Alibaba Cloud

ApsaraDB for MongoDB
Product Usage

(-) Alibaba 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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice 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.
🔔 Warning	A warning notice 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 restart an instance.
🔊 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔊 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings> Network> Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK.**
Courier font	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Hot issues

1.1. ApsaraDB for MongoDB instance types

For more information, see Instance types.

1.2. Why am I unable to find the ApsaraDB for MongoDB instance that I created?

If you cannot find your instance in the ApsaraDB for MongoDB console, see the following table.

Possible cause	Troubleshooting
You selected the wrong region in the console.	Follow these steps to select the correct region: 1. Log on to the ApsaraDB for MongoDB console. 2. In the top navigation bar, select the region where your instance resides.
Your instance does not belong to the instance list you selected.	Follow these steps to switch to the instance list: 1. Log on to the ApsaraDB for MongoDB console. 2. In the top navigation bar, select the region where your instance resides. 3. In the left-side navigation pane, click **Replica Set Instances** or **Sharded Cluster Instances**. ② **Note** Standalone and replica set instances both appear on the **Replica Set Instances** page.
The instance list in the ApsaraDB for MongoDB console was not updated or was updated before your instance was created.	Wait several minutes and then update the instance list to check whether the instance appears on the **Replica Set Instances** or **Sharded Cluster Instances** page.
Resources are insufficient.	The system may fail to create your instance due to insufficient resources. In this case, your payment is refunded. You can check the refund on the Orders page. After you confirm the refunded fees, you can try to create your instance in another region or zone. If you cannot select another region or zone, submit a ticket.

1.3. What is the difference between ApsaraDB for MongoDB and MongoDB?

ApsaraDB for MongoDB is a document database service that is compatible with MongoDB protocols. It supports most of the MongoDB commands and allows access from MongoDB-compatible clients.

For more information about the advantages of ApsaraDB for MongoDB, see Comparison between ApsaraDB for MongoDB and self-managed databases.

For more information about ApsaraDB for MongoDB, see What is ApsaraDB for MongoDB?.

1.4. What database versions does ApsaraDB for MongoDB support?

ApsaraDB for MongoDB supports the following MongoDB versions: 5.0, 4.4, 4.2, 4.0, and 3.4. We recommend that you use drivers running the same database version as the ApsaraDB for MongoDB instance to access it. You can download drivers in different languages from the MongoDB official website.

> ⑦ **Note**   For more information about differences between the database versions, see MongoDB versions and storage engines.

How do I view the database version of an ApsaraDB for MongoDB instance?

1. Connect to a replica set instance by using the mongo shell. For more information, see Connect to a replica set instance by using the mongo shell.

2. Run the following command to view the database version:

   ```
   db.version()
   ```

References

Upgrade MongoDB versions

1.5. Can I add nodes to an ApsaraDB for MongoDB instance?

You can add nodes to a replica set or sharded cluster instance.

> ⑦ **Note**   You are not allowed to add nodes to standalone instances.

- Replica set instances

  By default, a replica set instance has three nodes. For more information, see Change the configurations of a replica set instance.

- Sharded cluster instances

By default, a sharded cluster instance has two shards and two mongos. For more information, see Overview.

1.6. Does ApsaraDB for MongoDB support Mongoose?

ApsaraDB for MongoDB supports Mongoose.

ApsaraDB for MongoDB is fully compatible with MongoDB Community Edition.

1.7. Does ApsaraDB for MongoDB support embedded/nested documents?

Yes, ApsaraDB for MongoDB supports embedded/nested documents. In the following example, the fields key specifies an embedded/nested document.

```
{
        "_id" : ObjectId("5cf0e51d8d1acb8a892ca65e"),
        "id" : "16399864",
        "timestamp" : "1453185620",
        "tablename" : "houseinfo",
        "dbname" : "corp_officebuilding",
        "primaryKeys" : "Id",
        "class" : "class com.uban.dts.bean.DtsLog",
        "dbType" : "MYSQL",
        "fieldCount" : "138",
        "opt" : "UPDATE",
        "fields" : {
                "Status" : {
                        "dest" : "0",
                        "orgi" : "1420041600"
                }
        }
}
```

1.8. What commands are supported and not supported by ApsaraDB for MongoDB?

For more information about official MongoDB commands, see MongoDB commands.

