Alibaba Cloud ApsaraDB for MongoDB

Quick Start for Cluster

Issue: 20190731

MORE THAN JUST CLOUD | Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed 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

1 Before you start

You can migrate data from a user-created MongoDB database to an ApsaraDB for MongoDB instance. You must take note of the limits of ApsaraDB for MongoDB, as shown in the following table.

Operation	Limit
Data version and storage engine	For more information, see Data versions and storage engines.
Public IP address	Access to an instance through a public IP address may incur security risks. No public IP address is configured when an instance is created. If you need to access a MongoDB database from the Internet, you can Apply for a public IP address.
Create an instance	· When creating a sharded cluster instance, you can select the specifications and numbers of both mongos and shards. · You can add mongos or shards when the instance is running. For more information, see Change the configuration.
Restart an instance	You can only restart instances from the ApsaraDB for MongoDB console or through the API.
Migrate data	Use the built-in commands of MongoDB to migrate data or Use DTS to migrate data.
Back up data	Automatic backups must use the physical backup method. You can select the physical or logical backup method to Manually back up ApsaraDB for MongoDB data.
Restore data	If you want to restore data, you must Create an instance based on a time point.

2 ApsaraDB for MongoDB console

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

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

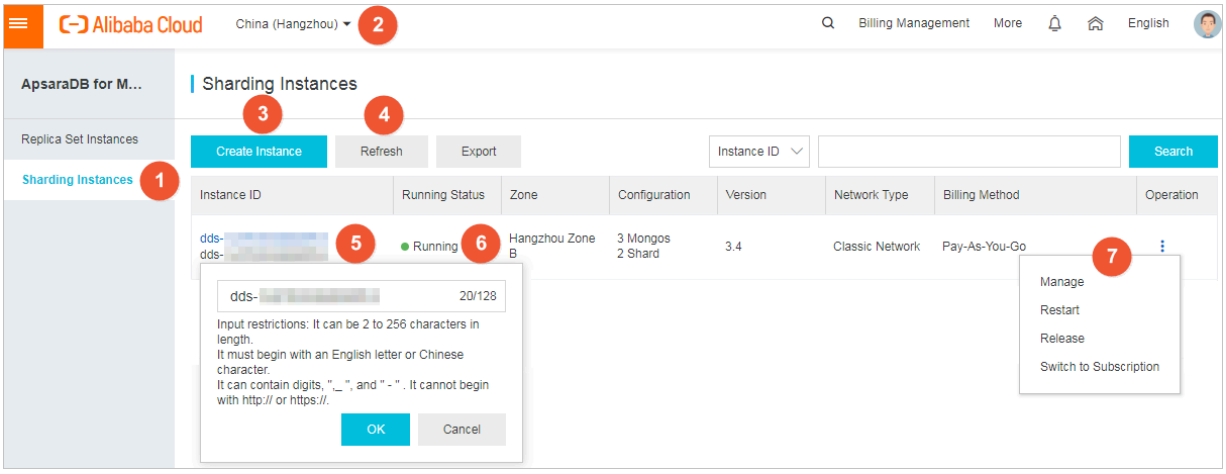
Prerequisites

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

Homepage

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

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



UI element description

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

MongoDB instance console

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

Page name	Area	Description	Link of common operation
Operation area in the window	-	You can back up and restart instances.	· Back up an instance · Restart an instance
Basic Information	Basic Information	You can view the basic information of an instance, such as its ID, region, network type, and storage engine.	-

Page name	Area	Description	Link of common operation
	Specifications	You can view instance specifications such as the database version, maintenance period, billing method, creation time, and expiration time.	Specify a maintenance period
	Mongos Nodes or Shard Nodes	· In the mongos list, you can locate a mongos ID and click the icon to change its configurations, and log on to or restart the mongos. · In the shard list, you can locate a shard ID and click the icon to log on to a shard, restart the shard, trigger a failover, or change the shard configurations. · Database read and write operations may fail when you restart nodes. We recommend that you do not perform the Create, Retrieve, Update, and Delete (CRUD) operations on databases when you restart nodes.	· Trigger a failover · Change configurations · Log on to a database · Restart a node · View monitoring information of mongos or shards
Backup and Restore	-	You can view and download a list of data backups for a specified time period, restore data from the specified time period, or create an instance from a specified backup point.	· Download backup data. · Create an instance at a specified time
Monitoring information	-	You can view monitoring information of mongos or shards for a specified time period based on specified metrics.	-
Security Control	Whitelist Settings	You can configure an IP whitelist.	Configure a whitelist.

Page name	Area	Description	Link of common operation
	Audit Log	MongoDB audit logs record all operations that your perform on a database. You can use these logs in analysis.	View audit logs.

3 Get started with ApsaraDB for MongoDB

Purpose

This topic describes how to create an ApsaraDB for MongoDB instance, specify its basic settings, and connect to the instance.This document can help you understand the basics of purchasing and using an ApsaraDB for MongoDB instance.

Quick start flowchart

If you use ApsaraDB for MongoDB for the first time, you can start with Before you start and ApsaraDB for MongoDB console.

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

4 Create a sharded cluster instance

You can use the ApsaraDB for MongoDB console to create a MongoDB sharded cluster instance. For more information about billing methods of MongoDB sharded cluster instances, see Billing items and pricing. This topic describes how to create an instance in the ApsaraDB for MongoDB console.

Prerequisites

- You have registered an Alibaba Cloud account. For more information about the registration process, see Register an Alibaba Cloud account.
- If you want to create a Pay-As-You-Go instance, make sure that your Alibaba Cloud account has sufficient balance.

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. Select Subscription (Sharding) or Pay-As-You-Go (Sharding).

   Note:
   For more information about billing methods, see Billing items and pricing.

5. Configure the parameters of the instance.

   For more information about the parameters, see Table 4-1: Instance parameters.

   Table 4-1: Instance parameters

Area	Parameter	Description
Basic Configuration	Region	The region to which the instance belongs. After the instance is created, you cannot modify its region. Use caution when selecting the region. Instances in the same region (such as an ECS instance and a MongoDB instance) are able to access to each other over the internal network.

Area	Parameter	Description
	Zone	The physical zones within a region that each have an independent power supply and separate network. For more information about regions and zones, see Regions and zones. An ECS instance and a MongoDB instance that reside in different zones within the same region can still access each other over the internal network. For more information, see Connect to a MongoDB instance in a different zone over the internal network. ECS instances and MongoDB instances that reside in the same zone are able to access each other with minimal network latency.
	Database Version	MongoDB sharded cluster instances support database versions 3.2, 3.4, and 4.0 We recommend that you select MongoDB 3.2 or later. Compared with MongoDB 3.2, MongoDB 3.4 provides higher performance and security. For more information, see Database versions and storage engines. Note: You can manually upgrade the database version to 3.4 or 4.0 when the instance is running. However, databases cannot be downgraded to previous versions. For more information, see Upgrade database versions.

Area	Parameter	Description
	Storage Engine	MongoDB sharded cluster instances support WiredTiger. For more information about the relationship between database versions and storage engines, see Database versions and storage engines.
Network Type	Classic Network	Cloud services in a classic network are not isolated . You can configure security groups or whitelist policies to block unauthorized access to classic network cloud services.
	VPC	Virtual Private Cloud (VPC) is an isolated network environment with higher security and performance than the classic network. You need to create VPCs in advance. For more information, see Configure VPC for a new instance. Note: · You can also modify the network type after creating an instance. For more information, see Modify the instance network type. · To smoothly migrate applications onto the cloud, you can use a leased line or VPN to integrate your own data center with resources on the cloud to make a virtual data center. For more information about the solution, see Migrate data from the classic network to VPC.
Mongos Specifications	Mongos Type	The specifications of mongos in sharded cluster instances. For more information about mongos specifications, see Instance types. When the instance is running, you can add mongos and upgrade or downgrade their configurations. After you modify instance specifications, your charges may change. For more information, see Billing items and pricing.

