Alibaba Cloud ApsaraDB for MongoDB

Quick Start for Replica Set

Issue: 20190425

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 easily migrate data from a user-created MongoDB instance to an ApsaraDB for MongoDB instance. Before using ApsaraDB for MongoDB, you need to pay attention to its constraints, as listed in the following table.

Operation	Constraints
Restart an instance	You must log on to the *ApsaraDB for MongoDB console* or call the RestartDBInstance operation to restart an instance.
Create a replica set	· A replica set automatically created by ApsaraDB for MongoDB consists of an operable primary node, a hidden secondary node (invisible to you), and the other operable secondary nodes. · You can select the number of nodes (such as three, five, or seven nodes) based on business requirements to increase or decrease secondary nodes on demand.
Migrate data	· We recommend that you use *Data Transmission Service (DTS)* to migrate data. · You can also use *the built-in commands of MongoDB* to migrate data.
Select the database version and storage engine	For more information, see *Versions and storage engines*.

2 Get started with ApsaraDB for MongoDB

Purpose

This document describes how to quickly create an ApsaraDB for MongoDB instance, specify its basic settings, and connect to the instance. It helps you understand the basic process from purchasing an ApsaraDB for MongoDB instance to using the instance.

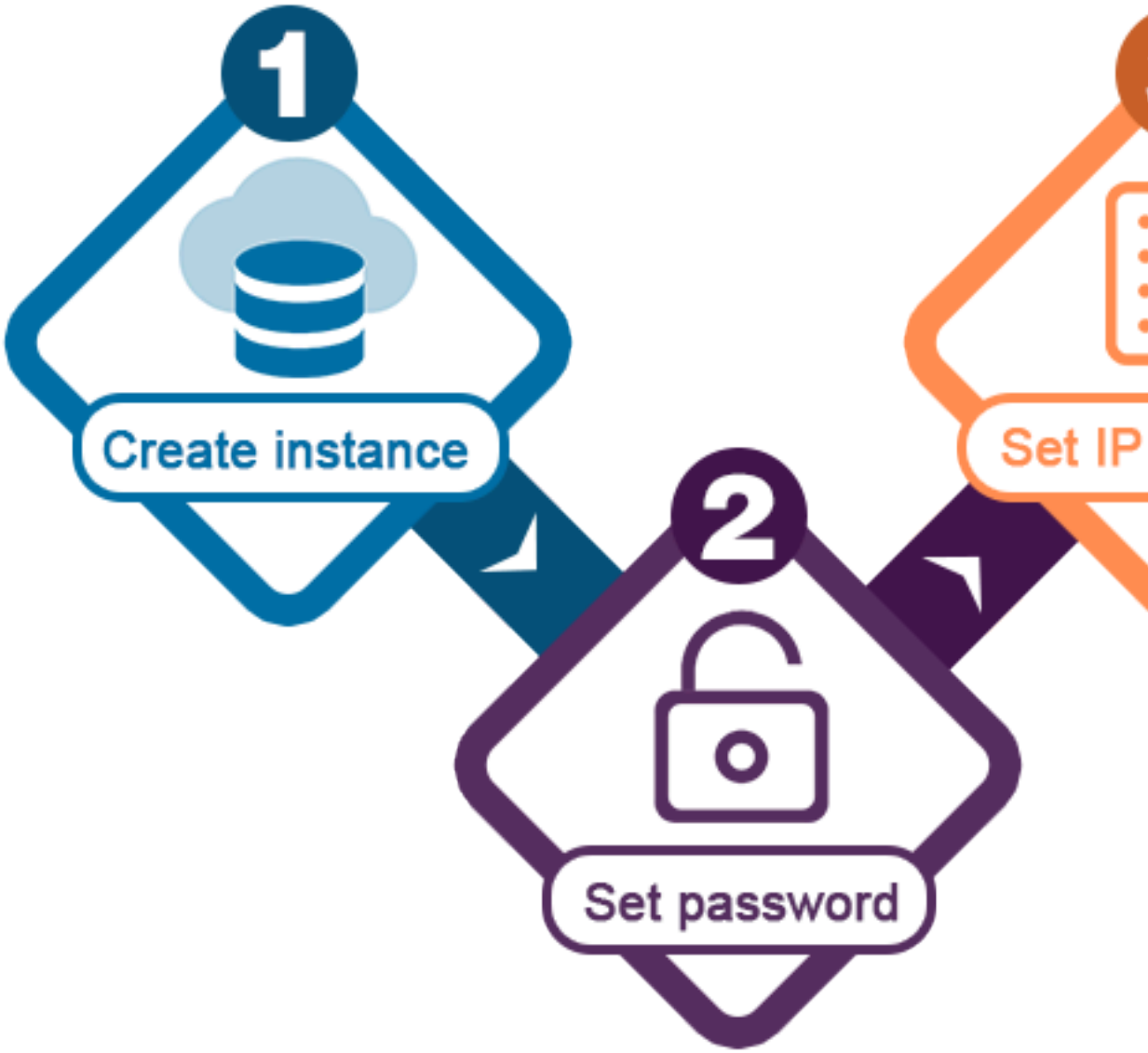
Intended audience

- Users who purchase an ApsaraDB for MongoDB instance for the first time.
- Users who need to specify basic settings after creating an ApsaraDB for MongoDB instance.
- Users who need to know how to connect to 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*.

The following figure shows the operations that you need to perform from purchasing an instance to using the instance.



3 Create an instance

You can log on to the Alibaba Cloud *ApsaraDB for MongoDB console* or call the CreateDBInstance operation to create an ApsaraDB for MongoDB instance. For more information about how to bill an instance, 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. If you do not have an Alibaba Cloud account, click *Register*.
- 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 left-side navigation pane, click Replica Set Instances. On the page that appears, click Create Instance to go to the instance creation page.
3. Select Subscription or Pay-As-You-Go. For more information about the billing methods, see *Billing items and pricing*.
4. Specify the instance configuration. The following table describes related parameters.

Area	Parameter	Description
Basic Configuration	Region	The geographical location of the instance. Instances in different regions cannot be interconnected through an intranet. Once an instance is created, you cannot change the region. Therefore, you need to select a region with caution. Instances in the same region (such as an *ECS* instance and an ApsaraDB for MongoDB instance) can be interconnected through an intranet.

Area	Parameter	Description
