable SSL encryption, and update and download an SSL CA certificate.

Notes

- Only replica set instances whose database version is MongoDB 3.4 or MongoDB 4.0 support SSL encryption.
- When you enable, update, or disable SSL encryption for an instance, the instance is restarted once. Therefore, we recommend that you perform such an operation during off-peak hours.
- You can download an SSL CA certificate only from the console.
- Due to the inherent defects of SSL encryption, this feature significantly increases the CPU usage. We recommend that you enable SSL encryption only when external

network links need to be encrypted. Intranet links are securer and generally do not need to be encrypted.

Enable SSL encryption

 Note:
When you enable SSL encryption for an instance, the instance is restarted once. Therefore, we recommend that you perform this operation during off-peak hours.

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 Data Security > SSL.
6. In the SSL area, turn on the SSL Status switch.
7. In the Restart Instance dialog box that appears, click OK.

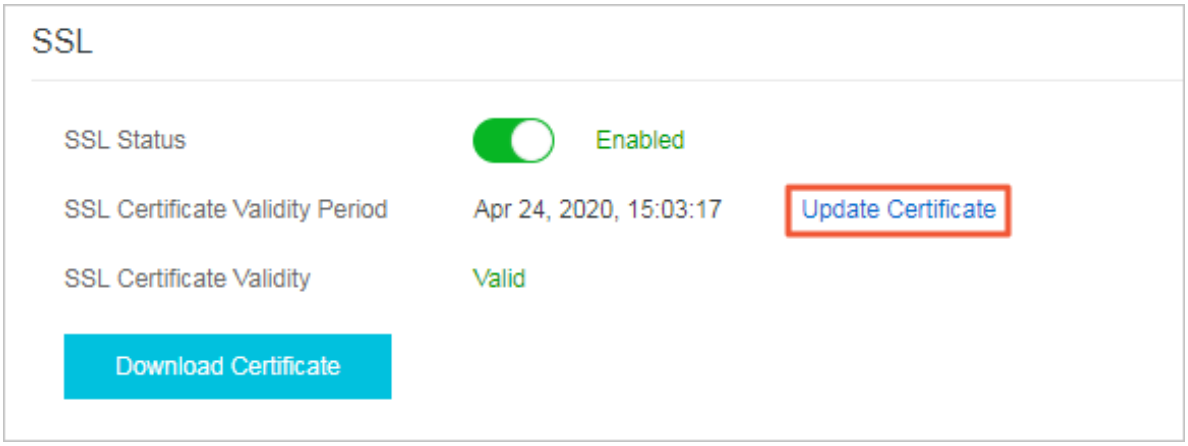
Update an SSL CA certificate

An SSL CA certificate is valid for a year. You can update an SSL CA certificate when it expires or within its validity period.

 Note:
When you update the SSL CA certificate for an instance, the instance is restarted once. Therefore, we recommend that you perform this operation during off-peak hours.

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 Data Security > SSL.

6. In the SSL area, click Update Certificate.



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

Download an SSL CA certificate

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 Data Security > SSL.
6. Click Download Certificate to download the SSL CA certificate to a local device.

Disable SSL encryption

You can disable SSL encryption when you do not need this feature.

> Note:
> When you disable SSL encryption for an instance, the instance is restarted once. Therefore, we recommend that you perform this operation during off-peak hours.

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 Data Security > SSL.
6. In the SSL area, turn off the SSL Status switch.
7. In the Restart Instance dialog box that appears, click OK.

9.4 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 #unique_63.

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.

```
 var   MongoClien  t  =  require (' mongodb '). MongoClien  t ,
   f  =  require (' util '). format ,
   fs  =  require (' fs ');

//  Read   the   CA   certificat  e .
 var   ca  = [ fs . readFileSy  nc ( __dirname  + "/ path / to / ca .
 pem ")];

//  Connect   to   the   MongoClien  t   and   validate   the
 certificat  e   returned   from   the   server .
 MongoClien  t . connect (" mongodb :// host01 : 27017 , host02 :
 27017 , host03 : 27017 /?  replicaSet = myreplset & ssl = true ", {
   server : {
       sslValidat  e : true ,
       checkServe  rIdentity : false , #  Ignore   host   name
 verificati  on .
       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.

```
function __construc t ($ uri = ' mongodb :// 127 . 0 . 0 . 1 /',
array $ uriOptions = [], array $ driverOpti ons = [])
```

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
$ client = new MongoDB \ Client (
    ' mongodb :// host01 : 27017 , host02 : 27017 , host03 : 27017 ',
    [ ' ssl ' => true ,
        ' replicaSet ' => ' myReplicaS et '
    ],
    [
        " ca_file " => "/ path / to / ca . pem ",
        " allow_inva lid_hostna me " => 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.

```
import com . mongodb . MongoClien tURI ;
import com . mongodb . MongoClien tOptions ;
MongoClien tOptions options
= MongoClien tOptions . builder (). sslEnabled ( true ). sslInvalid
HostNameAl lowed ( true ). build ();
MongoClien t client = new MongoClien t (" mongodb :// host01
: 27017 , host02 : 27017 , host03 : 27017 /? replicaSet = myreplset
", options );
```

Run a keytool command to specify the CA certificate.

```
keytool - importcert - trustcacer ts - file < path to
certificat e 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.

```
System . setPropert y (" javax . net . ssl . trustStore ","/ trust /
mongoStore . ts ");
```

```
System . setPropert  y (" javax . net . ssl . trustStore  Password
"," 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.

```
import   ssl
from   pymongo   import   MongoClien  t

uri  = " mongodb :// host01 : 27017 , host02 : 27017 , host03 : 27017
/?  replicaSet = myreplset "
client  =  MongoClien  t ( uri ,
                    ssl = True ,
                    ssl_ca_cer  ts =' 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.

```
mongoc_cli  ent_t  * client  =  NULL ;
client  =  mongoc_cli  ent_new  (
     " mongodb :// host01 : 27017 , host02 : 27017 , host03 : 27017
/?  replicaSet = myreplset & ssl = true ");
const   mongoc_ssl  _opt_t  * ssl_defaul  t  =  mongoc_ssl
_opt_get_d  efault  ();
mongoc_ssl  _opt_t   ssl_opts  = {  0  };

/*  Optionally   copy   a   certificat  e   in   a   custom   trust
directory   or   file ;  otherwise   the   default   is   used . */
memcpy  (& ssl_opts ,  ssl_defaul  t ,  sizeof   ssl_opts );
ssl_opts . ca_file  = "/ path / to / ca . pem "
ssl_opts . allow_inva  lid_hostna  me  =  false
mongoc_cli  ent_set_ss  l_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.

Note:

Currently, you cannot ignore host name verification for the MongoDB C++ driver.

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

 mongocxx :: options :: client   client_opt  ions ;
 mongocxx :: options :: ssl   ssl_option  s ;

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

 client_opt  ions . ssl_opts ( ssl_option  s );

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

Scala

Related link: MongoDB Scala Driver

Sample code

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.

```
import   org . mongodb . scala . connection .{  NettyStrea
mFactoryFa  ctory ,  SslSetting  s }

MongoClien  tSettings . builder ()
                  . sslSetting  s ( SslSetting  s . builder ()
                                        . enabled ( true )

                                        . invalidHos  tNameAllow
ed ( true )
                                        . build ())

                  . streamFact  oryFactory ( NettyStrea  mFactoryFa
ctory ())
                  . build ()
val   client :  MongoClien  t  =  MongoClien  t (" 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 - trustcacer ts - file < path to
certificat e authority file >
       - keystore < path to trust store > - storepass <
password >
```

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

```
System . setPropert y (" javax . net . ssl . trustStore ","/ trust /
mongoStore . ts ");
System . setPropert y (" javax . net . ssl . trustStore Password
"," 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.

```
import (
   " crypto / tls "
   " crypto / x509 "
   " gopkg . in / mgo . v2
)
rootPEM , err := ioutil . ReadFile (" path / to / ca . pem ")
roots := x509 . NewCertPoo l ()
ok := roots . AppendCert sFromPEM ([] byte ( rootPEM )
tlsConfig := & tls . Config {
                  RootCAs : roots ,
       InsecureSk ipVerify : 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 := DialWithIn fo ( dialInfo )
if err ! = nil {
    panic ( err )
}
session . Close ()
```

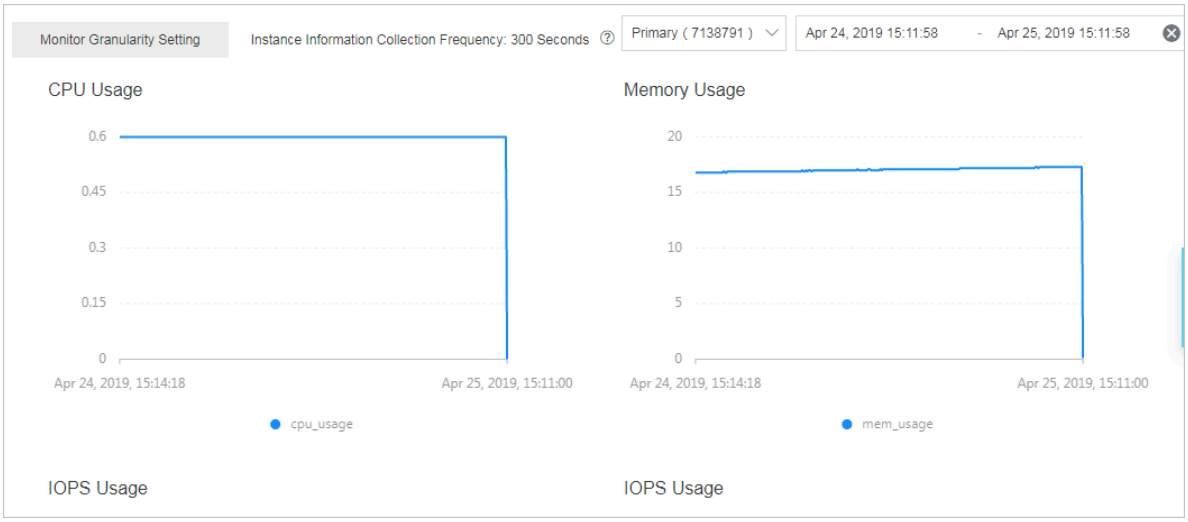
10 Monitoring and alerting

10.1 View the monitoring information

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

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, the monitoring data of the last 24 hours is displayed by default. You can also specify a time range to view the historical monitoring data.



Note:

- If the target instance is a replica set instance, you can select Primary or Secondary to view the monitoring information about the primary node or a secondary node.

- If the target instance is a sharded cluster instance, you can select Mongos or Shard to view the monitoring information about the selected node.