The following table lists the commands supported on ApsaraDB for MongoDB.

Command category	Supported	Not supported

Command category	Supported	Not supported
Aggregation Commands	- aggregate - distinct - count - mapReduce	group ⑦ Note The command is not supported in MongoDB 3.4 and later.
Geospatial Commands	- geoNear - geoSearch	None
Query and Write Operation Commands	- insert - update - delete - findAndModify - getLastError - getPrevError - resetError - parallelCollectionScan	eval ⑦ Note The command is not supported in MongoDB 3.0 version. The command is removed in MongoDB 4.2.
Query Plan Cache Commands	- planCacheListFilters - planCacheSetFilter - planCacheClearFilters - planCacheListQueryShapes - planCacheListPlans - planCacheClear	None
Authentication	- logout - authenticate	- authSchemaUpgrade - copydbgetnonce - getnonce ⑦ Note The command is not supported in MongoDB 4.0 and later.
User Management Commands	- createUser - updateUser - dropUser - dropAllUsersFromDatabase - grantRolesToUser - revokeRolesFromUser - usersInfo	None

Command category	Supported	Not supported
Role Management Commands	- createRole - updateRole - dropRole - dropAllRolesFromDatabase - grantPrivilegesToRole - revokePrivilegesFromRole - grantRolesToRole - revokeRolesFromRole - rolesInfo - invalidateUserCache	None

Command category	Supported	Not supported
Diagnostic Commands	- explain - listDatabases - dbHash - listCommands - availableQueryOptions - buildInfo - collStats - dbStats - cursorInfo - dataSize - ping - profile - top - whatsmyuri - serverStatus - features - isSelf - validate - netstat - hostInfo ⓘ **Note** The command is supported in Mongodb_20190725_1.1.8 and later, and is not supported in Mongodb_20190409_1.1.7 and earlier. - currentConn The command is self-developed for ApsaraDB for MongoDB and can be used to view the current connection status of an instance. Example: db.adminCommand({currentConn:1}).	- driverOIDTest - connPoolStats - shardConnPoolStats - getCmdLineOpts - diagLogging

Command category	Supported	Not supported
Instance Administration Commands	- renameCollection - dropDatabase - listCollections - drop - create - cloneCollectionAsCapped - convertToCapped - filemd5 - createIndexes - listIndexes - dropIndexes - fsync - connectionStatus - collMod - reIndex - touch - getParameter - compact	- copydb - clone - clean - shutdown - logRotate - repairDatabase - repairCursor - setParameter - connPoolSync - cloneCollection
Replication Commands	- isMaster - applyOps	- replSetInitiate - replSetFreeze - replSetMaintenance - replSetGetConfig - replSetRequestVotes - replSetReconfig - replSetStepDown - replSetSyncFrom - replSetElect - replSetUpdatePosition - resync - appendOplogNote
Sharding Commands	-	- addShard - removeShard - getShardVersion - setShardVersion - unsetSharding - checkShardingIndex

1.9. What is the impact of changes made to the storage space on an ApsaraDB for MongoDB instance?

When you change the storage space of an ApsaraDB for MongoDB instance, there is a brief disconnection of about 30 seconds. This brief disconnection does not affect data in the instance.

If the storage space of an ApsaraDB for MongoDB instance is insufficient, you can expand the storage space. For more information, see Overview.

> ⑦ **Note**   If your application is in a production environment, we recommend that you use a connection string URI to connect to the instance. This way, the read/write operations of your application remain available even if there is a primary/secondary switchover. For more information, see Connect to a replica set instance.

1.10. What is the node fault handling mechanism of ApsaraDB for MongoDB instances?

This topic describes the node fault handling mechanism of ApsaraDB for MongoDB instances.

Standalone instances

A standalone instance has only one node. If the node is faulty, the system troubleshoots the node. The relevant services become unavailable during troubleshooting.

> ⑦ **Note**   Standalone instances are applicable to test, training, and non-core business scenarios. If your application is in a production environment, we recommend that you use replica set or sharded cluster instances to ensure high availability (HA).