	Zone	The physical area with its power supply and network isolated from other counterparts in the same region. For more information about regions and zones, see *Regions and zones*. An ApsaraDB for MongoDB instance and an ECS instance in different zones of the same region can be interconnected through an intranet. For more information, see *Connect to an ApsaraDB for MongoDB instance through a cross-zone intranet*. The network latency is minimal when an ECS instance and an ApsaraDB for MongoDB instance in the same zone are interconnected through an intranet.
	Database Version	The database version of the instance. Currently, ApsaraDB for MongoDB supports MongoDB 3.2, MongoDB 3.4, and MongoDB 4.0. Note: · You can manually upgrade the database version of an instance from MongoDB 3.2 to MongoDB 3.4 when the instance is running, but cannot downgrade the upgraded version. · You cannot upgrade the database version of an existing instance from MongoDB 3.2 or MongoDB 3.4 to MongoDB 4.0. To use MongoDB 4.0, you need to select MongoDB 4.0 for Database Version when you create an instance. · The database version of an instance cannot be downgraded. You need to select a database version based on business requirements.
	Storage Engine	The storage engine of the instance. You can select WiredTiger, RocksDB, or TerarkDB. Note: Supported storage engines may vary in different database versions. For more information about database versions and storage engines, see *Versions and storage engines*.

Area	Parameter	Description
	Replication Factor	The number of nodes that you select for the replica set instance based on business requirements. For example , you can select more nodes for business scenarios that require more read operations than write operations.
Network Type	Classic Network	The classic network on which cloud services are not isolated. You can configure a security group or whitelist policy for them to block unauthorized access.
	VPC	The Virtual Private Cloud, which is recommended. A VPC is an isolated network environment that provides enhanced security and better performance than traditional classic networks. You need to create a VPC in advance. For more information about how to configure a VPC, see *Configure a VPC for a new instance*. Note: · You can also change the network type of an instance after creating the instance. For more information, see *Switch the network type of an instance*. · You can use a physical connection or Virtual Private Network (VPN) to combine your on-premises IDC with cloud resources in Alibaba Cloud VPC to create a virtual IDC, so that you can smoothly migrate your applications to the cloud. For more information, see *Configure a hybrid access solution to smoothly switch from a classic network to a VPC*.
Specifications	Specification	The CPU and memory occupied by the instance.
	Storage Space	The maximum number of connections and maximum input/output operations per second (IOPS) vary depending on specifications. The maximum IOPS indicates the maximum number of read or write operations separately . The maximum sum of read and write operations can be twice the maximum IOPS.