Area	Parameter	Description
	Quantity	The number of mongos. A MongoDB sharded cluster instance can contain 2 to 32 mongos.
Shard Specifications	Shard Type	The specifications of the shard in the instance. For more information about the shard specifications, see Instance types. When the instance is running, you can add shards and upgraded or downgraded their configurations. After you modify instance configurations, your charges may change. For more information, see Billing items and pricing.
	Storage Capacity	The storage space of the shard in the instance. A shard can provide a storage space of 10 to 1,000 GB. The storage space of a shard includes the space for data files, system files, and log files.
	Quantity	The number of shards. A MongoDB sharded cluster instance can contain 2 to 32 shards.
Configserver Specifications	Configserver Type	The config server specifications are fixed at 1 core 2 GB CPU and memory with 20 GB of storage space. These specifications cannot be modified.

Area	Parameter	Description
Set Password	-	The account password used to connect to the MongoDB database for the first time. · It must contain three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include !@#$%^&*()_+-= · The password must be 8 to 32 characters in length. You can specify a password when creating an instance, or specify a password or reset it when the instance is running.

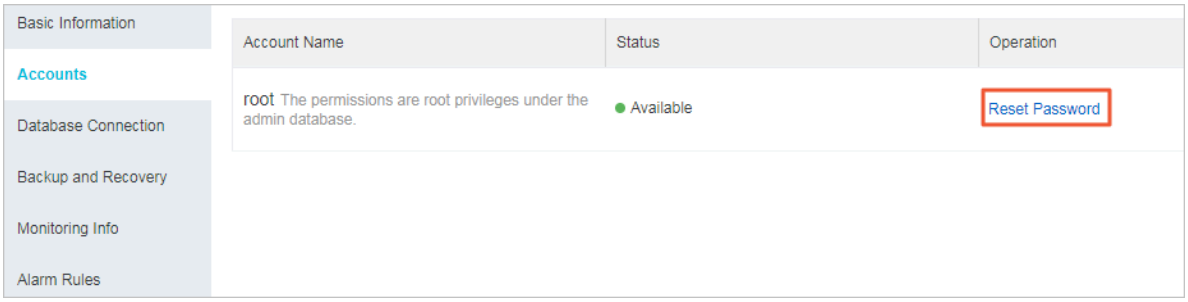
6. After you configure the parameters, click Buy Now.
7. On the Confirm Order page, select ApsaraDB for MongoDB Agreement of Service and complete the payment process as prompted.

5 Set a password

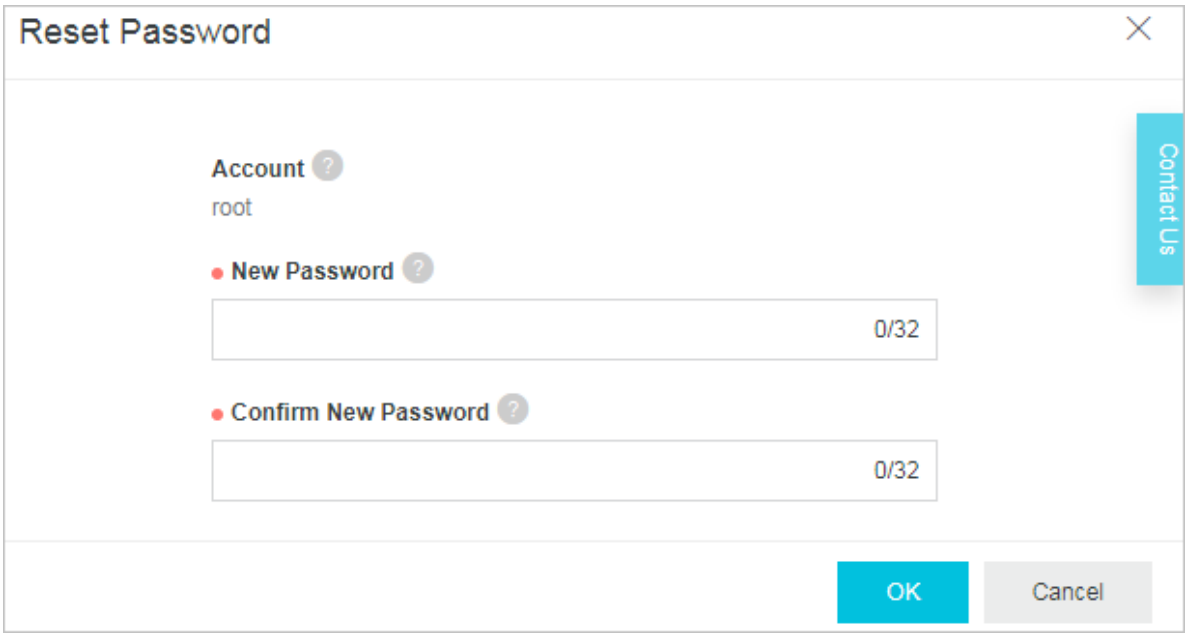
If you did not set a password when you created an instance, or have forgotten the password, you can reset the password of the instance.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the 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 and confirm the new password. Click OK.



Note:

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

6 Configure a whitelist

After a MongoDB instance is created, you must configure a whitelist to allow external devices to access the instance. The default whitelist contains only the default IP address 127.0.0.1, indicating that no device is able to access the MongoDB instance. This topic describes how to configure a whitelist in the console.

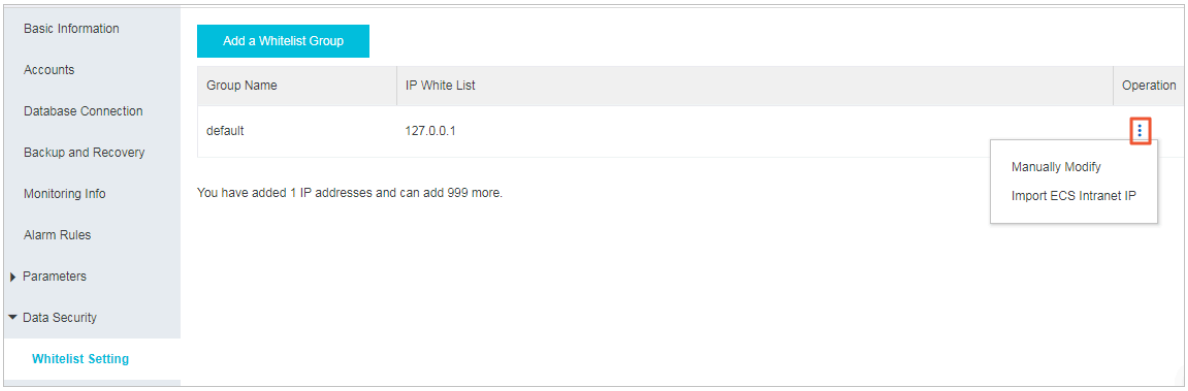
Precautions

- Before using an instance for the first time, you must modify its whitelist. After the whitelist is configured, the network addresses of the instance are displayed on the Basic Information and Database Connection pages.
- Proper configuration of the whitelist can enhance access security protection for ApsaraDB for MongoDB. We recommend that you regularly maintain the whitelist.