Replica set instances

Architecture of replica set instances
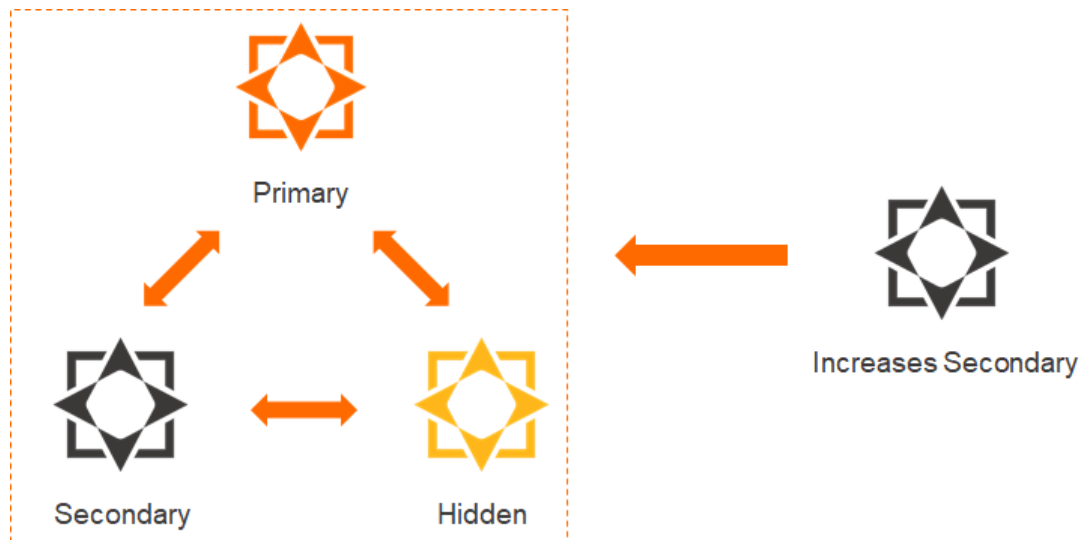
A replica set instance provides multiple nodes. If one of the nodes is faulty, the system switches over to the secondary or hidden node without interruption and then troubleshoots the faulty node. The entire process is transparent to you. A transient connection error of less than 30 seconds may occur during the process. We recommend that you configure your application to automatically reconnect to the replica set instance after a transient connection error occurs.

> ? **Note** If your application is in a production environment, we recommend that you do not use the connection string of the primary node but use a connection string URI to connect your application to a replica set instance. When you use the connection string URI, the read/write operations of your application remain available even if a node of the instance is faulty. For more information, see Connect to a replica set instance.

Sharded cluster instances

Architecture of sharded cluster instances
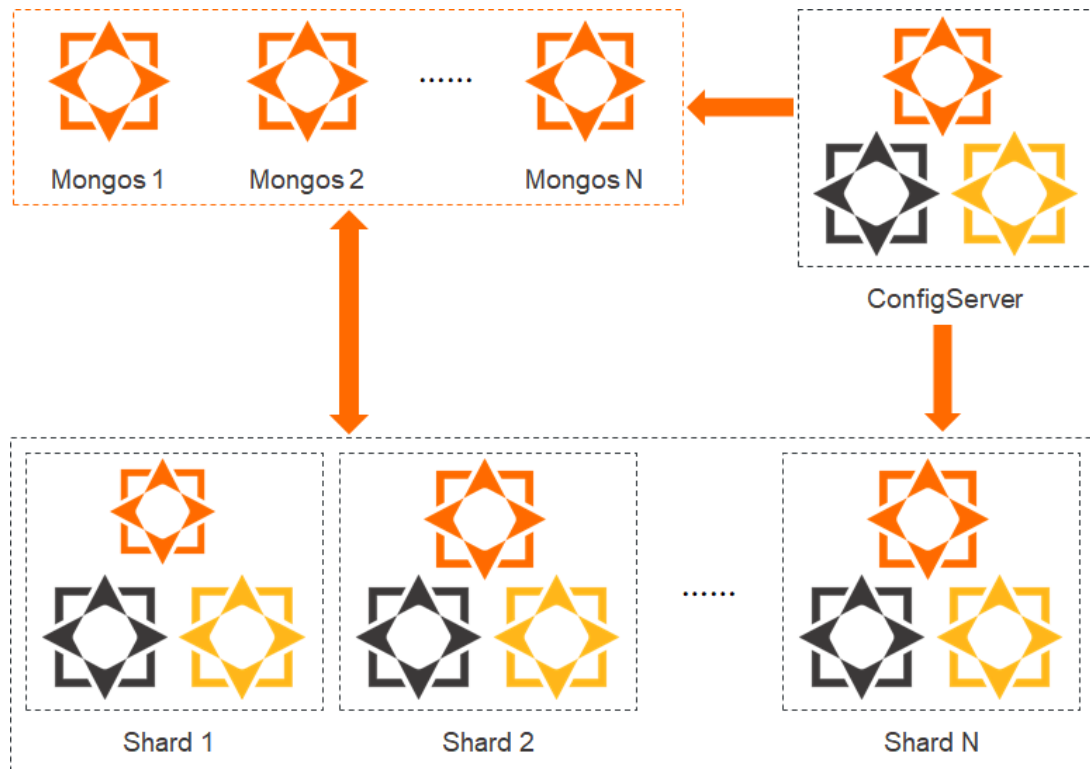
In a sharded cluster instance, both the shard and Configserver nodes use three-node replica set architecture. When a node is faulty, the system switches over to the hidden node without interruption and then troubleshoots the faulty node. The entire process is transparent to you. A transient connection error of less than 30 seconds may occur during the process. We recommend that you configure your application to automatically reconnect to the sharded cluster instance after a transient connection error occurs.