Area	Parameter	Description
Set Password	· Set now · Set after purchase	The password used to connect to ApsaraDB for MongoDB for the first time. · 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 (=). [DO NOT TRANSLATE] · The password must be 8–32 characters in length. You can set a password when creating the instance. You can also set a password or *reset the password* when the instance is running.
Purchase Quantity	Duration Quantity	· Subscription: Select the duration and quantity for the subscription-based instance to be purchased. You can select one to nine months for the subscription period on a monthly basis, or one to three years for the subscription period on a yearly basis. · Pay-As-You-Go: Select the quantity for the Pay-As-You-Go instance to be purchased with the same configuration. You can select an integer in the range of 1 to 10.

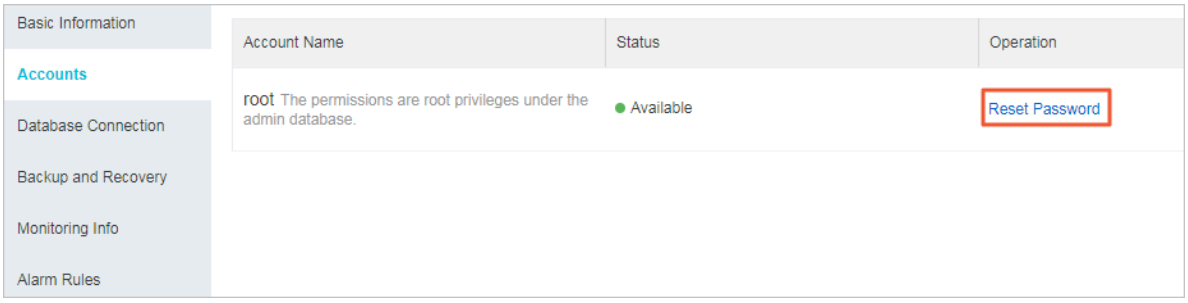
5. Click Buy Now to go to the Confirm Order page.
6. On the Confirm Order page that appears, read and select ApsaraDB for MongoDB Agreement of Service and follow the instructions to complete the payment process.

4 Set a password

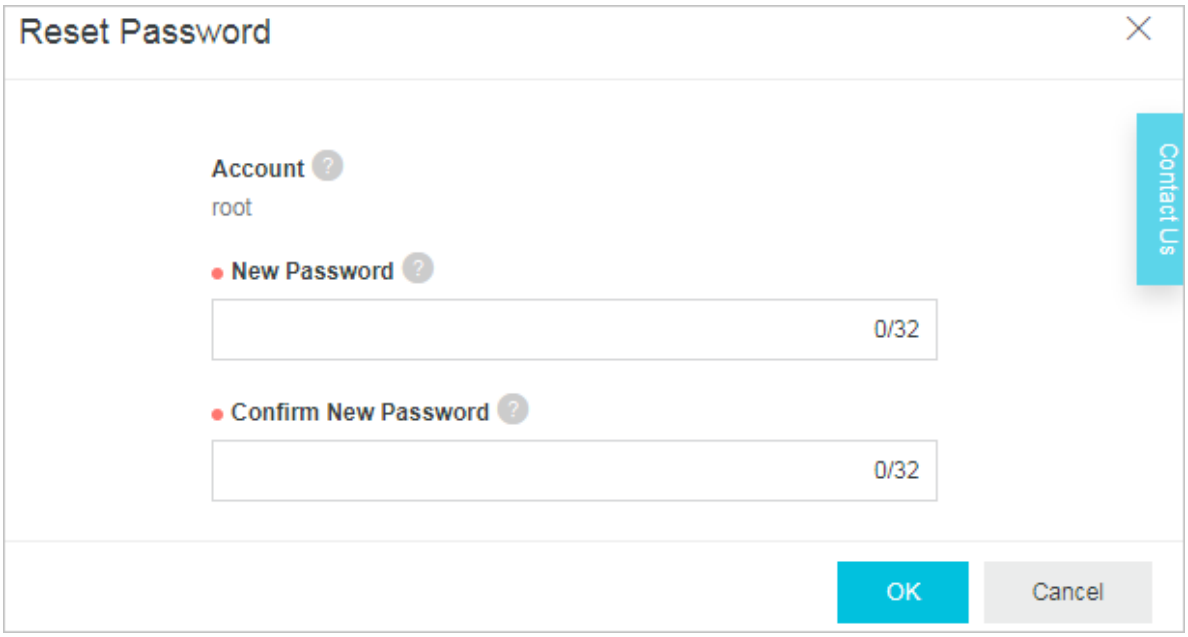
If you did not set a password when creating an instance or need to change the old password when using ApsaraDB for MongoDB, you can reset the password for the instance.