Metrics

Metric	Description
CPU Usage	The CPU usage of the instance.
Memory Usage	The memory usage of the instance.
IOPS Usage	The input/output operations per second (IOPS) of the instance, including: · The IOPS on the data disk. · The IOPS on the log disk.
IOPS Usage	The ratio of the IOPS of the instance to the maximum IOPS.
Disk Space Usage	The disk space usage of the instance, including: · The total disk space usage. · The disk space usage on the data disk. · The disk space usage on the log disk.
Disk Space Usage	The ratio of the total disk space usage of the instance to the maximum available disk space.
Opcounters	The queries per second (QPS) of operations on the instance, including: · The number of insert operations. · The number of query operations. · The number of delete operations. · The number of update operations. · The number of getmore operations. · The number of commands.
Connections	The current number of connections to the instance.
Cursors	The current number of cursors used by the instance, including: · The number of currently opened cursors. · The number of timed-out cursors.
Network	The network traffic of the instance, including: · The inbound traffic. · The outbound traffic. · The number of processed requests.

Metric	Description
GlobalLock	The number of operations that are currently queued and waiting for the global lock, including: · The number of operations queued waiting for the read lock. · The number of operations queued waiting for the write lock. · The total number of operations queued waiting for the global lock.
WiredTiger	The statistics on the cache of the WiredTiger storage engine, including: · The amount of data read into the cache. · The amount of data written from the cache. · The maximum configured size of the cache.

10.2 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.

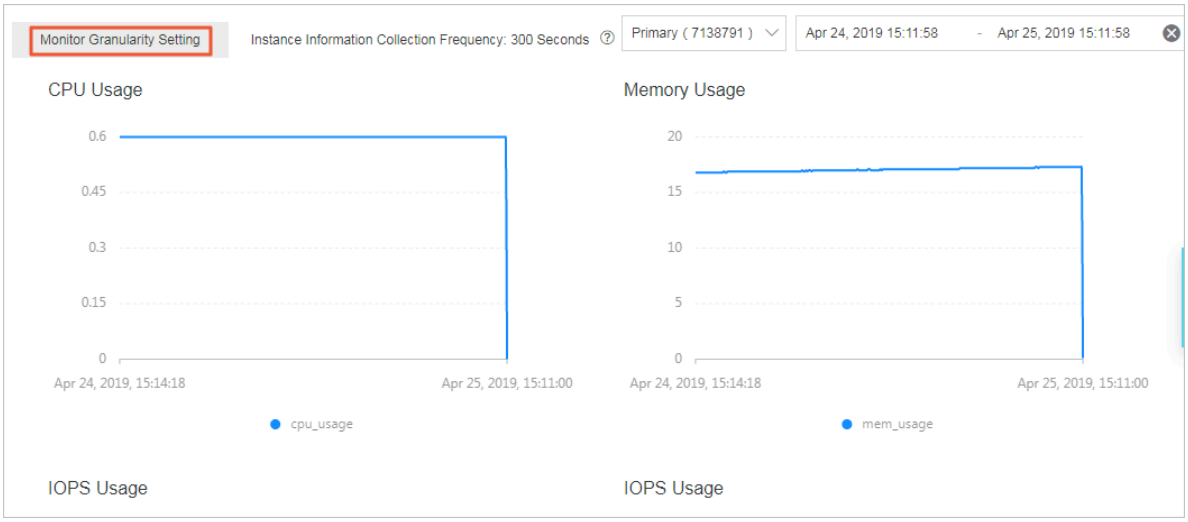
- For ApsaraDB for MongoDB instances created before December 5, 2017 with the database version of MongoDB 3.4, if they have been restarted once since December 5, 2017, they can be automatically upgraded to the latest minor database version. If they have never been restarted since December 5, 2017, you 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.

Metric	Every second	Every 300s
Disk Space Usage	N/A	Supported in MongoDB 3.2, MongoDB 3.4, and MongoDB 4.0.
Disk Space Usage		
CPU Usage	Supported in MongoDB 3.4 (upgraded to the latest minor database version) and MongoDB 4.0.	
Memory Usage		
IOPS Usage		
Opcounters		
Connections		
Cursors		
Network		
GlobalLock		
WiredTiger		

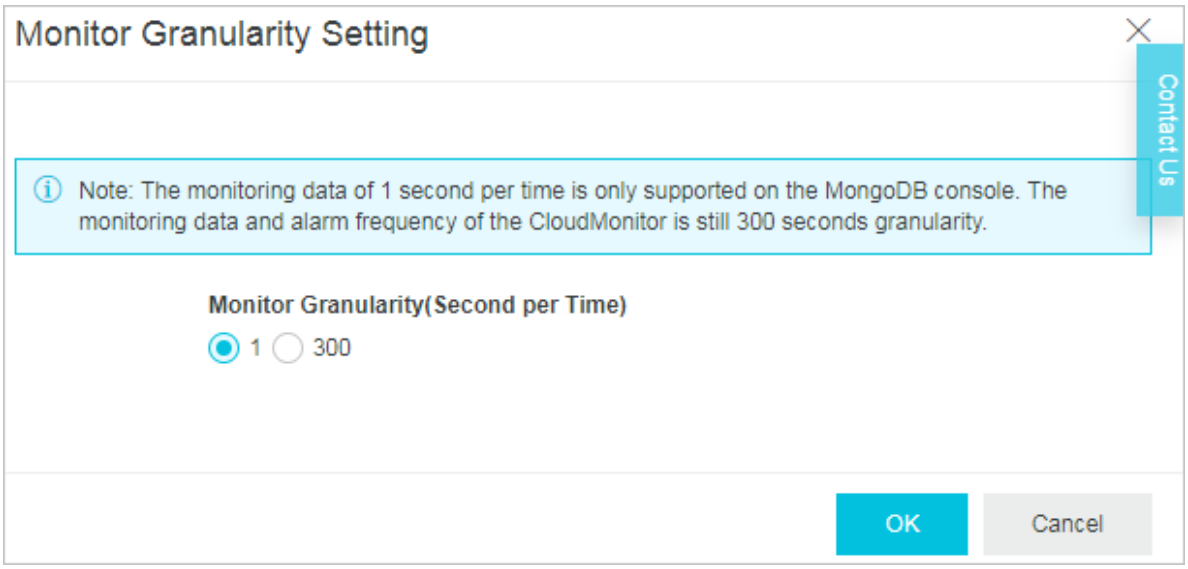
Procedure

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.

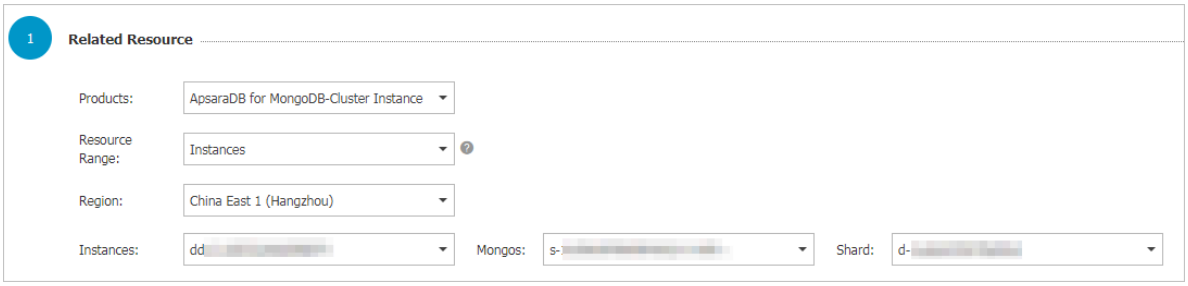
10.3 Set alert rules

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.



Parameter	Description
Products	The architecture of the instance. · ApsaraDB for MongoDB-Instance Copy · ApsaraDB for MongoDB-Cluster Instance · ApsaraDB for MongoDB-Single node instance Note: If you select ApsaraDB for MongoDB-Cluster Instance, you need to select the mongos nodes and shards to be monitored for Mongos and Shard, respectively.
Resource Range	· If you select All Resources, the alerting service sends an alert notification when any ApsaraDB for MongoDB instances match alert rules. · If you select Instances, the alerting service sends an alert notification when any selected ApsaraDB for MongoDB instances match alert rules.
Region	The region where the instance is located.
Instances	The ID of the instance to be monitored. You can select multiple instance IDs.

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.

11 Parameter settings

11.1 Set database parameters

ApsaraDB for MongoDB allows you to set part of database parameters based on your own requirements to adapt the features of ApsaraDB for MongoDB properly to your business.

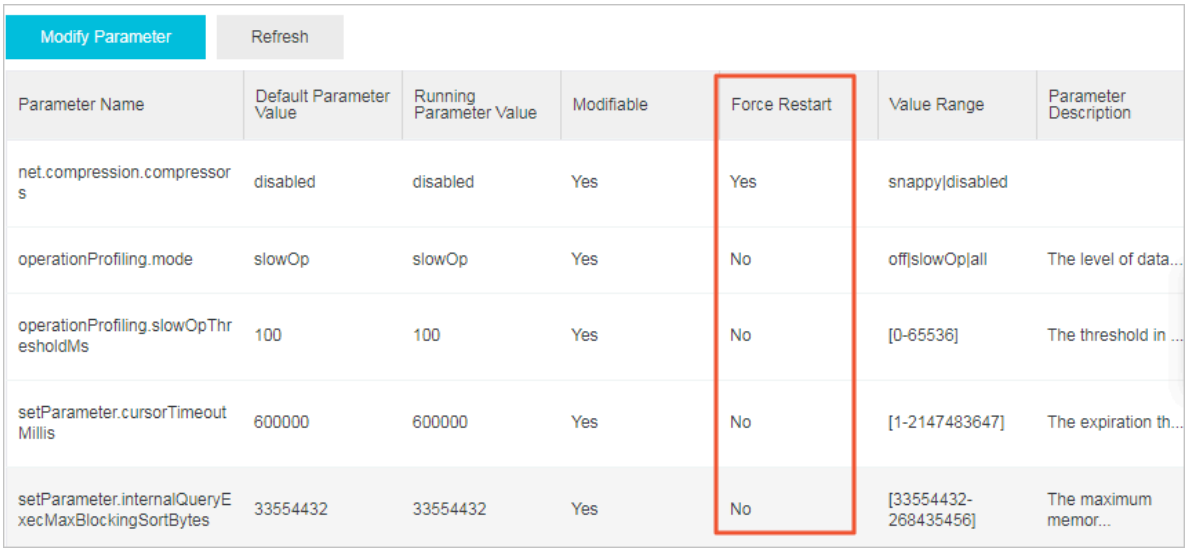
Notes

- You can set database parameters for standalone and replica set instances, but not for sharded cluster instances.
- You must modify parameter values within their respective value ranges specified in the console.
- After you submit some parameters to be modified for an instance, the instance is automatically restarted. For more information, see the Force Restart column on the Parameter List page. Because an instance may be disconnected during a restart, you need to restart it with caution and make arrangements for business interruption.

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 > Parameter List.