> ⑦ **Note**
>
> - A mongos node uses a single-node architecture. If a mongos node is faulty, the relevant services become unavailable.
>
> - If your application is in a production environment, we recommend that you do not use the connection string of a mongos node but use a connection string URI to connect your application to a sharded cluster instance. When you use the connection string URI, your client automatically redirects requests to the mongos node in the normal status if the connected mongos node is faulty. For more information, see Connect to a sharded cluster instance.

2.Database connections

2.1. Troubleshoot logon issues for the mongo shell

You can use DMS or the mongo shell to log on to the ApsaraDB for MongoDB console. This topic describes the typical problems that may occur when you use the mongo shell to log on to ApsaraDB for MongoDB and the corresponding solutions.

The message "connection attempt failed" is displayed

Issue:

```
#mongo --host ali12345678.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase admin -u r
oot -p xxx
MongoDB shell version: 3.2.3
DB Prefix:
connecting to: 10.1.2.8:3717/admin
2016-05-31T15:25:58.940+0800 W NETWORK  Failed to connect to 10.*.*.8:3717 after 5000 milli
seconds, giving up.
2016-05-31T15:25:58.943+0800 E QUERY    Error: couldn't connect to server 10.*.*.8:3717 (10
.1.2.8), connection attempt failed
    at connect (src/mongo/shell/mongo.js:181:14)
    at (connect):1:6 at src/mongo/shell/mongo.js:181
exception: connect failed
```

Possible cause	Solution

Possible cause	Solution
The Elastic Compute Service (ECS) instance on which you run the mongo shell command and the ApsaraDB for MongoDB instance are not in the same virtual private cloud (VPC) or have different network types.	- If the ECS instance and the ApsaraDB for MongoDB instance are not located in the same VPC, switch the network type of the ApsaraDB for MongoDB instance to classic network and then switch back to VPC. ⓘ **Note** When you switch the network type of the ApsaraDB for MongoDB instance back to VPC, select the VPC of the ECS instance for the ApsaraDB for MongoDB instance. - If the ECS instance and the ApsaraDB for MongoDB instance have different network types, perform the corresponding operations. For more information, see Connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different.

Supplementary troubleshooting method: You can run a Telnet command such as `telnet dds-ali123456789.mongodb.rds.aliyuncs.com 3717` to check whether the ApsaraDB for MongoDB instance is accessible.

```
root@mongotest:~# telnet            .mongodb.rds.aliyuncs.com 3717
Trying            7...
Connected to         mongodb.rds.aliyuncs.com.
Escape character is '^]'.
```

This figure shows that the URL can be resolved and port 3717 works as expected.