Procedure

1. Log on to the *ApsaraDB for MongoDB console*.
2. Click the target instance ID or choose > Manage in the Operation column corresponding to the target instance.
3. In the left-side navigation pane, click Accounts. On the page that appears, click Reset Password, as shown in the following figure.



4. In the Reset Password dialog box that appears, enter a new password, confirm your password, and click OK, as shown in the following figure.



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.

5 Configure a whitelist

After an ApsaraDB for MongoDB instance is created, it automatically adds the IP address 127.0.0.1 to a whitelist, indicating that all IP addresses and CIDR blocks are prohibited to access this instance. To guarantee database security and stability, you must delete 127.0.0.1 and add IP addresses or CIDR blocks that need to access your ApsaraDB for MongoDB instance to the whitelist. Otherwise, you cannot view the connection information about the instance on its Basic Information page. ApsaraDB for MongoDB allows you to add a maximum of 1,000 IP addresses to the whitelist.

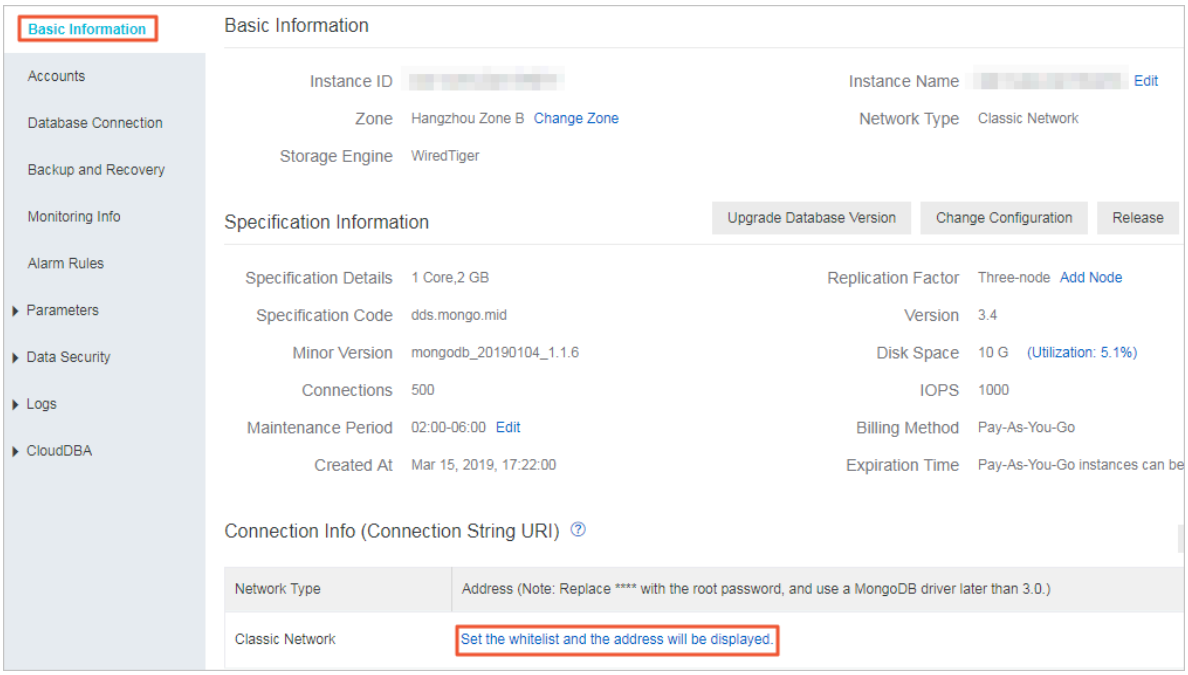
Context

Before using the target instance for the first time, you must configure its whitelist.