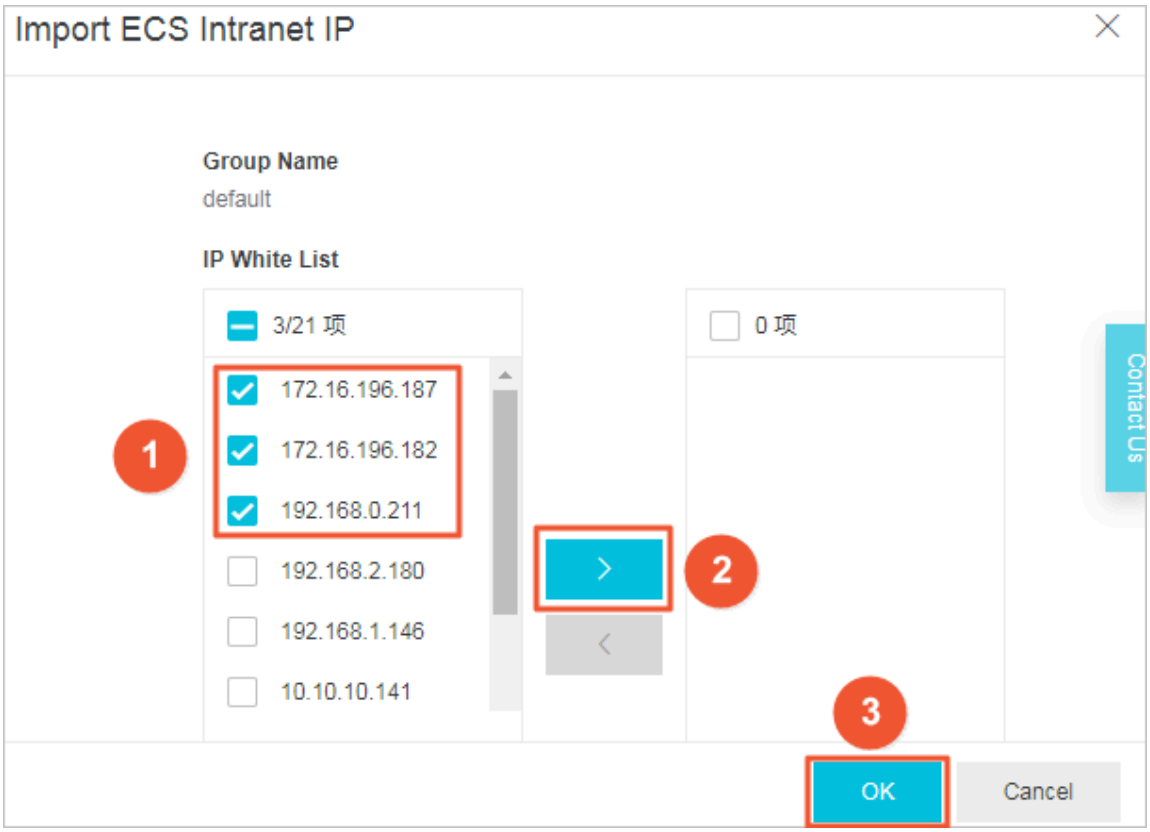
Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the instance ID.
5. In the left-side navigation pane, choose Security Control > Whitelist Settings.

6. Select Manually Modify or Import ECS Intranet IP.



- Click Manually Modify, enter an IP address or CIDR block, and clickOK.
- Click Import ECS Intranet IP. All internal IP addresses of ECS instances that belong to your account are displayed. You can select some ECS internal IP addresses, add them to the whitelist, and click OK.



Note:

- Separate multiple addresses with commas (,). No duplicate addresses are allowed. A maximum of 1,000 addresses can be added. Format: 0.0.0.0/0, 10.23.12.24 (IP address), or 10.23.12.24/24 (CIDR block. /24 represents the length of the prefix in the IP address. The prefix length can range from 1 to 32).

- 0.0.0.0/0 or null represents no IP address is blocked. This configuration incurs a high security risk to the database. We recommend that you add only the public IP addresses or CIDR blocks of your own Web servers to the whitelist.

Delete a whitelist

You can delete any whitelist group other than the default whitelist group.

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the instance ID.
5. In the left-side navigation pane, choose Security Control > Whitelist Settings.
6. Locate the whitelist group and choose > Delete Whitelist Group in the Actions column.
7. In the Delete Whitelist Group message that appears, click OK.

7 Apply for a public IP address

ApsaraDB for MongoDB allows you to apply for a public IP address to connect to an instance over the Internet. You can use the console to apply for a public IP address. Allowing access through a public IP address may incur security risks to the associated instance. To ensure data security, you must release public IP addresses in a timely manner when they are not needed.

Address types

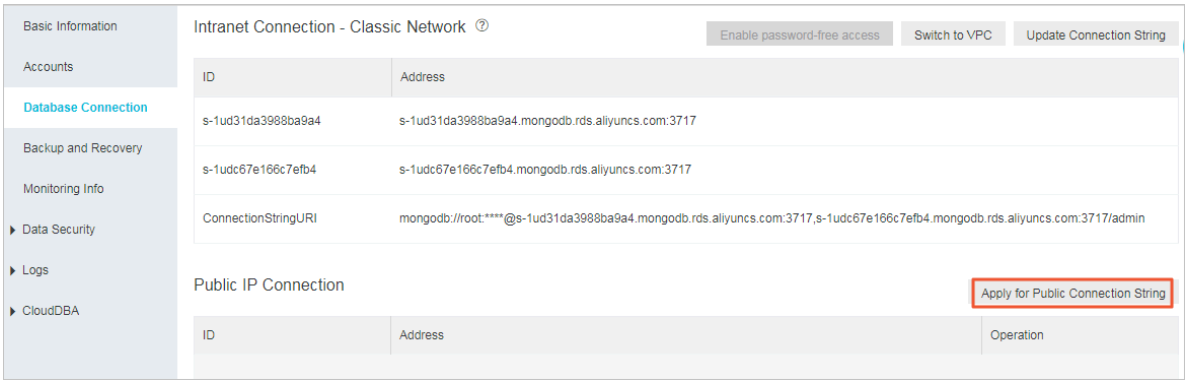
You can connect to ApsaraDB for MongoDB internally through a VPC or classic network, or externally over the Internet.

Address type	Description
Internal IP address – VPC	· Virtual Private Cloud (VPC) is an isolated network environment with higher security and performance than the classic network. VPCs must be created in advance. For more information, see Configure VPC for a new instance. · If your applications are deployed in an ECS instance, and the ECS instance is in the same region and has the same network type as the MongoDB instance, the two instances can access to each other through the internal network. · Accessing an instance through the VPC is more secure and offers better performance.
Internal IP address – classic	Cloud services in a classic network are not isolated. Unauthorized access to cloud services is blocked only by the security group or whitelist policy. For more information about modifying the network type to VPC, see Modify the instance network type.

Address type	Description
Public IP address	· You must apply for and release public IP addresses manually. · If you cannot access a MongoDB instance through the internal network, you must apply for a public IP address. Some scenarios include: - You want to access a MongoDB instance from an ECS instance, but they are in different regions or have different network types. - You want to access a MongoDB instance from a device that is not a public offering on Alibaba Cloud. Note: Access to a MongoDB instance through a public IP address may incur security risks. We recommend that you migrate your applications to an ECS instance that is in the same region and has the same network type as the MongoDB instance. Then you can use the VPC internal network.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the instance ID.
5. On the Basic Information page that appears, click Database Connection in the left-side navigation pane.
6. On the Database Connection page that appears, click Apply for Public Connection String on the right of Public IP Connection.



7. In the Apply for Public Connection String dialog box that appears, select Mongos and click OK.

    Note:

   You can repeat this step to apply for public IP addresses for multiple mongos based on your business needs. You can only apply for another public IP address after the current requested public IP address has been created.

   After you obtain the public IP address, you must add it to the whitelist before you can use this address to access the instance. For more information, see Configure a whitelist.

8. When the application is completed, the public IP address and its associated connection string URI are added to the mongos in the instance details page. For more information, see Connect to a MongoDB sharded cluster instance.