6. Click Modify Parameter.



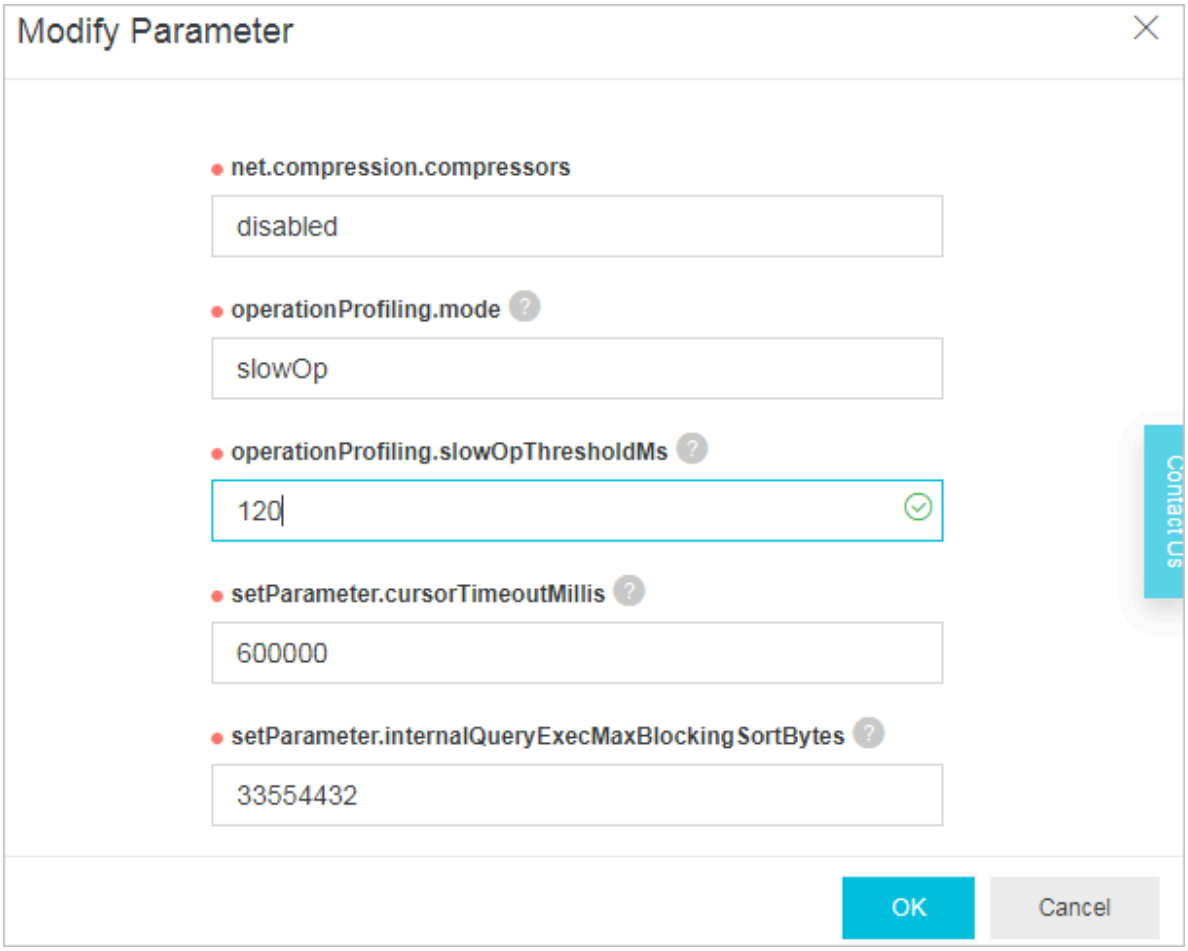
Modify Parameter | Refresh

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

Note:

On the parameter list, you can view all parameters that you can modify, whether the instance needs to be restarted, and the effective rule for each parameter.

7. On the Modify Parameter page that appears, modify parameters as required.



You can modify multiple parameters in this step.

8. Click OK.

11.2 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.

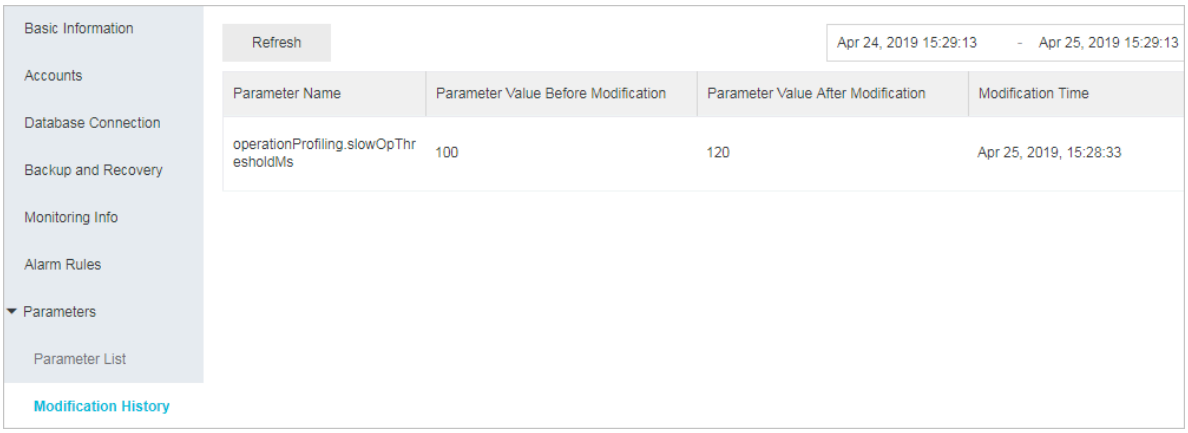
Prerequisites

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.

12 Primary/Secondary failover

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

An ApsaraDB for MongoDB replica set instance consists of three nodes by default. ApsaraDB for MongoDB provides 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.

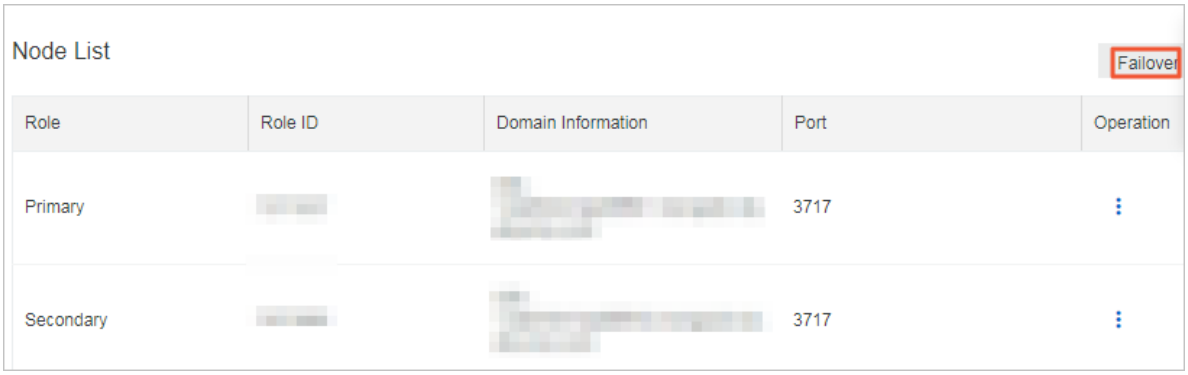
Note:

- 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.



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.

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

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

Notes

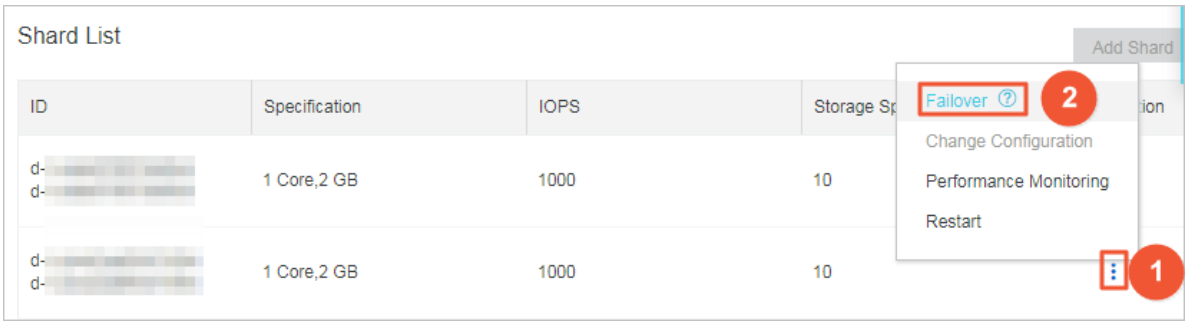
ApsaraDB for MongoDB provides addresses for you to connect to the primary node and a secondary node of a shard. The other secondary node is hidden as a backup to guarantee high availability. After you log on to the ApsaraDB for MongoDB console or call the SwitchDBInstanceHA operation to trigger a primary/secondary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

shard of a sharded cluster instance, ApsaraDB for MongoDB interchanges the roles of the primary and secondary nodes.

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

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the 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.**

13 Data backup

13.1 Automatically back up ApsaraDB for MongoDB data

ApsaraDB for MongoDB automatically backs up data according to the default backup policy. You can also set a backup policy based on business requirements to automatically back up data for your ApsaraDB for MongoDB instances as required.

Notes

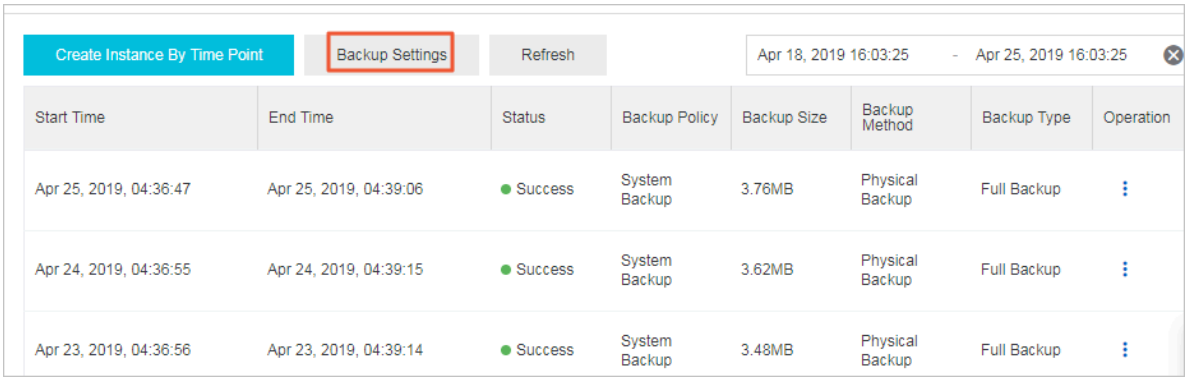
- ApsaraDB for MongoDB stores its generated backup files in OSS to free up the storage space of ApsaraDB for MongoDB instances.
- The backup method for standalone instances is fixed to snapshot backup, which affects their I/O performance in the backup process.
- The backup method for replica set and sharded cluster instances is physical backup.

> **Note:**
> 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.