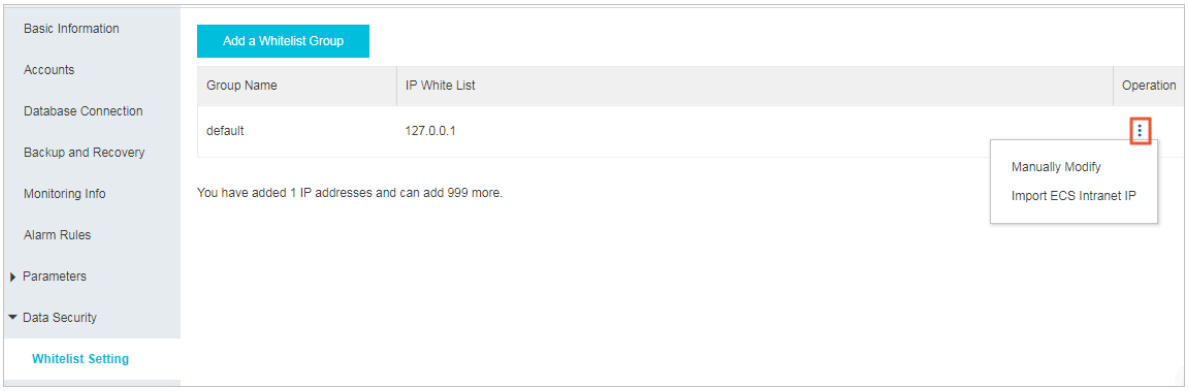
Procedure

1. Log on to the *ApsaraDB for MongoDB console*.
2. Click the target instance ID or choose > Manage in the Operation column corresponding to the target instance.

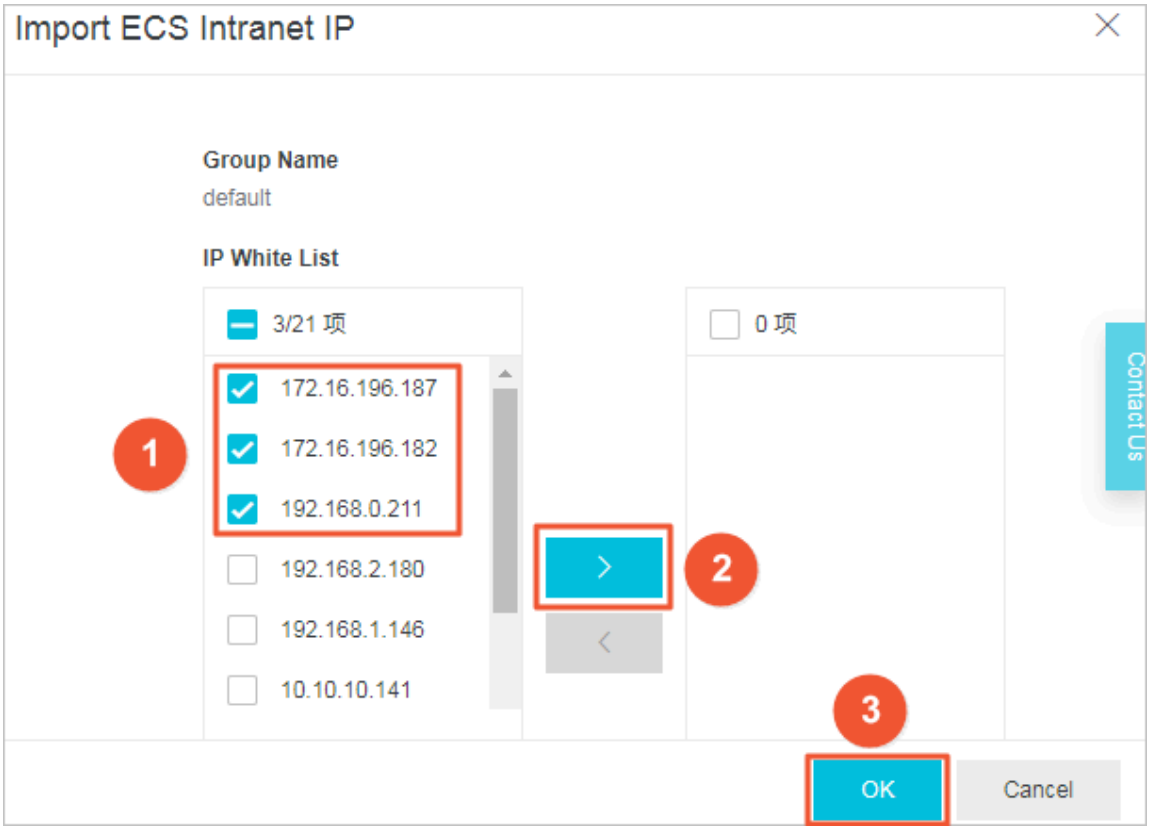
3. On the Basic Information page that appears, click Set the whitelist and the address will be displayed., as shown in the following figure.