8 Connect to an instance

8.1 Connect to an ApsaraDB for MongoDB instance

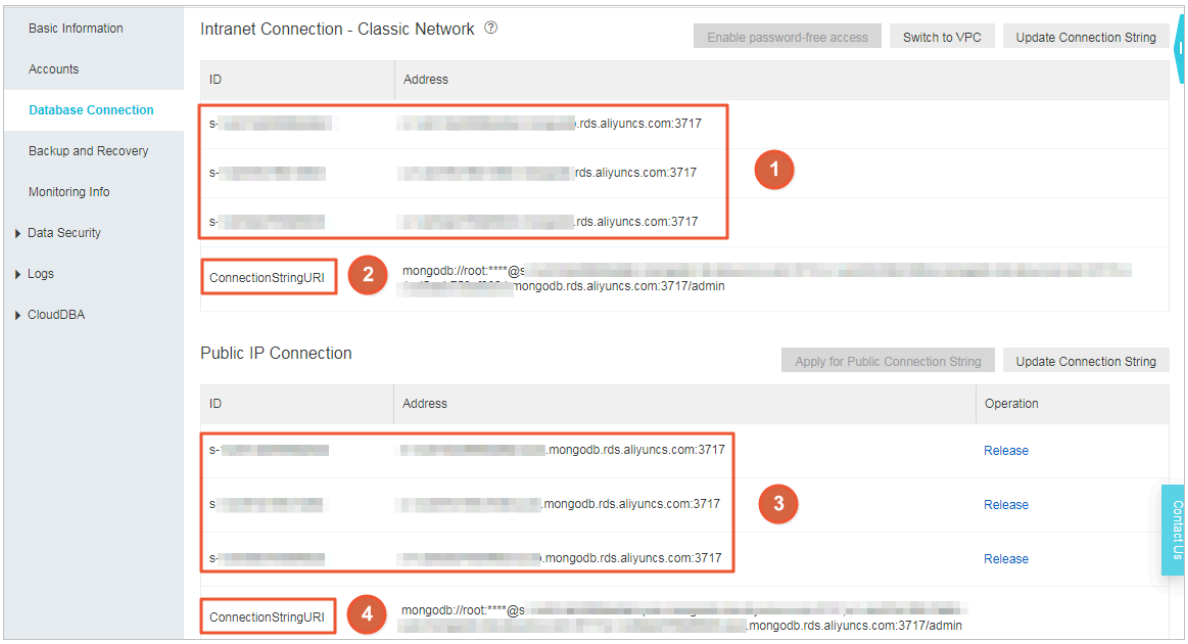
MongoDB sharded cluster instances provide individual addresses to connect to mongos as well as high-availability connection string URIs to connect to applications. This topic describes how to obtain and connect to these two address types.

Obtain the instance address

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the instance ID.
5. On the Basic Information page that appears, click Database Connection in the left-side navigation pane to view the addresses of the instance.

   You can see the internal and public IP addresses of all mongos in the instance. (Domain Information is addresses).

   As shown in the following figure, the instance has three mongos. Each mongos has a different address but uses the same port, 3717. You can log on to the instance through any mongos.



Connection information description

Parameter	Description
Network Type	· Intranet Connection – Classic Network: Cloud services in a classic network are not isolated. Security groups or whitelist policies can be used to block unauthorized access to such services. · Intranet Connection – VPC: Virtual Private Cloud (VPC) is an isolated network environment with higher security and performance than the classic network. You need to create VPCs in advance. For more information, see Configure VPC for a new instance. · Public IP Connection: Instances are not configured with a public IP address by default to ensure security. You must apply for a public IP address for the instance manually. For more information, see Apply for a public IP address.
mongos ID	The address of a mongos that you obtain from the console is in the following format: `< host >:< port >` · <host>: the domain address used to connect to the instance. · <port>: the port used to connect to the instance. Note: During routine tests, you can directly connect to any mongos. Take note that the client cannot provide load balancing and failover when you are only connected to a single mongos.

Parameter	Description
Connection string URI	The connection string URI you obtain from the console is in the following format: `mongodb ://[ username : password @] host1 [: port1 ][, host2 [: port2 ],...[, hostN [: portN ]]][/[ database ][? options ]]` · mongodb://: the prefix, representing a connection string URI. · username:password@: the username and password used to connect to the instance. Separate them with a colon (:). · hostX:portX: the address of a mongos in the instance. · /database: the name of the authentication database for the instance. · ? options: additional connection options. Note: We recommend that in the production environment you use connection string URIs to connect ApsaraDB for MongoDB instances from applications. The client can provide load balancing by automatically distributing requests to multiple mongos. When a mongos fails, the client can automatically fail over to another mongos in the normal state.

Connect to an ApsaraDB for MongoDB instance

1. In addition to the preceding Database Connection information, you also need to obtain the following information.
   · The username used to connect to the instance.

   Note:
   We recommend that you do not log on to an ApsaraDB for MongoDB instance in the production environment as the root user. You can log on to the instance

as the root user, and then create users and grant permissions. For more information, see db.createUser().

- The password used to connect to the instance. If you forget the password of the root account, you can reset the password by using the method specified in Set a password.
- Log on to the authentication database with the corresponding authentication information of the instance. If you log on to the instance as the root user, use the admin information for authentication.

2. Log on to the ApsaraDB for MongoDB instance.
   - Connect to an ApsaraDB for MongoDB instance through the mongo shell
   - Connect to an ApsaraDB for MongoDB instance through the program code

8.2 Connect to an ApsaraDB for MongoDB instance through the mongo shell

You can install the Mongo shell in an ECS instance and use the mongo shell to connect to an ApsaraDB for MongoDB instance.

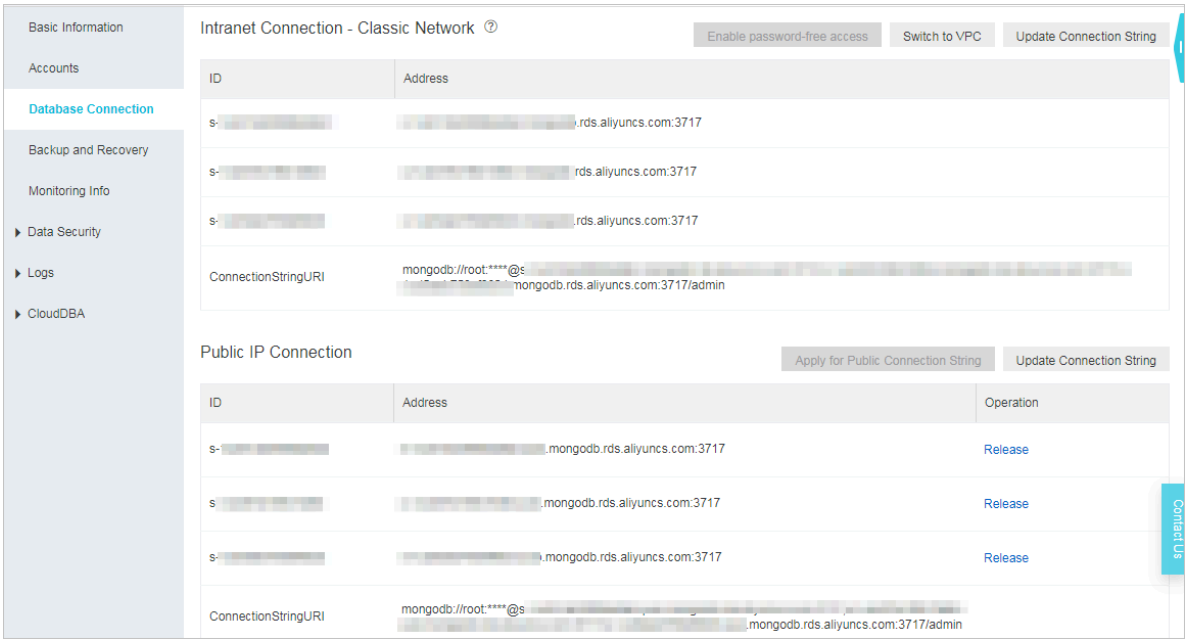
Prerequisites

- For successful authentication, you must use mongo shell 3.0 or later to connect to an ApsaraDB for MongoDB instance. For more information about the installation procedure, see Install MongoDB.
- Add the server IP addresses that need access to the instance to the whitelist in advance. For more information, see Configure a whitelist.
- To log on to ApsaraDB for MongoDB over the Internet, you must apply for a public IP address. For more information, see Apply for a public IP address.

Procedure

1. Log on to the ApsaraDB for MongoDB console.
2. In the upper-left corner of the homepage, select the region of the instance.
3. In the left-side navigation pane, click Sharding instances.
4. Locate the target instance, and then click the instance ID.