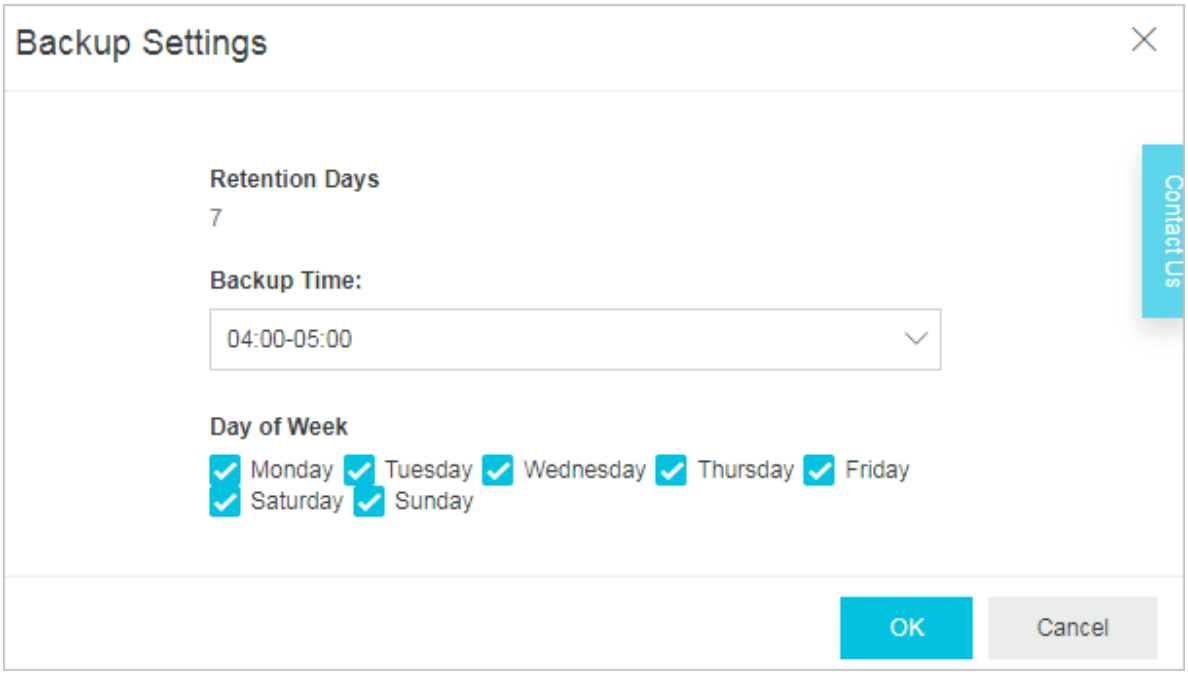
Set an automatic backup policy

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 Backup and Recovery.

6. Click Backup Settings.



7. In the Backup Settings dialog box that appears, set related parameters.



Parameter	Description
Retention Days	The number of days for keeping backup data. It is fixed to seven days.
Backup Time	The backup time in units of hours. You can set any time as required. We recommend that you back up data during off-peak hours.
Day of Week	The backup cycle. You can select one or more days in a week.

8. After setting the preceding parameters, click OK.

13.2 Manually back up ApsaraDB for MongoDB data

You can set a backup policy to adjust the default backup settings of ApsaraDB for MongoDB to automatically back up data. Alternatively, you can manually back up ApsaraDB for MongoDB data.

Notes

- ApsaraDB for MongoDB stores its generated backup files in OSS to free up the storage space of ApsaraDB for MongoDB instances.
- The backup method for standalone instances is fixed to snapshot backup, which affects their I/O performance in the backup process.
- Replica set and sharded cluster instances support physical backup and logical backup.
- A physical backup or logical 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.

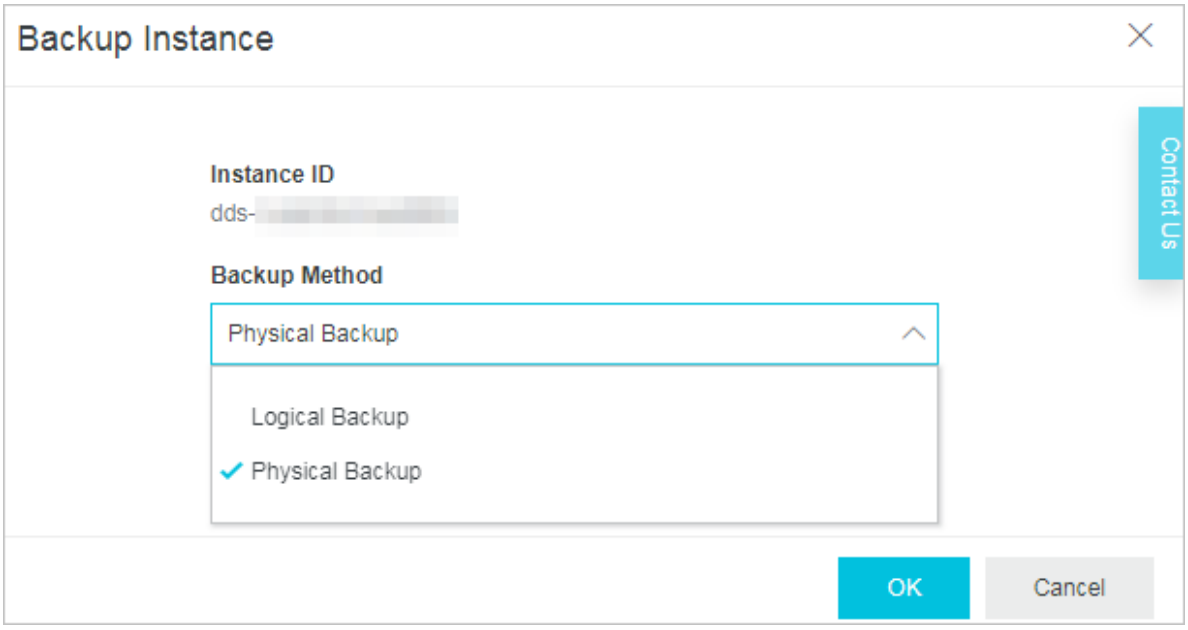
Backup methods

- Snapshot backup: Due to the special single-node architecture, the data of standalone instances is backed up in snapshots. A snapshot backup can keep the status of disk data at a specific time point.
- Physical backup: Physical database files are backed up for ApsaraDB for MongoDB instances.
- Logical backup: You can run a mongodump command to logically back up ApsaraDB for MongoDB data.

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 Backup and Recovery. In the upper-right corner of the page that appears, click Backup Instance.

6. In the Backup Instance dialog box that appears, select a backup method for Backup Method.



7. Click OK.

14 Data recovery

14.1 Restore a database in ApsaraDB for MongoDB

ApsaraDB for MongoDB supports database restoration. You can restore the data of a certain database or databases back to a specific point in time in a newly-created instance.

Instruction

Instances created after March 26, 2019 support database restoration. This function will be open to existing instances in the future. Follow the official website for update information.

Prerequisites

- The instance is created after March 26, 2019.
- The instance is located in China (Qingdao), China (Beijing), China (Zhangjiakou-Beijing Winter Olympics), China (Hohhot), China (Hangzhou), China (Shanghai), China (Shenzhen), or Singapore. Other regions are not supported.
- The instance type is replica set. Standalone and sharded cluster instances are not supported.
- The instance version is 3.4 or 4.0. V3.2 is not supported.
- The storage engine of the instance is WiredTiger. RocksDB and TerarkDB are not supported.
- The backup file in the instance backup list contains the database to be restored.

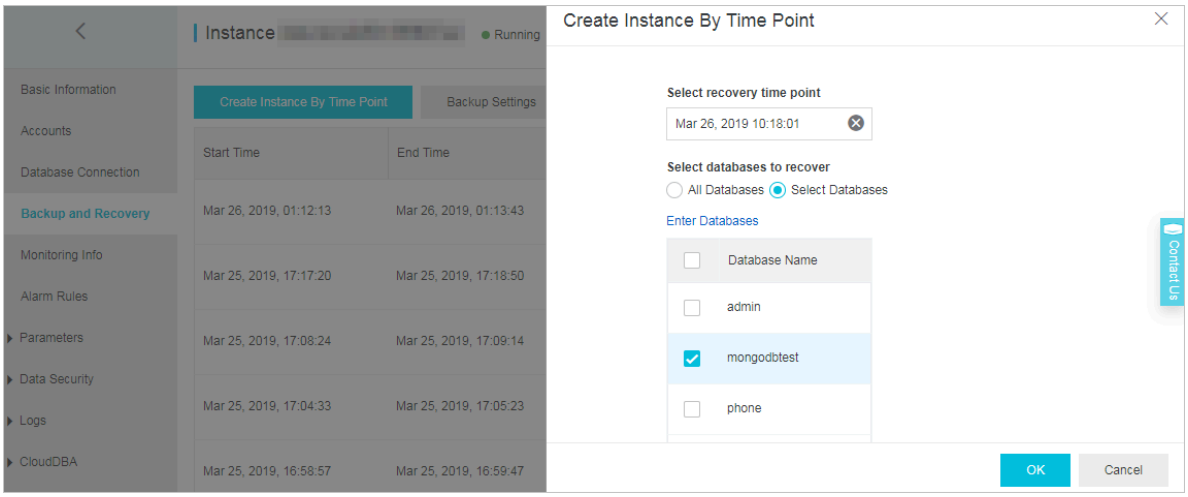
Precautions

- You can only restore databases from physical backups. Restoring databases from logical backups is not supported.
- You can only restore databases to newly-created instances. Restoring databases to each original instance is not supported.

> Note:
> An instance will be created during database restoration. For the billing method of the instance, see #unique_16.

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. Locate the instance to be restored and click the instance 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, set the parameters listed in the following table.



Parameter	Description
Select recovery time point	Select a point in time that you want to restore the database to. You can restore the database to any point within the past seven days. **Note:** The time must be earlier than the current time, and must be later than the time when the database was created.
Select recovery database	· All Databases · Select Databases: Restore certain databases in the instance. You can select the databases that you want to restore or click Enter Databases to enter databases manually. **Note:** If you want to restore more than one database, separate the multiple database names with commas (,).

8. Click OK. The Instance Purchase page appears.

9. Configure the new instance.

> Note:
> In the Basic Configuration area, do not configure Region, Database Version, Storage Engine, and Node Number.

Category	Parameter	Description
Basic Configuration	Zone	A zone indicates a physical area in a region with independent power grids and networks. For more information, see Regions and zones. ApsaraDB for MongoDB and ECS instances between different zones in the same region can be connected through the internal network. For more information, see #unique_82. Note: When the ECS instance and the ApsaraDB for MongoDB instance in the same zone are connected through the internal network, the network latency is the minimum.
Network Type	-	· Classic Network: Cloud services on a classic network are not isolated, and unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service. · VPC (recommended): A Virtual Private Cloud (VPC) is an isolated network with higher security and performance than the traditional classic network. The VPC is to be created before creating an instance. For more information, see Configure a VPC for a new instance.
Specification Configuration	Specification	The CPU and memory occupied by the instance. · The maximum number of connections and maximum IOPS vary depending on different specifications. The maximum IOPS is measured respectively under the read and write permissions, and can double under the mixed read/write permissions.