The message "Authentication failed" is displayed

Issue:

```
#mongo --host ali12345678.mongodb.rds.aliyuncs.com:3717  --authenticationDatabase admin -u
root -p xxx
MongoDB shell version: 3.2.3
connecting to: 10.1.2.8:3717/test
2016-05-31T15:50:18.623+0800 E QUERY    Error: 18 Authentication failed.
    at DB._authOrThrow (src/mongo/shell/db.js:1271:32)
    at (auth):6:8
    at (auth):7:2 at src/mongo/shell/db.js:1271
exception: login failed
```

Possible cause	Solution
The username used for database logon is incorrect.	Log on to the database with the correct username.
The password used for database logon is incorrect.	Log on to the database with the correct user password. If you forget the password, reset the password for the root user in the ApsaraDB for MongoDB console. For more information, see Reset the password of an account for an ApsaraDB for MongoDB instance.
The logon user does not match the authentication database.	Match the user with the authentication database. For example, the root user is a user of the admin database, so the authentication database must be assigned as admin if you want to use this user for database logon.
The client version is outdated.	Make sure that the mongo shell version is 3.0 or later. For information about how to install the mongo shell, see Install MongoDB. For version requirements of clients in other programming languages, see Driver compatibility documentation.

A network error occurs when you run the isMaster command

Issue:

```
#mongo --host ali12345678.mongodb.rds.aliyuncs.com:3717 --authenticationDatabase test -u ro
ot -p xxxxxx
MongoDB shell version v3.4.10
connecting to: mongodb:ali1234567878.mongodb.rds.aliyuncs.com:3717/
2018-12-18T14:26:11.946+0800 E QUERY    [thread1] Error: network error while attempting to
run command 'isMaster' on host 'ft12345678.mongodb.rds.aliyuncs.com:3717'  :
connect@src/mongo/shell/mongo.js:237:13
@(connect):1:6
exception: connect failed
```

Possible cause	Solution
The IP address of the ECS instance is not in a whitelist of the ApsaraDB for MongoDB instance.	Add the IP address of the ECS instance to a whitelist of the ApsaraDB for MongoDB instance. For more information, see Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

The message "Timeout while receiving message" is displayed

```
org.springframework.data.mongodb.UncategorizedMongoDbException: Timeout while receiving mes
sage; nested exception is com.mongodb.MongoSocketReadTimeoutException: Timeout while receiv
ing message
```

Possible cause	Solution

Possible cause	Solution
Abnormal slow queries occupy instance resources and cause CPU utilization to surge or even peak.	Check for slow queries. We recommend that you create indexes for optimization. For more information, see Troubleshoot the high CPU utilization of ApsaraDB for MongoDB.
The configuration of the application connection pool, such as the timeout setting, is incorrect.	For more information, see How do I query and limit the number of connections?

Common connection scenarios

- Connect to an ApsaraDB for MongoDB instance over the Internet
- Connect an ECS instance to an ApsaraDB for MongoDB instance when their network types are different
- How to connect an ECS instance to an ApsaraDB for MongoDB instance when they are in different regions
- Connect an ECS instance with an ApsaraDB for MongoDB instance in another Alibaba Cloud account

2.2. How can I connect to an ApsaraDB for MongoDB instance by using DMS?

You can connect to an ApsaraDB for MongoDB instance by using DMS.

For more information, see Connect to an ApsaraDB for MongoDB replica set instance by using DMS.

2.3. What do I do if the "Authentication failed" message is displayed?

You need to specify the database permissions for the login account.

If the "Authentication failed" message is displayed while you run the following command to connect to your ApsaraDB for MongoDB instance with the account that you created in the instance (both the username and password are test), see the following explanations:

```
mongo --host $myhost --port $myport -u test -p test
```

In ApsaraDB for MongoDB, an account is authorized for certain databases. To be authenticated when you connect to an ApsaraDB for MongoDB instance, you must specify the database where your account is created. For example, the test account is created in the admin database. You can run the following command to connect to your instance:

```
mongo --host $myhost --port $myport -u test -p test --authenticationDatabase admin
```

Or you can run the following commands:

```
mongo --host $myhost --port $myport
mongo> use admin
mongo> db.auth("test", "test")
```

2.4. What client languages does ApsaraDB for MongoDB support?

ApsaraDB for MongoDB is fully compatible with MongoDB. It supports all MongoDB-compliant clients.

Programming languages such as C, C++, C#, Java, Node.js, Python, PHP, and Perl are all supported. For more information, visit MongoDB Drivers and ODM.

> ? Note   To be authenticated when you connect to an ApsaraDB for MongoDB instance, install drivers that run MongoDB 3.0 or later.

2.5. How do I query and limit the number of connections?

This topic describes how to query the connection usage and how to specify the maximum number of connections for a connection pool. You can use Data Management (DMS) or the mongo shell to log on to ApsaraDB for MongoDB databases.