5. On the Basic Information page that appears, click Database Connection in the left-side navigation pane to obtain the address of the mongos.



Three mongos are displayed in this example. You can use one of their addresses to log on to the mongos.

6. Connect to a mongos from a local server or an ECS instance with mongo shell installed.

```
mongo -- host < mongos_hos t > - u < username > - p --
authentica tionDataba se < database >
```

Note:

- <mongos_host>: the address of any mongos in the ApsaraDB for MongoDB instance.
- <username>: the account for the instance. The default username is root.

- <database>: the name of the authentication database for the local MongoDB database. The default database name is admin.

Example:

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

7. When `Enter password :` is displayed, enter the password. If you forget the password of the root account, you can reset the password by using the method specified in Set a password.

> Note:
> The characters entered into the password field are not displayed.

More information

We recommend that you do not log on to an ApsaraDB for MongoDB instance as the root user in the production environment. You can create users and grant permissions based on your business needs. For more information, see db.createUser().

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

Related links

- MongoDB Drivers
- Connection String URI Format

> Note:
> The connection sample code in this topic applies when you use internal IP addresses provided by Alibaba Cloud to connect to ApsaraDB for MongoDB.

- For more information about how to obtain connection strings of ApsaraDB for MongoDB, see Connect to an ApsaraDB for MongoDB instance.

Node.js

Related links: MongoDB Node.js Driver

1. Initialize a project.

```
mkdir node - mongodb - demo
cd node - mongodb - demo
```

```
    npm   init
```

2. **Install the driver package and toolkit.**

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

3. **Obtain connection strings of ApsaraDB for MongoDB instances.**
4. **Use the following Node.js sample code.**

```
   ' use   strict ';
    var   uuid  =  require (' node - uuid ');
    var   sprintf  =  require (" sprintf - js "). sprintf ;
    var   mongoClien  t  =  require (' mongodb '). MongoClien  t ;
    var   host1  = " demotest - 1 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com ";
    var   port1  =  27017 ;
    var   host2  = " demotest - 2 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com ";
    var   port2  =  27017 ;
    var   username  = " demouser ";
    var   password  = " 123456 ";
    var   replSetNam  e  = " mgset - 1441984991 ";
    var   demoDb  = " test ";
    var   demoColl  = " testColl ";
   //  The   officially   recommende  d   solution .
    var   url  =  sprintf (" mongodb ://% s :% d ,% s :% d /% s ?
replicaSet =% s ",  host1 ,  port1 ,  host2 ,  port2 ,  demoDb ,
replSetNam  e );
    console . info (" url :",  url );
   //  Obtain   the   MongoClien  t .
    mongoClien  t . connect ( url ,  function ( err ,  db ) {
        if ( err ) {
            console . error (" connect   err :",  err );
            return   1 ;
       }
       //  Authentica  te .  Here ,  the   username   is   for
authentica  tion   of   the   admin   database .
        var   adminDb  =  db . admin ();
        adminDb . authentica  te ( username ,  password ,  function
( err ,  result ) {
            if ( err ) {
                console . error (" authentica  te   err :",  err );
                return   1 ;
           }
           //  Obtain   the   collection   handle .
            var   collection  =  db . collection ( demoColl );
            var   demoName  = " NODE :" +  uuid . v1 ();
            var   doc  = {" DEMO ":  demoName , " MESG ": " Hello
AliCoudDB   For   MongoDB "};
            console . info (" ready   insert   document : ",  doc
);
           //  Insert   data .
            collection . insertOne ( doc ,  function ( err ,  data
) {
                if ( err ) {
                    console . error (" insert   err :",  err );
                    return   1 ;
               }
                console . info (" insert   result :",  data ["
result "]);
               //  Read   data .
                var   filter  = {" DEMO ":  demoName };
```

```
                collection . find ( filter ). toArray ( function (
err ,  items ) {
                    if ( err ) {
                        console . error (" find   err :",  err );
                        return   1 ;
                    }
                    console . info (" find   document : ",  items
);
                    //  Close   the   client   and   release
resources .
                    db . close ();
                });
            });
        });
    });
```

PHP

Related links:

Mongodb php driver

1. **Install the driver package and toolkit.**

```
    $  pecl   install   mongodb
    $  echo  " extension = mongodb . so " >> ` php  -- ini  |  grep
" Loaded   Configurat  ion " |  sed  - e  " s |.*:\ s *||"`
    $  composer   require  " mongodb / mongodb =^ 1 . 0 . 0 "
```

2. **Obtain connection strings of ApsaraDB for MongoDB instances.**
3. **Use the following PHP sample code.**

```
    <?  php
     require  ' vendor / autoload . php '; //  include   Composer
goodies
        #  Instance   informatio  n
        $ demo_seed1  = ' demotest - 1 . mongodb . test . aliyun -
inc . com : 3717 ';
        $ demo_seed2  = ' demotest - 2 . mongodb . test . aliyun -
inc . com : 3717 ';
        $ demo_repln  ame  = " mgset - 1441984463 ";
        $ demo_user  = ' root ';
        $ demo_passw  ord  = ' 123456 ';
        $ demo_db  = ' admin ';
        #  Construct   the   mongodb   connection   string   based
on   the   instance   informatio  n .
        #  mongodb ://[ username : password @] host1 [: port1 ][,
host2 [: port2 ],...[, hostN [: portN ]]][/[ database ][?  options
]]
        $ demo_uri  = ' mongodb ://' . $ demo_user  . ':' . $
demo_passw  ord  . '@' .
          $ demo_seed1  . ',' . $ demo_seed2  . '/' . $ demo_db  .
'?  replicaSet =' . $ demo_repln  ame ;
        $ client  =  new   MongoDB \ Client ($ demo_uri );
        $ collection  = $ client -> testDb -> testColl ;
        $ result  = $ collection -> insertOne ( [ ' name ' => '
ApsaraDB   for   Mongodb ', ' desc ' => ' Hello ,  Mongodb '  ] );
         echo  " Inserted   with   Object   ID  '{$ result ->
getInserte  dId ()}'", "\ n ";
```

```
        $ result  = $ collection -> find ( [ ' name ' => ' ApsaraDB
for   Mongodb '] );
      foreach  ($ result   as  $ entry )
      {
         echo  $ entry -> _id , ': ', $ entry -> name , "\ n ";
      }
 ? >
```

Java

Related links:

- Official Quick Start
- JAR package download

1. Obtain connection strings of ApsaraDB for MongoDB instances.
2. Use the following Java sample code.
   - Maven configuration

```
    < dependenci  es >
        < dependency >
            < groupId > org . mongodb </ groupId >
            < artifactId > mongo - java - driver </ artifactId >
            < version > 3 . 0 . 4 </ version >
        </ dependency >
    </ dependenci  es >
```