Category	Parameter	Description
	Storage Space	The storage space exclusive to each node in the replica set instance. Note: The storage space of a node stores your data, system, and log files.
Password Settings	· Set Now · Set Later	Set the password of the MongoDB database. You can set the password when creating or running the instance. · The password must contain characters from at least three of the following categories: uppercase letters, lowercase letters, digits, and special characters. Special characters include !#$%^&*()_+-= · The password must be 8 to 32 characters in length.
Quantity Purchased	Duration	Subscription: Set the duration and quantity of the subscription instance. The subscription duration can be one to nine months or one to three years. Note: This parameter is required only for subscription instances.
	Quantity	Set the number of instances with the same specifications. It can be set to an integer of 1 to 10.

10. Click Buy Now. The Confirm Order page appears.
11. On the Confirm Order page, read and select ApsaraDB for MongoDB Agreement of Service, and make the payment as prompted.

14.2 Create an instance based on a backup

ApsaraDB for MongoDB allows you to create an instance based on a backup of an existing instance. The data of the new instance is recovered from that of the selected backup. This method applies to data recovery or data verification scenarios.

Prerequisites

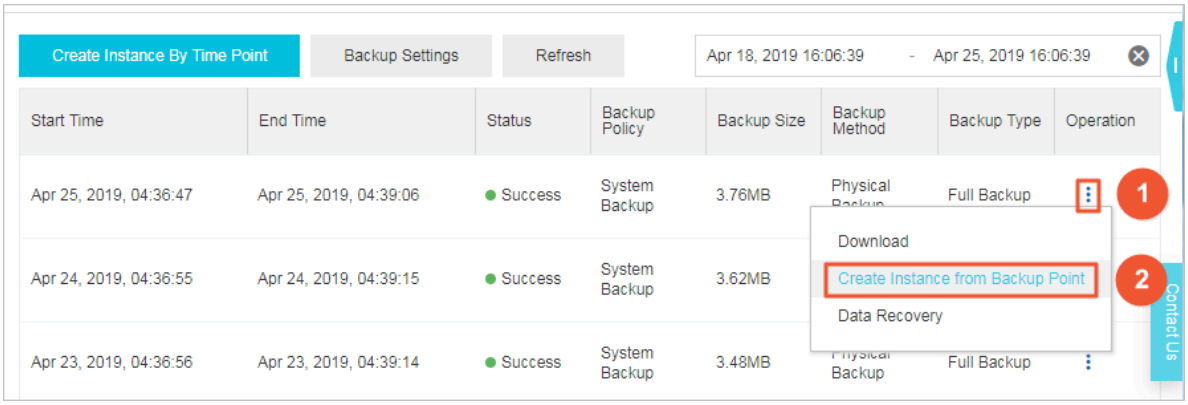
· The existing instance must be a standalone or replica set instance.

Note:
Sharded cluster instances do not support this feature.

- Currently, you can select a backup only in the last seven days.
- If you create an instance based on a backup, you need to pay for the created instance. For more information about how to bill an instance, see Billing items and pricing.
- To create a Pay-As-You-Go instance, ensure that your account balance is sufficient.

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 backup and choose > Create Instance from Backup Point in the Operation column.



7. On the ApsaraDB for MongoDB instance creation page that appears, select the billing method of the instance to be created.
8. Set parameters for the instance as required.

> Note:
> The storage space of the new instance must be larger than or equal to that of the existing instance.

9. Click Buy Now to go to the Confirm Order page.
10. Read and select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the payment process.

14.3 Create an instance based on a time point

ApsaraDB for MongoDB allows you to create an instance based on a running time point of an existing instance. The data of the new instance is recovered from that of the existing instance at the selected time point. This method applies to data recovery or data verification scenarios.

Prerequisites

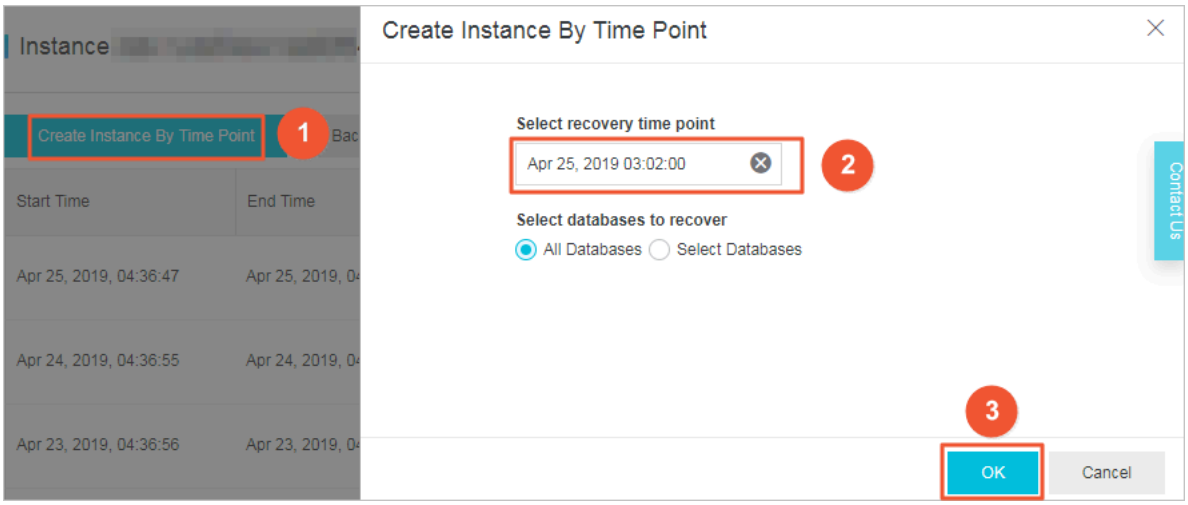
- The existing instance must be a replica set or sharded cluster instance.

   Note:
  Standalone instances do not support this feature.
- Currently, you can select a time point only in the last seven days.
- If you create an instance based on a time point, you need to pay for the created instance. For more information about how to bill an instance, see Billing items and pricing.
- To create a Pay-As-You-Go instance, ensure that your account balance is sufficient.

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

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



8. On the ApsaraDB for MongoDB instance creation page that appears, select the billing method of the instance to be created.
9. Set parameters for the instance as required.

> Note:
>
> · Replica set instance: The storage space of the new instance must be larger than or equal to that of the existing instance.
> · Sharded cluster instance:
>
> - The number of shards for the new instance must be greater than or equal to that for the existing instance.
> - The storage space of each shard for the new instance must be larger than or equal to that for the existing instance.

10. Click Buy Now to go to the Confirm Order page.
11. Read and select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the payment process.

14.4 Recover backup data in the current instance

Data recovery can help you minimize the loss caused by database misoperations. ApsaraDB for MongoDB provides multiple recovery methods. This topic describes how to recover backup data in the current instance.

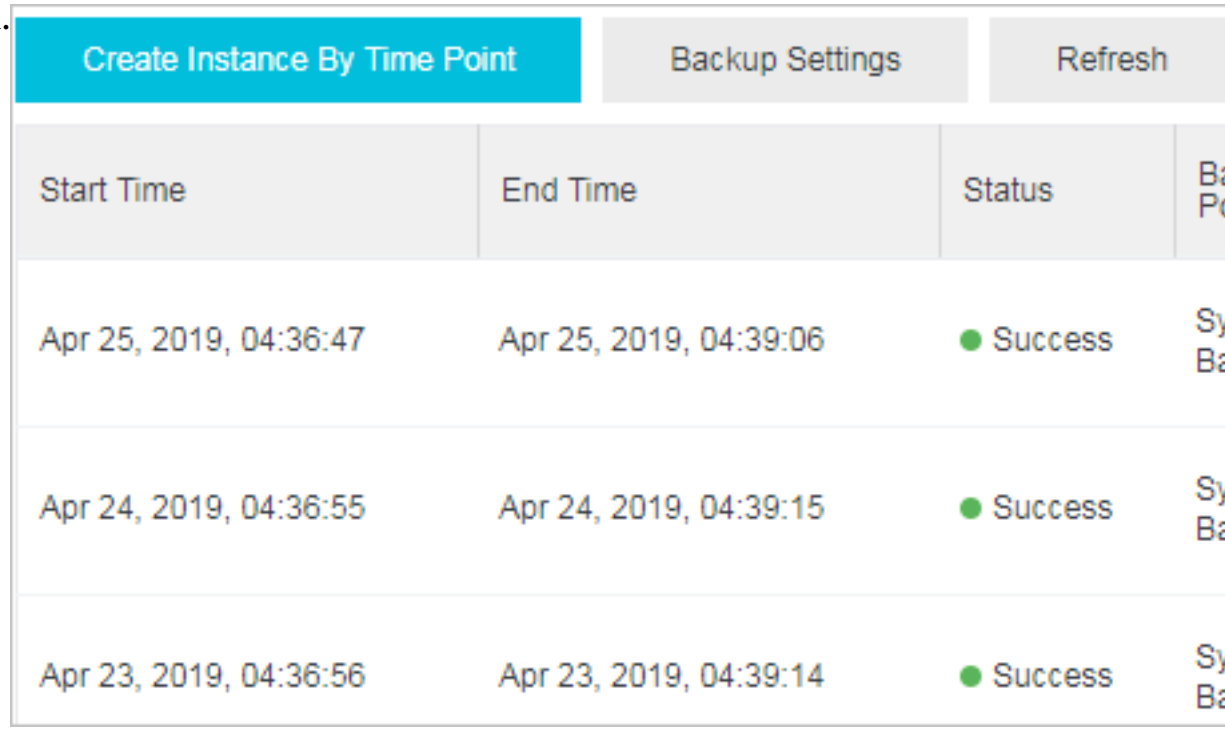
Notes

- Currently, only three-node replica set instances support this feature.
- If you recover backup data in the current instance, the original data of the current instance is overwritten and cannot be recovered. Therefore, you need to perform this operation with caution.
- Considering high risks for recovering backup data in the current instance, we recommend that you create an instance based on a time point or backup to recover data. For more information, see #unique_88 and #unique_89 . After verifying the recovered data, use DTS to migrate data to the current instance.

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 backup and choose > Data Recovery in the Operation column.



7. In the Recover Backup Instance dialog box that appears, click OK.
8. The instance enters the Restoring from Backup status. You can click Refresh to update and check the instance status. The data is successfully recovered when the instance status changes to Running.

14.5 Recover logical backup data in a user-created MongoDB instance

Replica set and sharded cluster instances support logical backup. You can start a full logical backup to back up instance data and download the logical backup file. Then, you can run a mongorestore command to recover the downloaded backup data in a 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 #unique_89.

To guarantee compatibility, we recommend that the database version of the user-created MongoDB instance be the same as that of the ApsaraDB for MongoDB instance .