Alternatively, in the left-side navigation pane, choose Data Security > Whitelist Setting, as shown in the following figure.



4. 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, as shown in the following figure.



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.

Result

After configuring the whitelist, you can view the VPC connection information about the instance on its Basic Information page.

Subsequent operations

- **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.**
- **If necessary, you can select Manually Modify or Import ECS Intranet IP to modify the whitelist.**

6 Apply for a public address

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

Context

There are certain security risks for connecting to an instance through the Internet. To guarantee data security, you need to release a public address in a timely manner after using it.

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. Click the target instance ID or choose > Manage in the Operation column corresponding to the target instance.
4. On the Basic Information page that appears, click Database Connection in the left-side navigation pane.
5. On the page that appears, click Apply for Public IP to the right of Public IP Connection.
6. In the Apply for Public IP dialog box that appears, click OK.

> Note:
> After the application procedure, you can use the obtained public address to access the target instance. Before that, you need to add the public IP address of the terminal that you use to connect to the instance to the whitelist. For more information, see *Configure a whitelist*.

Result

After the application procedure, the target ApsaraDB for MongoDB replica set instance provides separate addresses for you to connect to the primary and secondary nodes. For more information, see *Obtain the replica set instance connection information*.

7 Connect to an instance

7.1 Obtain the replica set instance connection information

An ApsaraDB for MongoDB replica set instance provides separate addresses for you to connect to the primary and secondary nodes, and a connection string URI for you to connect your applications to the instance in high availability mode.

Required information

Before connecting to an ApsaraDB for MongoDB instance, obtain the following information:

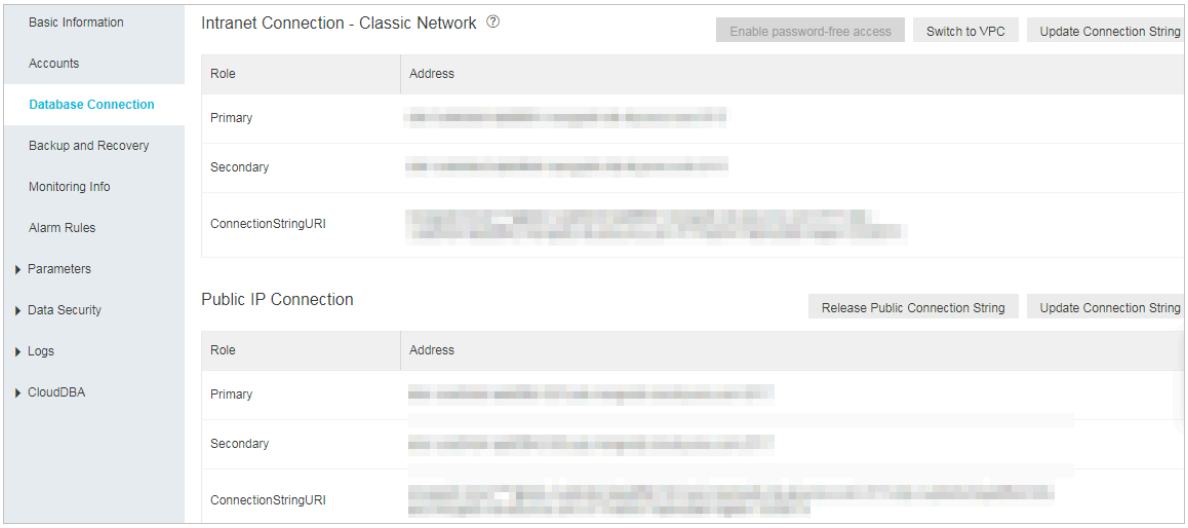
- Database name, which is admin by default
- Username of an instance account
- Password of the instance account
- Connection information
  - Domain names and port numbers of the primary and secondary nodes
  - Connection string URI for the high availability mode

Procedure

1. Log on to the *ApsaraDB for MongoDB console*.
2. Click the target instance ID or choose > More in the Operation column

   corresponding to the target instance.
3. On the Basic Information page that appears, you can view the connection string URI for connections in high availability mode and the connection information

of the primary and secondary nodes (including their domain names and port numbers), as shown in the following figure.