   - Java sample code

```
     import   java . util . ArrayList ;
     import   java . util . List ;
     import   java . util . UUID ;
     import   org . bson . BsonDocume  nt ;
     import   org . bson . BsonString ;
     import   org . bson . Document ;
     import   com . mongodb . MongoClien  t ;
     import   com . mongodb . MongoClien  tOptions ;
     import   com . mongodb . MongoClien  tURI ;
     import   com . mongodb . MongoCrede  ntial ;
     import   com . mongodb . ServerAddr  ess ;
     import   com . mongodb . client . MongoColle  ction ;
     import   com . mongodb . client . MongoCurso  r ;
     import   com . mongodb . client . MongoDatab  ase ;
     public   class   Main  {
         public   static   ServerAddr  ess   seed1  =  new
ServerAddr  ess (" demotest - 1 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com ",  27017 );
         public   static   ServerAddr  ess   seed2  =  new
ServerAddr  ess (" demotest - 2 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com ",  27017 );
         public   static   String   username  = " demouser ";
         public   static   String   password  = " 123456 ";
         public   static   String   ReplSetNam  e  = " mgset -
1441984463 ";
         public   static   String   DEFAULT_DB  = " admin ";
         public   static   String   DEMO_DB  = " test ";
         public   static   String   DEMO_COLL  = " testColl ";
```

```
        public   static   MongoClien  t   createMong  oDBClient
() {
           //  Construct   a   seed   list .
            List < ServerAddr  ess >  seedList  =  new
ArrayList < ServerAddr  ess >();
            seedList . add ( seed1 );
            seedList . add ( seed2 );
           //  Construct   authentica  tion   informatio  n .
            List < MongoCrede  ntial >  credential  s  =  new
ArrayList < MongoCrede  ntial >();
            credential  s . add ( MongoCrede  ntial . createScra
mSha1Crede  ntial ( username ,
                    DEFAULT_DB ,  password . toCharArra  y ()));
           //  Construct   operation   options .  Configure
options   other   than   requiredRe  plicaSetNa  me   based   on
  your   actual   requiremen  ts .  The   default   parameter
settings   are   sufficient   for   most   scenarios .
            MongoClien  tOptions   options  =  MongoClien
tOptions . builder ()
                   . requiredRe  plicaSetNa  me ( ReplSetNam  e
). socketTime  out ( 2000 )
                   . connection  sPerHost ( 1 ). build ();
            return   new   MongoClien  t ( seedList ,
credential  s ,  options );
       }
        public   static   MongoClien  t   createMong  oDBClientW
ithURI () {
           //  Use   a   URI   to   initialize   the
MongoClien  t .
           // mongodb ://[ username : password @] host1 [: port1
][, host2 [: port2 ],...[, hostN [: portN ]]][/[ database ][?
options ]]
            MongoClien  tURI   connection  String  =  new
MongoClien  tURI (" mongodb ://" +  username  + ":" +  password
+ "@" +

seed1  + "," +  seed2  + "/" +

DEFAULT_DB  +

"?  replicaSet =" +  ReplSetNam  e );
            return   new   MongoClien  t ( connection  String );
       }
        public   static   void   main ( String   args []) {
            MongoClien  t   client  =  createMong  oDBClient ();
           // or
           // MongoClien  t   client  =  createMong  oDBClientW
ithURI ();
            try  {
               //  Obtain   the   collection   handle .
                MongoDatab  ase   database  =  client .
getDatabas  e ( DEMO_DB );
                MongoColle  ction < Document >  collection  =
database . getCollect  ion ( DEMO_COLL );
               //  Insert   data .
                Document   doc  =  new   Document ();
                String   demoname  = " JAVA :" +  UUID .
randomUUID ();
                doc . append (" DEMO ",  demoname );
                doc . append (" MESG ", " Hello   AliCoudDB   For
  MongoDB ");
                collection . insertOne ( doc );
                System . out . println (" insert   document : "
+  doc );
```

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

Python

Related links:

- Pymongo download
- Official documentation

1. **Install PyMongo.**

```
    pip   install   pymongo
```

2. **Obtain the connection strings of ApsaraDB for MongoDB instances.**
3. **Use the following Python sample code.**

```
    import   uuid
        from   pymongo   import   MongoClien  t
       #  Specify   two   addresses   used   to   connect   to
the   primary   and   secondary   nodes   of   the   instance .
        CONN_ADDR1  = ' demotest - 1 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com : 27017 '
        CONN_ADDR2  = ' demotest - 2 . mongodb . tbc3 . newtest .
rdstest . aliyun - inc . com : 27017 '
        REPLICAT_S  ET  = ' mgset - 1441984463 '
        username  = ' demouser '
        password  = ' 123456 '
       #  Obtain   the   MongoClien  t .
        client  =  MongoClien  t ([ CONN_ADDR1 ,  CONN_ADDR2 ],
replicaSet = REPLICAT_S  ET )
       #  Authentica  te .  Here ,  the   username   is   for
authentica  tion   of   the   admin   database .
        client . admin . authentica  te ( username ,  password )
       #  Use   the   collection : testColl   of   the   test
database   as   an   example .  Insert   doc   and   search   for
documents   based   on   the   demo   name .
        demo_name  = ' python -' +  str ( uuid . uuid1 ())
        print  ' demo_name :',  demo_name
        doc  =  dict ( DEMO = demo_name ,  MESG =" Hello   ApsaraDB
  For   MongoDB ")
        doc_id  =  client . test . testColl . insert ( doc )
```

```
        print  ' doc_id :',  doc_id
        for   d   in   client . test . testColl . find ( dict (
DEMO = demo_name )):
            print  ' find   documents :',  d
```

9 Migrate data

9.1 Use DTS to migrate data

When using DTS to migrate data, you cannot directly migrate data from the user-created database. However, you can migrate data from each shard in the user-created database to the cloud. With the incremental migration feature of DTS, you can migrate data to the cloud without stopping local application services.

Prerequisites

- Each shard of the user-created database has a public IP address or a port that is open to the public network.
- The user-created database supports database versions 3.0, 3.2, 3.4, and 3.6. MongoDB 4.0 is not supported.

  Note:
  For more information about data migration in MongoDB 4.0, see Use the built-in commands of MongoDB to migrate data.
- The storage space of the ApsaraDB for MongoDB instance should be larger than that of the user-created database.

Precautions

- Data in the admin database cannot be migrated, even if it is selected.
- The config database is an internal database. Do not migrate data from the config database unless otherwise required.
- ApsaraDB for MongoDB supports database versions 3.4 and 4.0.
- To avoid adverse impacts on your business, try to perform data migration during off-peak hours.

Billing for data migration

Migration type	Billing for link configuration	Billing for public traffic
Full data migration	Free	Free
Incremental data migration	Paid. For more information, see DTS pricing.	Free

Migration type description

- Full data migration: All existing data in the user-created database is migrated to the ApsaraDB for MongoDB instance. The following migration types are supported:
  - Database migration
  - Collection migration
  - Index migration
- Incremental data migration: Data that is updated in the user-created database is incrementally synchronized to the ApsaraDB for MongoDB instance. The following update operations are synchronized:
  - Create and delete operations on databases
  - Create, delete, and update operations on documents
  - Create and delete operations on collections
  - Create and delete operations on indexes

Permission requirements for data migration

Instance type	Full data migration	Incremental data migration
Local MongoDB instance	Read	· Read permission on the database to be migrated · Read permission on the admin database · Read permission on the user-created database
ApsaraDB for MongoDB instance	Read/write	Read/write

Preparations before migration

To avoid adverse impacts on your business, try to perform data migration during off-peak hours.

1. Disable the balancer of the user-created database. For more information, see Disable MongoDB sharded cluster balancer.

2. Clean up orphaned files in the user-created database to avoid block migration failure. For more information, see CleanupOrphaned.

   Example:

   For some shard, clean up all orphaned files from the user collection of the test database.

```
var nextKey = { };
var result ;

while ( nextKey != null ) {
  result = db . adminComma nd ( { cleanupOrp haned : " test .
user ", startingFr omKey : nextKey } );

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

  printjson ( result );

  nextKey = result . stoppedAtK ey ;
}
```

   Note:

   - You must perform this operation on each shard.
   - If any orphaned files remain, there may be ID conflicts between migrated files and the migration performance may be affected.