Query the number of current connections

The maximum number of connections varies based on the specifications of the ApsaraDB for MongoDB instance that you purchase. For more information, see Instance types.

> ? Note   The maximum number of connections applies to each node of the instance. For example, if you purchase a three-node replica set instance that has 1 vCPU and 2 GiB memory, up to 500 connections are allowed for each of the primary and secondary nodes of the instance. The hidden node does not provide services for external systems due to the special architecture of hidden nodes.

Use the mongo shell to connect to the instance. For more information, see Connect to an ApsaraDB for MongoDB instance. Then, run the `db.serverStatus().connections` command.

```
mgset-123456:PRIMARY> db.serverStatus().connections
{
        "current" : 1,
        "available" : 999,
        "internal_current" : 10,
        "internal_available" : 990,
        "totalCreated" : 632
}
```

> ⑦ **Note** Take note of the following parameters and the values of the parameters.
> - "current": the number of established connections.
> - "available": the number of available connections.

Query the source IP addresses of the current connections

1. Use the mongo shell to connect to an ApsaraDB for MongoDB instance. For more information, see Connect to an ApsaraDB for MongoDB instance. Then, switch to the admin database.

```
use admin
```

2. Run the `db.runCommand({currentOp: 1, $all: true})` command.

```
mgset-123456:PRIMARY> db.runCommand({currentOp: 1, $all:[{"active" : true}]})
```

You can query the source IP address of each connection in the command output. This way, you can obtain the number of connections that are established between each terminal and the ApsaraDB for MongoDB instance. For more information, visit db.currentOp().

Limit the number of connections

You can use a connection string URI to connect to an ApsaraDB for MongoDB instance. If you use a connection string URI to connect to a database, append `&maxPoolSize=<integer>` to the URI. maxPoolSize specifies the maximum number of connections in the connection pool.

The following code shows how to use the mongo shell to connect to the database. The maximum number of connections is set to 10 in this example.

```
mongo "mongodb://root:xxxxxx@dds-bpxxxxxxxx-pub.mongodb.rds.aliyuncs.com:3717,dds-bpxxxxxxx
x-pub.mongodb.rds.aliyuncs.com:3717/admin? replicaSet=mgset-xxxxxx&maxPoolSize=10"
```

> ⑦ **Note** For more information about how to limit the number of connections in the connection pool for clients in different programming languages, visit Start Developing with MongoDB.

2.6. How to troubleshoot database connection failures after the number of connections reaches the upper limit

After the number of connections to an ApsaraDB for MongoDB instance reaches the limit, new connection requests cannot be responded. This topic describes how to handle database connection failures after the number of connections reaches the upper limit.

The maximum number of connections varies depending on ApsaraDB for MongoDB instance types. For more information, see Instance types.

Symptom

- The application fails to connect to the database.

- The whitelist has been properly set. However, the following error message is displayed when you use the mongo shell to connect to the database:

```
2019-07-10T10:30:43.597+0800 E QUERY    [js] Error: network error while attempting to run
command 'isMaster' on host 'dds-bpxxxxxxxx.mongodb.rds.aliyuncs.com:3717'  :
connect@src/mongo/shell/mongo.js:328:13
@(connect):1:6
exception: connect failed
```

- The whitelist has been properly set. However, the following error message is displayed when you use DMS to connect to the database.
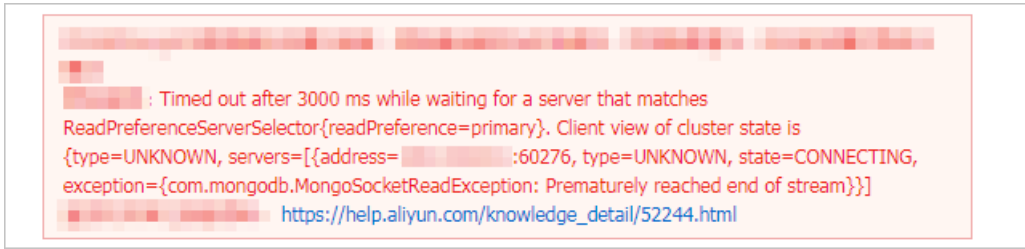
Check whether the number of connections has reached the upper limit

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and region to which the instance belongs.