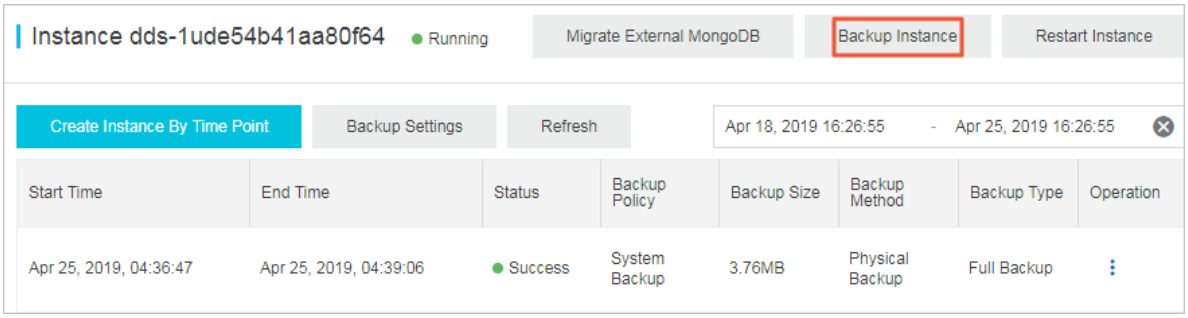
Context

You can run a mongodump command to start a full logical backup to back up ApsaraDB for MongoDB data. During the backup, you can still read data from and write data into the ApsaraDB for MongoDB instance.

> **Note:**
> A full logical 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 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 Backup and Recovery.
6. In the upper-right corner of the Backup and Recovery page that appears, click Backup Instance.



7. In the Backup Instance dialog box that appears, select Logical Backup as the backup method.
8. Click OK and wait until the instance data is successfully backed up.
9. On the Backup and Recovery page, locate the completed logical backup and choose > Download in the Operation column.

10. After downloading the backup file, run the following command to import the backup data into the user-created MongoDB instance:

```
mongoresto re - h < hostname > -- port < server port > - u
 < username > - p < password > -- drop -- gzip -- archive =<
backupfile > - vvvv -- stopOnErro r
```

Notes:

- <hostname>: The server address of the user-created MongoDB instance. Set this parameter to 127.0.0.1 if the user-created MongoDB instance is deployed on the current server.
- <server port>: The port used by the user-created MongoDB instance.
- <username>: The database username used to log on to the user-created MongoDB instance.
- <password>: The database password used to log on to the user-created MongoDB instance.
- <backupfile>: The name of the downloaded logical backup file.

Example:

```
mongoresto re - h 127 . 0 . 0 . 1 -- port 27017 - u
root - p xxxxxxxx -- drop -- gzip -- archive = hins1111_d
ata_201907 10 . ar - vvvv -- stopOnErro r
```

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

14.6.1 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 #unique_89.

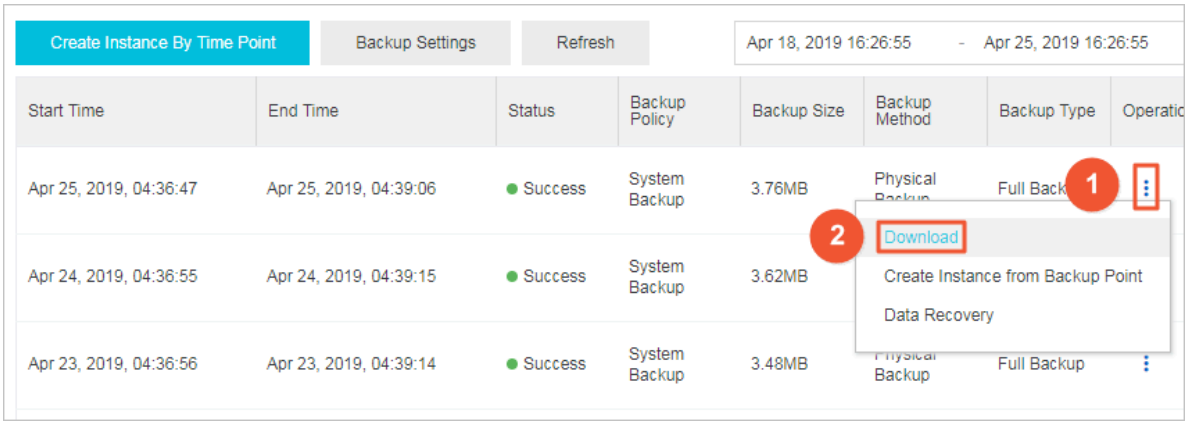
Context

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.

14.6.2 Recover ApsaraDB for MongoDB physical backup data in a user-created MongoDB instance

You can log on to the ApsaraDB for MongoDB console to download a physical backup file from an ApsaraDB for MongoDB instance. This topic describes how to recover ApsaraDB for MongoDB physical backup data in a user-created MongoDB instance.

Prerequisites

- This feature applies only to ApsaraDB for MongoDB replica set instances.
- The storage engine of the ApsaraDB for MongoDB instance must be WiredTiger or RocksDB.

  Note:
  If the storage engine of the ApsaraDB for MongoDB instance is TerarkDB, you can recover logical backup data in the user-created MongoDB instance. For more information, see #unique_95.
- If the storage engine of the ApsaraDB for MongoDB instance is RocksDB, you need to compile and install a MongoDB application that is configured with the RocksDB storage engine.
- The user-created MongoDB instance must be compatible with the ApsaraDB for MongoDB instance. The following table lists the mappings between the database versions of the ApsaraDB for MongoDB instance and the user-created MongoDB instance.

Database version of the ApsaraDB for MongoDB instance	Database version of the user-created MongoDB instance
MongoDB 3.2	MongoDB 3.2 or MongoDB 3.4
MongoDB 3.4	MongoDB 3.4
MongoDB 4.0	MongoDB 4.0

Preparations

The following procedure uses a Linux server as an example. (MongoDB of the required version has been installed on the Linux server. For more information about how to install MongoDB, see the official MongoDB manual.)

Download a physical backup file from an ApsaraDB for MongoDB instance and decompress the downloaded file in the user-created MongoDB instance.

1. Download a physical backup file from an ApsaraDB for MongoDB instance.
2. Clear data from the data directory (which must be empty) where MongoDB is installed on the local server.

   Assume that */ path / to / mongo* is the directory used for physical recovery operations of the user-created MongoDB instance.

   ```
   cd  / path / to / mongo / data /
   rm  - rf  *
   ```

3. Copy the downloaded ApsaraDB for MongoDB physical backup file to the */ path / to / mongo / data /* directory and decompress the file.

   ```
   tar   xzvf   hins_xxx . tar . gz
   ```

Recover ApsaraDB for MongoDB physical backup data in standalone mode

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

   ```
   touch   mongod . conf
   ```

2. Modify the mongod.conf file to meet the configuration requirements for starting the user-created MongoDB instance recovered from ApsaraDB for MongoDB physical backup data.

   Based on the storage engine, select a configuration template for the startup of the user-created MongoDB instance in standalone mode with authentication enabled. You can copy the selected configuration template to the mongod.conf file.

   - WiredTiger

     ```
     systemLog :
         destinatio  n :  file
         path : / path / to / mongo / mongod . log
         logAppend :  true
     security :
         authorizat  ion :  enabled
     storage :
         dbPath : / path / to / mongo / data
         directoryP  erDB :  true
     net :
         http :
             enabled :  false
         port :  27017
         unixDomain  Socket :
             enabled :  false
     processMan  agement :
         fork :  true
     ```

```
    pidFilePat  h : / path / to / mongo / mongod . pid
```

Note:

The ApsaraDB for MongoDB instance is configured with the WiredTiger storage engine by default, and the directoryPerDB option is enabled. Therefore, this option is specified in the configuration.

- **RocksDB**

```
systemLog :
 destinatio  n :  file
 path : / path / to / mongo / logs / mongod . log
 logAppend :  true
security :
 authorizat  ion :  enabled
storage :
 dbPath : / path / to / mongo / data
        engine :  rocksdb
net :
 http :
  enabled :  false
 port :  27017
 unixDomain  Socket :
  enabled :  false
processMan  agement :
 fork :  true
 pidFilePat  h : / path / to / mongo / logs / mongod . pid
```

3. **Use the new configuration file mongod.conf to start the user-created MongoDB instance.**

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

4. **Log on to the user-created MongoDB instance through the mongo shell on the local server.**

```
mongo  -- host  127 . 0 . 0 . 1  - u  < username > - p  < password
> -- authentica  tionDataba  se  admin
```

Notes:

- **<username>: The database username used to log on to the ApsaraDB for MongoDB instance. The default username is root.**
- **<password>: The database password used to log on to the ApsaraDB for MongoDB instance.**

Start the user-created MongoDB instance in replica set mode

ApsaraDB for MongoDB physical backup data contains the replica set configuration of the original ApsaraDB for MongoDB instance by default. You need to start the user-created MongoDB instance in standalone mode. Otherwise, you may fail to access it.

To start the user-created MongoDB instance in replica set mode, you need to recover ApsaraDB for MongoDB physical backup data in standalone mode first and do as follows:

1. Log on to the user-created MongoDB instance through the mongo shell on the local server.
2. Remove the original replica set configuration.

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

3. Shut down MongoDB.

```
use   admin
db . shutdownSe  rver ()
```

4. Modify the mongod.conf file in the */ path / to / mongo /* directory to add the replication configuration. For more information about the command, see Deploy a Replica Set in the official MongoDB manual.
5. Use the new configuration file mongod.conf to start the user-created MongoDB instance.

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

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

Note:

This step uses an `rs . initiate ()` command. For more information about the command, see rs.initiate() in the official MongoDB manual.

15 CloudDBA

15.1 Optimize indexes

During the use of ApsaraDB for MongoDB, query statements may run slowly or time out if you miss indexes or use incorrect indexes. In this case, the high CPU usage affects business. ApsaraDB for MongoDB provides an index optimization feature to detect slow queries caused by missing or incorrect indexes. This feature also chooses the optimal indexes for these slow queries to improve the performance of ApsaraDB for MongoDB.

Constraints

- Standalone instances do not support this feature.
- Currently, you can enable index optimization only in the following regions: China (Hangzhou), China (Shanghai), China (Shenzhen), China (Qingdao), and China (Beijing).
- Before enabling index optimization for an instance, you must enable log auditing.

Rules for generating index optimization reports

- ApsaraDB for MongoDB automatically generates index optimization reports every day covering 0:00 to 24:00.