4. In the left-side navigation pane, click Accounts to view the username, which is root by default.

> **Note:**
> You can set a database logon password when creating an instance or reset the password after creating the instance. For more information, see *Set a password*.

Connection instructions

ApsaraDB for MongoDB provides two connection modes for a replica set instance as follows:

- Separate addresses for you to connect to the primary and secondary nodes of the instance.
  - If you connect to the primary node, you have read and write permissions.
  - If you connect to a secondary node, you have only the read permission.
  - During daily testing, you can use this connection mode to directly connect to the primary node or a secondary node.
  - We do not recommend that you use these separate addresses to directly connect your online applications to an instance. If you do so and a failover is triggered between the primary and secondary nodes, read and write operations may be affected in your applications.

- A connection string URI for you to connect to the instance to achieve load balancing.

  > Note:
  > We recommend that you use the connection string URI to connect your applications in the production environment to an instance. In this case, if a node is faulty and a failover is triggered between the primary and secondary nodes, the connection of your applications is not affected.

  The following example shows a connection string URI that you can obtain from the console.

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

  Notes:

  - mongodb://: The prefix, which indicates that this address is a connection string URI.
  - username:password@: The username and password used to log on to a database . If authentication is enabled, a password is required.
  - hostX:portX: The list of addresses used to connect to nodes in the replica set.
  - /database: The database corresponding to the username and password if authentication is enabled.
  - ?options: The additional connection options.
- All official MongoDB drivers support the connection to an instance by using a connection string URI. For more information, see *Connection sample code for MongoDB drivers*.

7.2 Connect to ApsaraDB for MongoDB through the mongo shell

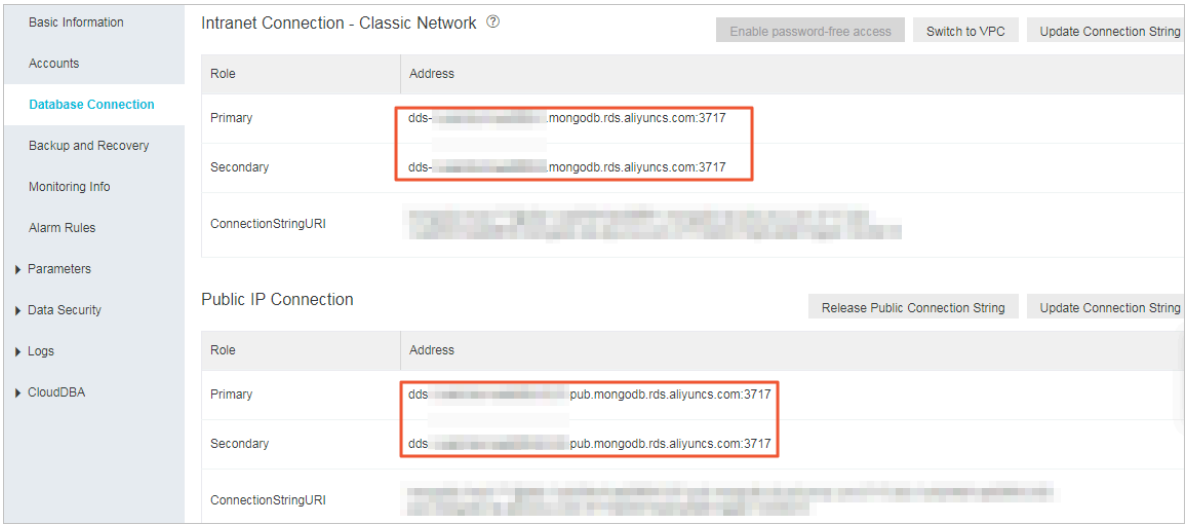
You can install the mongo shell on an ECS instance and use the mongo shell to connect to ApsaraDB for MongoDB.

Prerequisites

To ensure successful authentication, use mongo shell 3.0 or a later version to connect to ApsaraDB for MongoDB.

Procedure

1. Log on to the *ApsaraDB for MongoDB console*.
2. Click the target instance ID or choose > Manage in the Operation column to go to the Basic Information page of the target instance.
3. In the left-side navigation pane, click Database Connection.