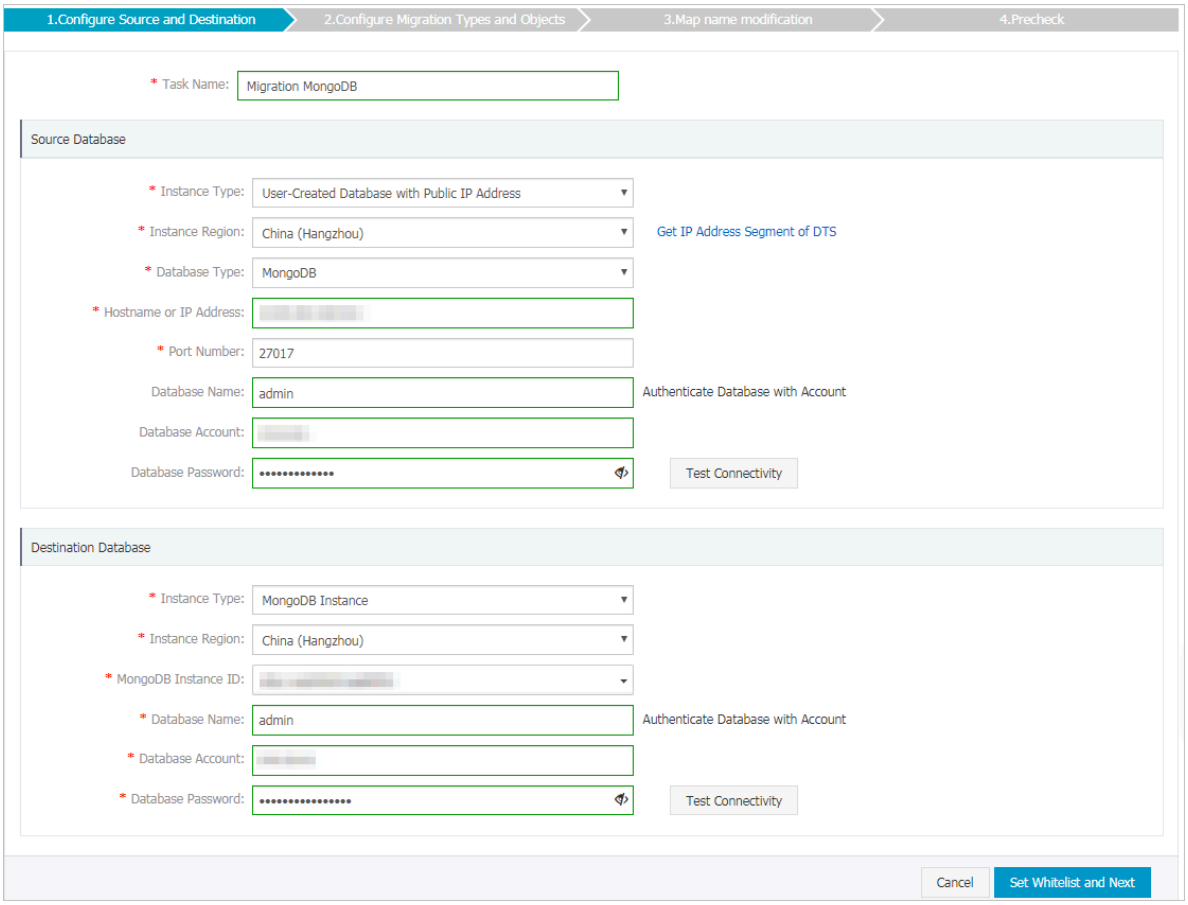
3. Configure data shards based on business needs. For more information, see Configure data shards (optional).

   Note:
   Before migration, you can create a database and collection for which to configure data shards. You can also configure data shards after migration.

Migration procedure

1. Log on to the Data Transmission Service console.
2. In the left-side navigation pane, click Data Migration.
3. In the upper-right corner of the Data Migration page, click Create Migration Task.

4. Configure the parameters of Source Database and Destination Database.



Parameters of source and destination databases	
Task Name	· DTS automatically generates a name for every task. Task names do not need to be unique. · You can modify task names as required. We recommend that you choose appropriate task names to easily identify tasks later.

Parameters of source and destination databases	
Source Database	· Instance Type: Select User-Created Database with Public IP Address. · Instance Region: Select the region where the source MongoDB database is located. · Database Type: Select MongoDB. · Hostname or IP Address: the domain name or IP address of a single shard. Note: DTS migrates data from every shard in the sharded cluster instance in sequence. Enter the domain name or IP address of the first shard here. When you create a second migration task later, enter the domain name or IP address of the second shard. Repeat this until all shards are migrated. · Database Name: the name of the authentication database for the source MongoDB database. · Database Account: the account used to log on to the source MongoDB database. For more information about permission requirements, see Permission requirements for data migration. · Database Password: the password used to log on to the source MongoDB database.
Destination Database	· Instance Type: Select MongoDB Instance. · Instance Region: Select the region where the destination MongoDB database is located. · MongoDB Instance ID: Select the ID of a sharded cluster instance . · Database Name: the name of the authentication database for the destination MongoDB database. The default database name is admin. · Database Account: the account used to log on to the destination MongoDB database. For more information about permission requirements, see Permission requirements for data migration. · Database Password: the password used to log on to the destinatio n MongoDB database.

Note:

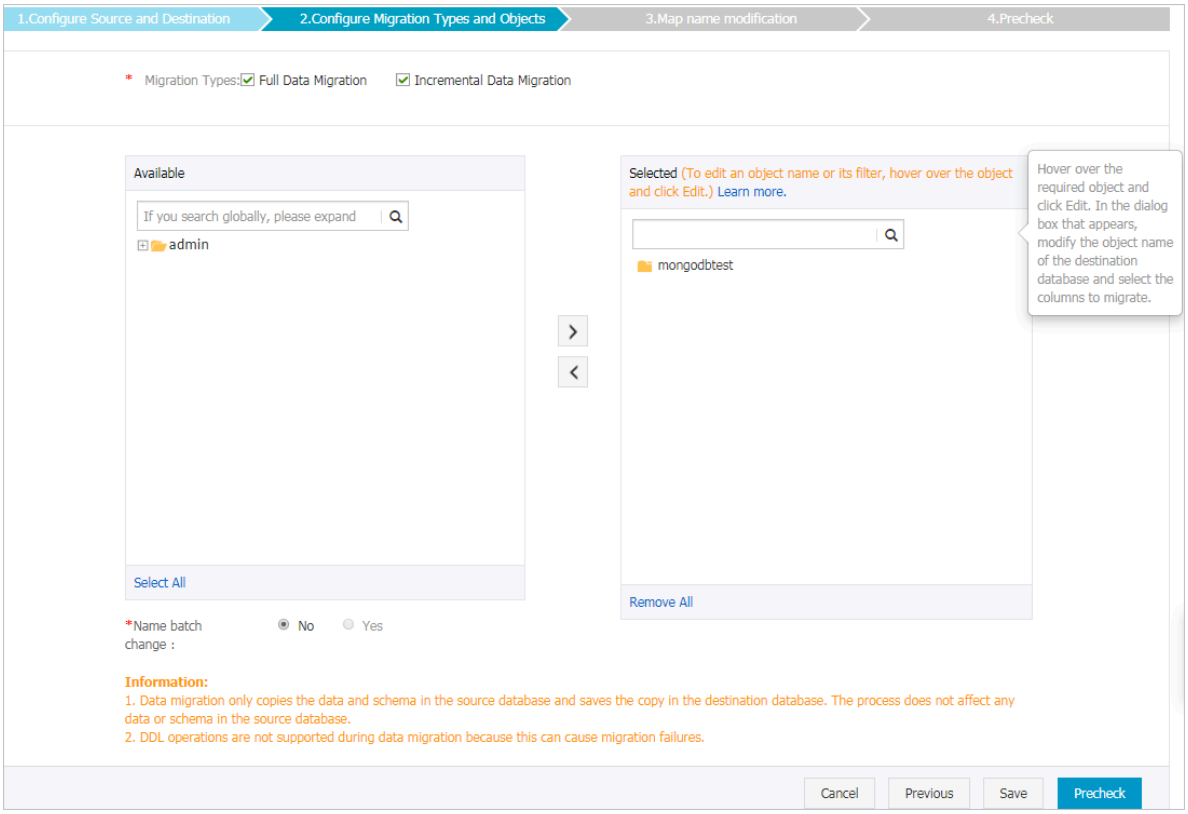
If the DTS server is not authorized to access the user-created database, you must grant access permission to the user-created database first. For more information,

click Get IP Address Segment of DTS on the Source Database and Destination Database page.

5. After you configure the parameters, click Set Whitelist and Next in the lower-right corner.

   You must add the public IP address of the DTS server to the whitelist of the destination MongoDB database to ensure the server can connect to the database. The public IP address of the DTS server can be removed from the whitelist after migration. For more information, see Configure a whitelist.