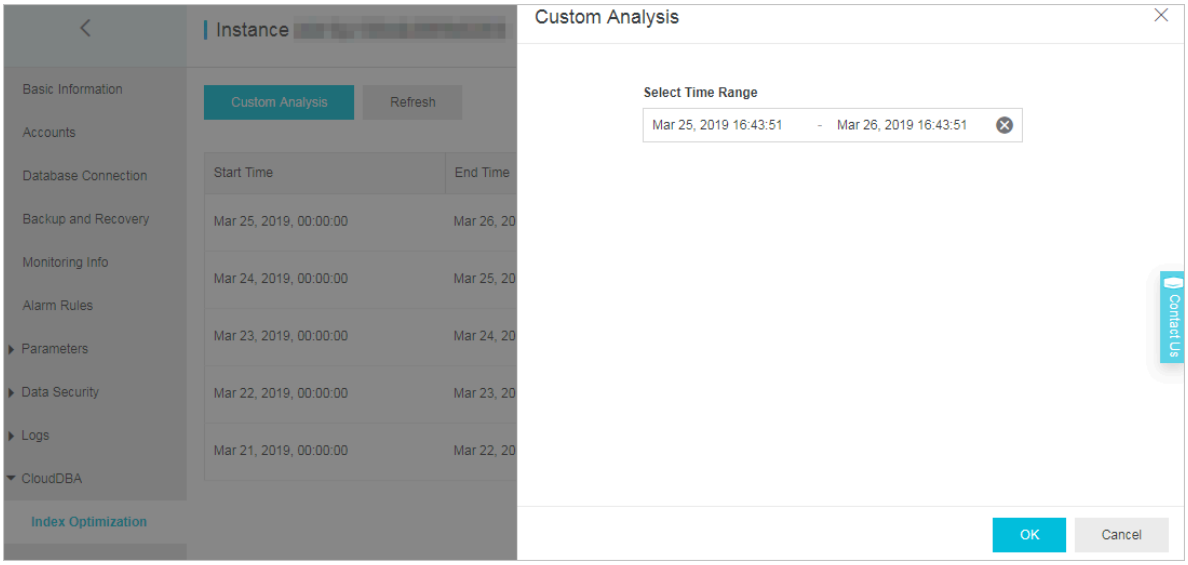
  Note:
  Index optimization reports are kept for seven days and automatically deleted after seven days.

- You can specify a time range in the last seven days to analyze slow queries within the selected time range and generate index optimization reports as required.
- Query statements whose execution time exceeds 100 ms are considered as slow queries.

Procedure

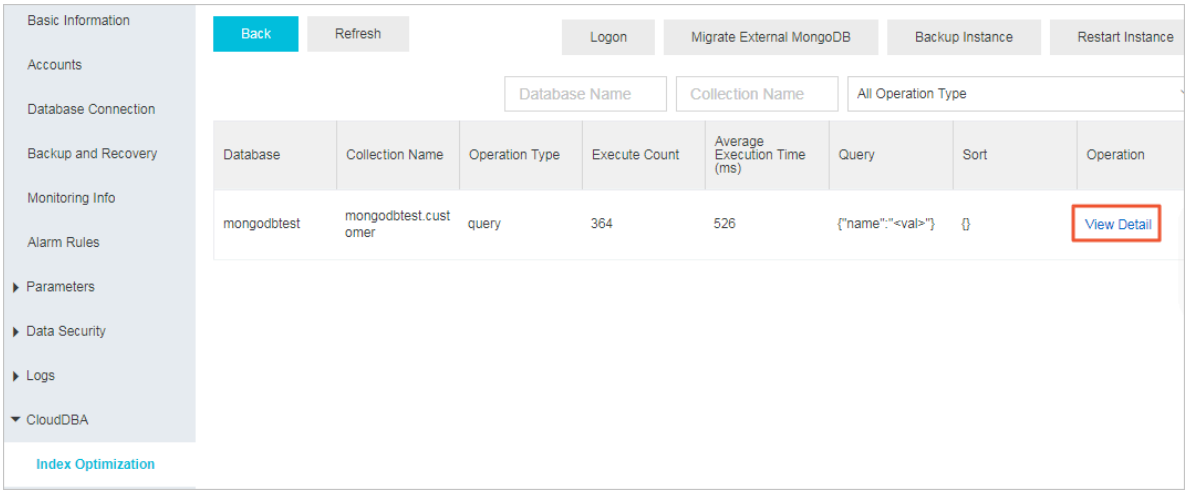
1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the home page, select the region where the target instance is located.

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, choose CloudDBA > Index Optimization.
6. Click Custom Analysis. The Custom Analysis dialog box appears.

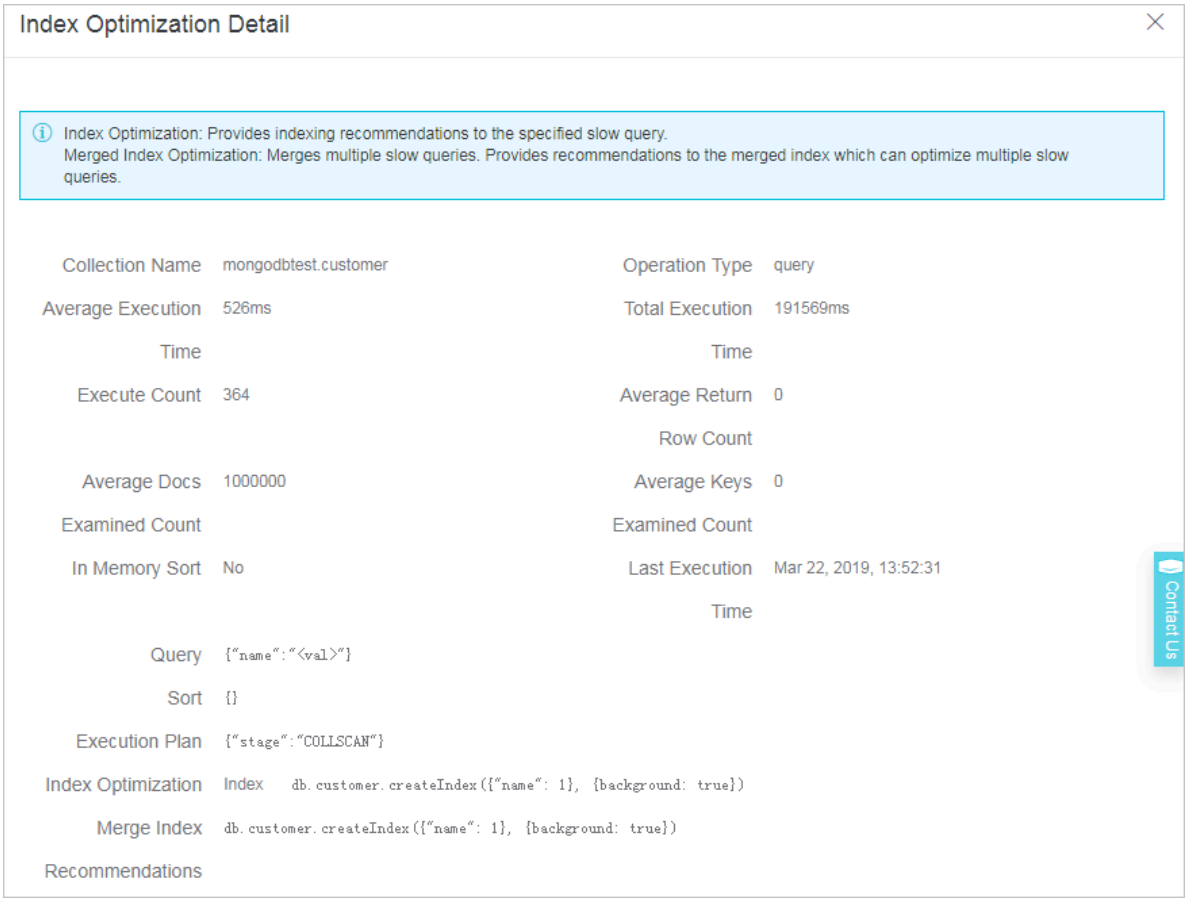


You can specify Select Time Range to view index optimization reports of the selected time range.

7. On the list, click View Detail in the Operation column corresponding to an index optimization report to view the details of index diagnosis.



8. In the Index Optimization Detail dialog box that appears, you can view the detailed information such as Index Optimization and Merge Index Recommendations.



16 Zone-disaster restoration solution

16.1 Create a multi-zone replica set instance

ApsaraDB for MongoDB provides a zone-disaster recovery solution for replica set instances. This solution deploys the three nodes of a three-node replica set instance in three different zones in the same region. The components in these zones exchange data through the internal network. If any of the three zones becomes unavailable due to unexpected events such as a power or network failure, the high availability system will automatically switch to another zone to ensure the availability of the entire replica set.

Precautions

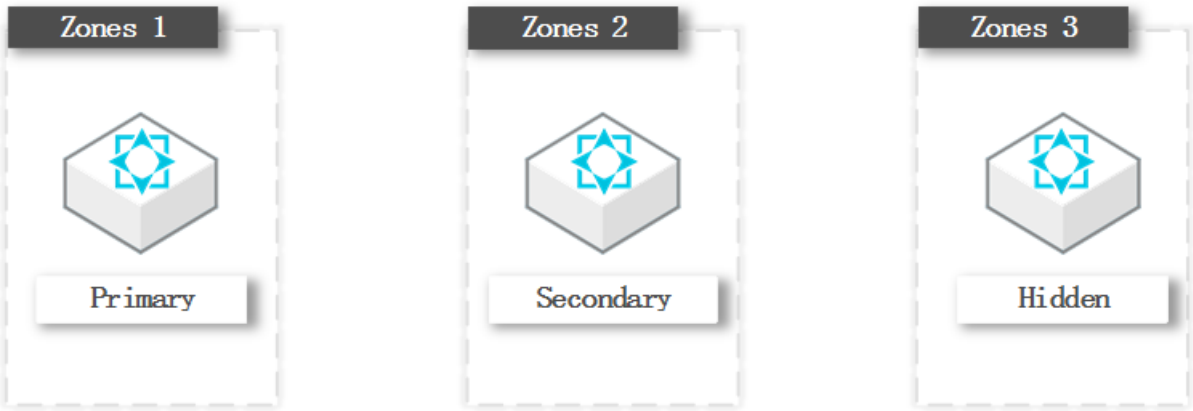
- Standalone instances do not support this function.
- You can create multi-zone sharded cluster instances in China (Hangzhou), China (Beijing), and China (Shenzhen).
- When you create a multi-zone replica set instance, you must set Replication Factor to Three Nodes Replica set.

> Note:
> If you need more nodes, you can reset the node number after you create the instance.

Node deployment policies of a replica set instance

When you create a single-zone instance, the system deploys the primary, secondary, and hidden nodes in one zone.