   - Red boxes highlight the connection information.
   - In each connection address, the default port number is 3717 and the domain name is separated from the port number with a colon (:).
4. In the ECS instance, run a mongo command to connect to ApsaraDB for MongoDB. The following command is an example:

```
mongo -- host dds - xxxx . mongodb . rds . aliyuncs . com : 3717
 - u root - p Ft123456 -- authentica tionDataba se admin
```

Result

Mongo shell FAQs

- *Connection*
- *Connections*
- *High load*

7.3 Connection sample code for MongoDB drivers

Related links

- *MongoDB Drivers*

- *Connection String URI Format*

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

Node.js

Related link: *MongoDB Node.js Driver*

1. Initialize a project.

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

2. Install the driver package and tool kit.

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

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

   For more information, see *Obtain the replica set instance connection information*.

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 the username and password used to
  log on to ApsaraDB for MongoDB . The username in
this sample code is used to log on to the admin
  database .
    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 link:

MongoDB PHP Driver

1. **Install the driver package and tool kit.**

```
$  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 the information required to connect to an ApsaraDB for MongoDB instance.**

   For more information, see *Obtain the replica set instance connection information*.

3. **Use the following PHP sample code.**

```
<? php
 require  ' vendor / autoload . php '; //  include   Composer
 goodies
#  Specify   the   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  a  MongoDB  connection  string  URI  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 the information required to connect to an ApsaraDB for MongoDB instance.

   For more information, see *Obtain the replica set instance connection information*.

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 .  Default   parameter
settings   can   meet   requiremen  ts   in   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 information required to connect to an ApsaraDB for MongoDB instance.**

   For more information, see *Obtain the replica set instance connection information*.

3. Use the following Python sample code.

```
import uuid
from pymongo import MongoClien t
# Specify two addresses for connecting to the
primary and secondary nodes of the instance .
CONN_ADDR1 = ' demotest - 1 . mongodb . tbc3 . newtest . rdstest .
aliyun - inc . com : 27017 '
CONN_ADDR2 = ' demotest - 2 . mongodb . tbc3 . newtest . rdstest .
aliyun - inc . com : 27017 '
REPLICAT_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 the username and password used to
log on to ApsaraDB for MongoDB . The username in
this sample code is used to log on to the admin
  database .
client . admin . 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
```

8 Migrate data

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

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

Notes

- Operations in this topic apply to newly purchased instances that do not contain data.
- Use the mongodump and mongorestore commands of MongoDB 3.0 or a later version.
- If you have used the mongodump command to back up data for any databases, move backup files from the dump folder to other directories. Ensure that the dump folder (which is the default backup folder) is empty. If it is not empty, existing backup files are overwritten during the migration.
- Run the mongodump and mongorestore commands on the server where MongoDB is installed. Do not run these commands in a mongo shell environment.

Back up the user-created MongoDB instance

On the server where the user-created MongoDB instance is deployed, run the following command to export all database data and store the exported files in the dump folder of the current directory:

```
mongodump  -- host  < mongodb_ho  st >: 27017  -- authentica
tionDataba  se   admin  - u  < username > - p  < password >
```

Notes:

- <mongodb_host>: 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.
- <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.

Wait until data backup is completed. The data of the user-created MongoDB instance is backed up in the dump folder of the current directory.

Migrate data to the ApsaraDB for MongoDB instance

1. Obtain the address used to connect to the primary node of the ApsaraDB for MongoDB replica set instance. For more information, see *Obtain the replica set instance connection information*.

    Note:
   If you need to connect to your ApsaraDB for MongoDB instance through the Internet, you can *apply for a public address*.

2. On the server where the user-created MongoDB instance is deployed, run the following command to import the backup data of the user-created MongoDB instance into the target ApsaraDB for MongoDB instance:

   ```
   mongoresto re -- host < Primary_ho st >: 3717 -- authentica
   tionDataba se admin - u < username > - p < password > <
   Backup directory >
   ```

   Notes:

   - <Primary_host>: The address used to connect to the primary node of the ApsaraDB for MongoDB replica set instance.
   - <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.
   - <Backup directory>: The directory that stores backup files. The default directory is the dump folder.

   Wait until data recovery is completed. Data is migrated from the user-created MongoDB instance to the ApsaraDB for MongoDB replica set instance.