6. Select Migration Objects and Migration Types.



Configure the migration object and migration type	
Migration type	· If you want to migrate all data, select Full Data Migration. Note: To ensure data consistency, do not write new data to the source MongoDB database when a full migration is being performed. · If live migration is required, select both Full Data Migration and Incremental Data Migration.

Configure the migration object and migration type	
Migration object	· Select the database to be migrated from the Available list and click  to move it to the Selected list. Note: - Data in the admin database cannot be migrated, even if it is selected. - The config database is an internal database. Do not migrate data from the config database unless otherwise required. · The migration object can be a database or a collection. · The object name will remain the same after it is migrated to the destination MongoDB instance as it was in the source MongoDB instance by default. Note: - Each shard may store different data than other shards or contain only a portion of the whole database. This is determined by the characteristics of the shard. When you perform a data migration , you must select which data is to be migrated. - If the object you migrate has different names in the source and destination instances, you must use the object name mapping feature provided by DTS. For more information, see Object name mapping.

7. After you configure the preceding settings, click Pre-check.

Note:

· A pre-check is performed before a migration task is formally started. Migration can be started only after a successful pre-check.

· If the pre-check fails, click the icon after each check item to view the failure details. Perform the pre-check again after you have rectified the failed items.

8. After the pre-check is successful, click Next.
9. On the Confirm Purchase Configuration page, select Link Type and Data Transmission (Pay-As-You-Go) Service Terms.

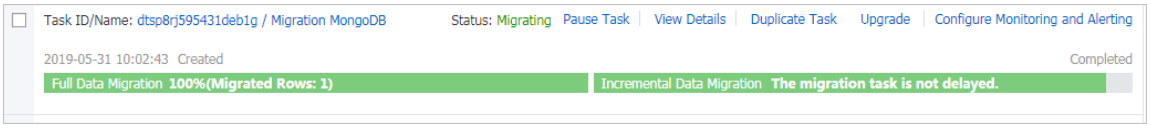
10. Click Buy and Start to start the migration task.

- Full data migration

  Wait until the migration task stops automatically.

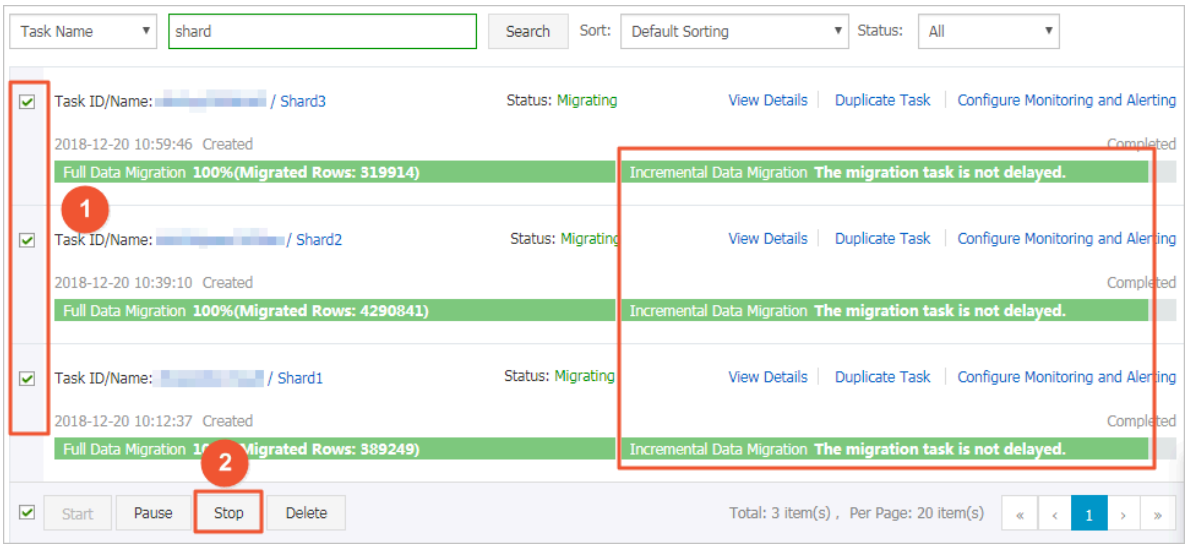
- Incremental data migration

  When the status of the migration task becomes Incremental migration without delay, it indicates that the data has been synchronized.



11. Repeat Steps 1 through 10 to create migration tasks for other shards and wait until all migration tasks are complete.

> **Note:**
>
> a. The incremental migration task does not end automatically. You can click Refresh to view the latest status of the migration task. When the migration tasks of all shards are Incremental migration without delay, stop the source database for several minutes.
>
> b. When the migration tasks of all shards become Incremental migration without delay again, stop the migration task.



Check whether the data is correct. If yes, you can switch your business from the user-created database to the ApsaraDB for MongoDB instance.

9.2 Use the built-in commands of MongoDB to migrate data

MongoDB provides the mongodump and mongorestore commands to migrate data from a user-created database to an ApsaraDB for MongoDB instance.

Precautions

- The operations in this topic apply to newly purchased instances that do not contain data.
- They are all full migration operations. To avoid inconsistencies in data, we recommend that you stop all write operations on the database before migration.
- Ensure that the mongodump and mongorestore versions are consistent with that of the user-created database.
- If you have used the mongodump command to back up data for any databases, back up the files in the dump folder to other directories. Ensure that the dump folder (which is the default backup folder) is empty. Otherwise, existing backup files are overwritten during the migration.
- Only run the mongodump and mongorestore commands on servers that MongoDB is installed on.

Back up the user-created database

This is a full migration operation. To avoid inconsistencies in data, stop the services related to the user-created database and stop all write operations on the database before migration.

1. On the server where the user-created database is deployed, run the following command to fully back up the data:

```
mongodump -- host < mongodb_ho st > -- port < port > - u <
username > -- authentica tionDataba se < database >
```

Note:

- <mongodb_host>: the server address of the user-created database. If this database is deployed on the current server, set this parameter to 127.0.0.1.
- <port>: the port number for the user-created database. It is 27017 by default.
- <username>: the username for the user-created database.
- <database>: the name of the authentication database for the user-created database. The default database name is admin.

Example:

```
mongodump -- host 127 . 0 . 0 . 1 -- port 27017 - u root
-- authentica tionDataba se admin
```

2. When `Enter password :` is displayed, enter the password to start data backup.

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

Configure data shards (optional)

If data shards are not configured, data is migrated to the primary shard. In this case, the storage space and computing performance of other shards are not used for data migration. For more information, see Configure data shards.

Note:
Before migration, you can create a database and collection for which to configure data shards. You can also configure data shards after migration.

Migrate data to the ApsaraDB for MongoDB instance

1. Obtain the public IP address of any mongos. For more information, see Connect to an ApsaraDB for MongoDB instance.

Note:
You must apply for a public IP address. For more information, see Apply for a public IP address.

2. On the server where the user-created database is deployed, run the following command to fully migrate the data to the ApsaraDB for MongoDB instance:

```
mongoresto re -- host < Mongos_hos t > - u < username > --
authentica tionDataba se < database > < Backup directory >
```

Note:

- <Mongos_host>: the address of any mongos in the ApsaraDB for MongoDB instance.
- <username>: the account for the ApsaraDB for MongoDB instance. The default username is root.
- <database>: the name of the authentication database for the ApsaraDB for MongoDB instance. The default database name is admin.
- <Backup directory>: the directory that stores backup files. The default backup directory is dump.

Example:

```
mongoresto re -- host s - bp **********- pub . mongodb . rds .
aliyuncs . com : 3717 - u root -- authentica tionDataba se
admin dump
```

3. When `Enter password :` is displayed, enter the password to start data migration.

Note:

If you have forgotten your password, see Set a password.

Wait until data migration is complete and check whether the data is correct. If yes, you can switch your business to the ApsaraDB for MongoDB instance.