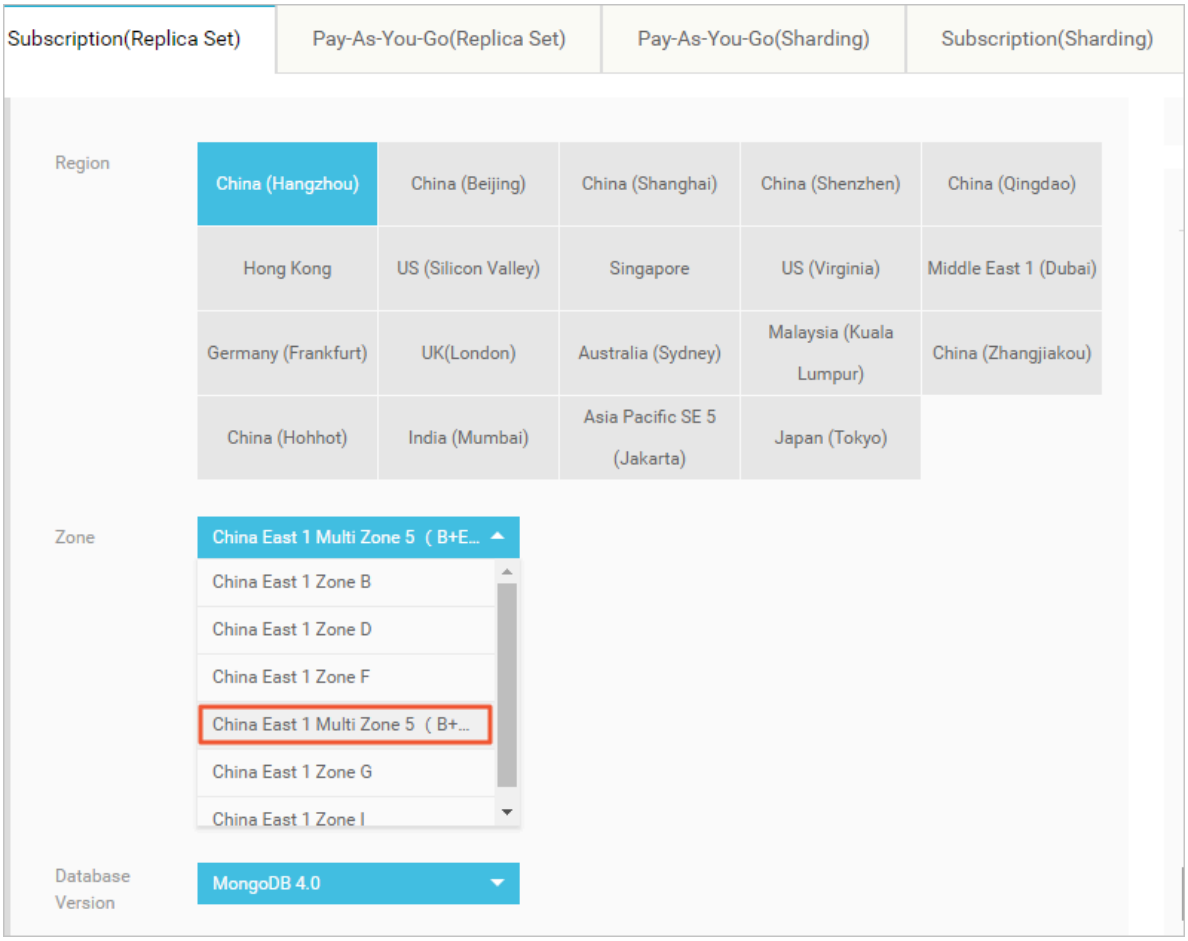
When you create a multi-zone instance, the system deploys the primary, secondary, and hidden nodes in three different zones.



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 (Shanghai), China (Shenzhen), or China (Beijing), and select a multi-zone you need.



5. For more information about other parameters for the instance, see Create an instance.
6. Click Buy Now. The Confirm Order page appears.
7. Read and select ApsaraDB for MongoDB Agreement of Service, and make the payment as prompted.

16.2 Create a multi-zone sharded cluster instance

ApsaraDB for MongoDB provides a zone-disaster recovery solution for sharded cluster instances. This solution deploys the components of a sharded cluster instance across three different zones in one region. The components in these zones exchange data through the internal network. If any of the three zones becomes unavailable due to unexpected events such as a power or network failure, the high availability system will automatically switch to another zone to ensure the availability of the entire sharded cluster.

Precautions

- Standalone instances do not support this function.
- You can create only multi-zone sharded cluster instances in China (Hangzhou), China (Beijing), and China (Shenzhen).

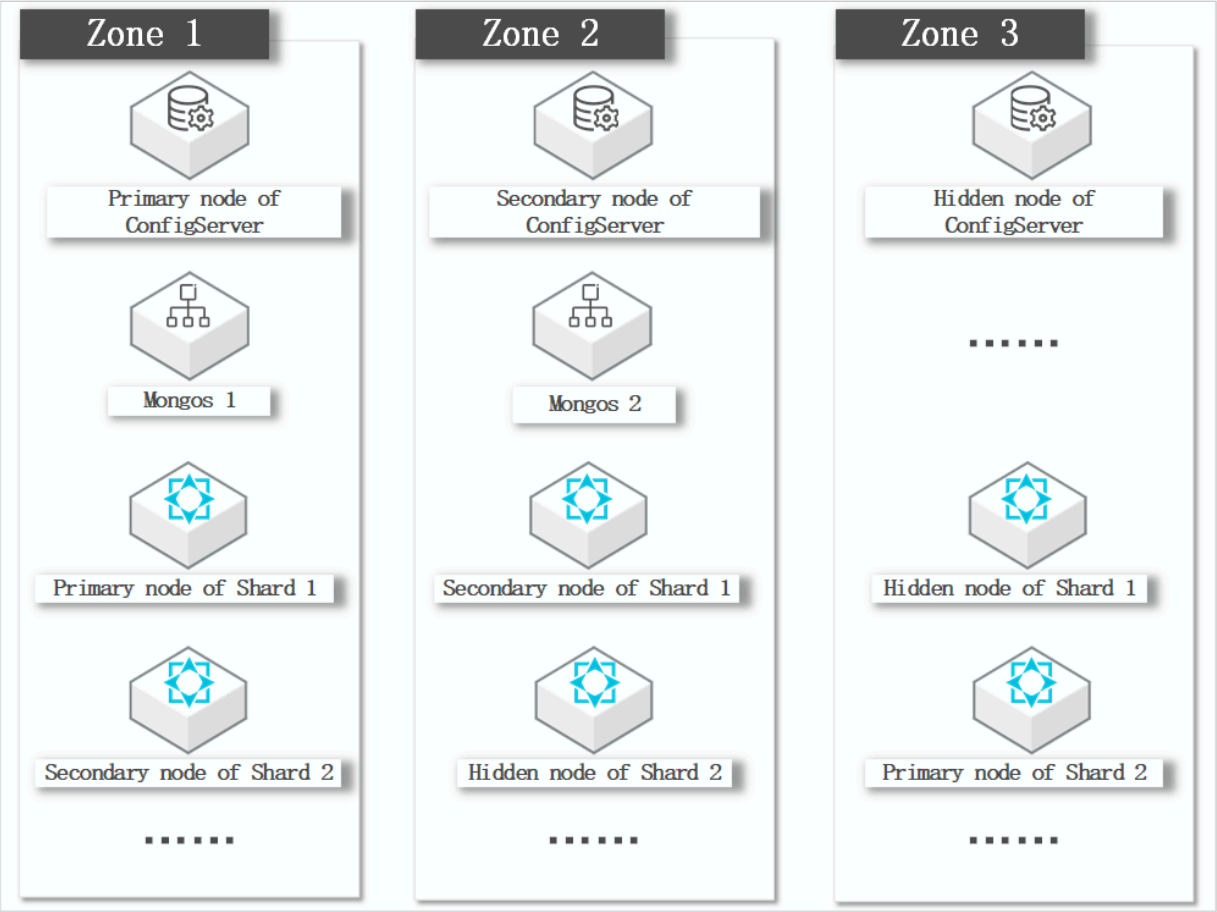
Node deployment policy

When you create a single-zone instance, the system deploys the components of the sharded cluster instance in one zone. When you create a multi-zone instance, the system deploys the components in three different zones.

- Mongos are evenly deployed across all data centers. At least two mongos are deployed across two zones. When a third mongos is added, it is deployed in the third zone. Each subsequent new mongos is deployed to each of the three zones in turn.
- The primary, secondary, and hidden nodes of each shard are not deployed to the three zones in sequence. The deployment of these nodes may change based on

whether manual switchover or HA failover between primary and secondary nodes is used.

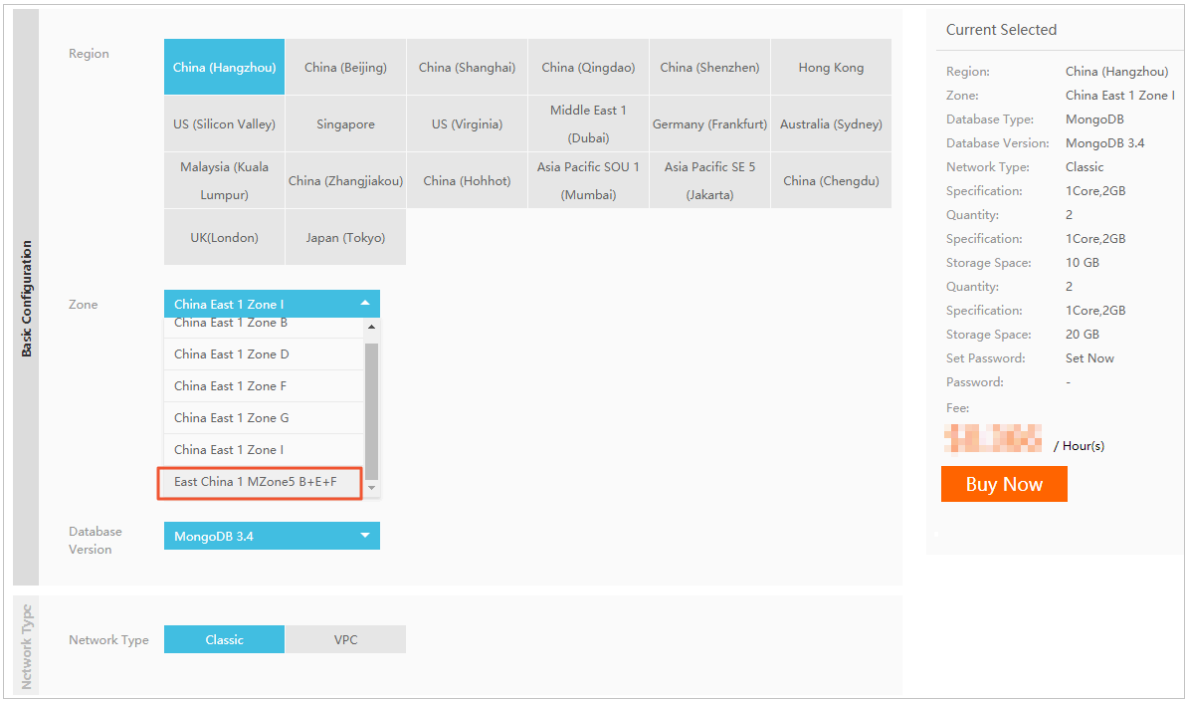
Figure 16-1: Deployment policy for the components and nodes in a multi-zone sharded cluster instance



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), or China (Shenzhen), and select the zones that you require.



5. For more information about other parameters for the instance, see #unique_105.
6. Click Buy Now. The Confirm Order page appears.
7. Read and select ApsaraDB for MongoDB Agreement of Service, and make the payment as prompted.