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

4. On the page that appears, find the instance that you want to manage and click its ID.

5. In the left-side navigation pane, click **Monitoring Data**.

6. On the **Monitoring Data** page, check the **Connections** information. The following figure shows that the number of connections to the instance is 500.

> ⓘ **Note**　If the instance is a sharded cluster instance, you must select the **Mongos** node in use in the upper-right corner on the page.
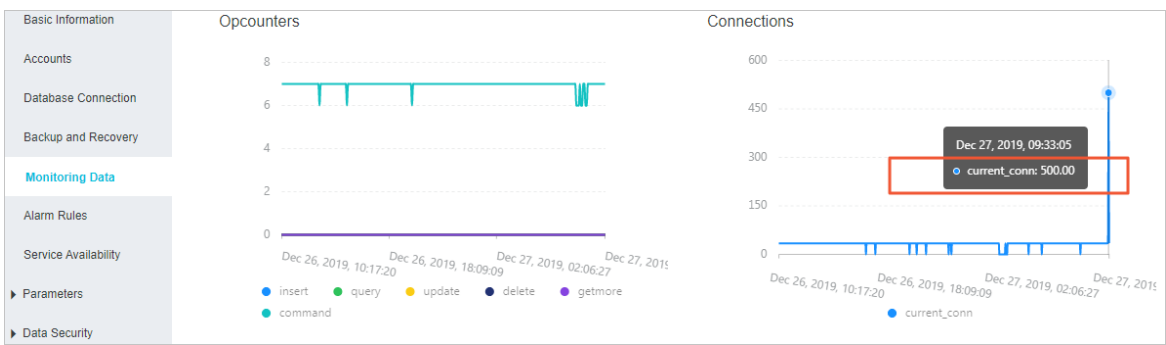
7. In the left-side navigation pane, click **Basic Information**.

8. On the **Basic Information** page, query the maximum number of connections corresponding to the current instance specifications. In this example, the number is 500.

> **Note** Based on the number of current connections, you can confirm that the number of connections has reached the upper limit.

Solution

You can restart the instance to temporarily release all connections. For more information, see Restart an ApsaraDB for MongoDB instance. To prevent this problem from occurring again, we recommend that you perform the following operations after restarting the instance:

> **Note** If you restart the instance, all instance nodes are restarted one by one. Each node has a transient disconnection of about 30 seconds. If there are a large number of collections (more than 10,000), the transient disconnections last longer. Before restarting the instance, arrange your business and ensure that your application has a reconnection mechanism.

- Configure the connection pool. For more information, see Limit the number of connections.
- Analyze the connection sources. For more information, see Query the source IP addresses of the current connections. If the service uses all the connections, upgrade the instance specifications. For more information, see Configuration change overview.

2.7. What do I do if the "Connection reset by peers" error is returned when I try to establish a connection to an ApsaraDB for MongoDB instance by using the mongo shell?

Errors

Errors similar to the following may be returned when you try to establish a connection by using the mongo shell:

```
    2015-12-21T10:20:36.084+0800 I NETWORK  Socket recv() errno:54 Connection reset by peer
1.2.3.4:27017
    2015-12-21T10:20:36.087+0800 I NETWORK  SocketException: remote: 1.2.3.4:27017 error: 9
001 socket exception [RECV_ERROR] server [1.2.3.4:27017]
    2015-12-21T10:20:36.087+0800 I NETWORK  DBClientCursor::init call() failed
```

Possible causes

The ApsaraDB for MongoDB instance rejects the connection request of your application because its connections are at the upper limit and cannot connect your application.

Solutions

1. Restart the ApsaraDB for MongoDB instance to release all connections.

2. Connect to a replica set instance by using the mongo shell.

3. Analyze the source IP addresses of connections and limit the number of connections from a client. For more information, see How do I query and limit the number of connections?.

> ⑦ Note   If the source IP addresses are normal, perhaps your current ApsaraDB for MongoDB instance is unqualified for your business needs. You can upgrade the instance to increase the upper limit of connections allowed. For more information, see Overview.

2.8. How can I install and use the mongo shell in a Windows operating system?

This topic describes how to install and use the mongo shell in a Windows operating system. The mongo shell allows you to connect to both MongoDB and ApsaraDB for MongoDB.

Install the mongo shell

1. Visit the MongoDB downloading center.

2. From the **Version** drop-down list, select the target MongoDB version.
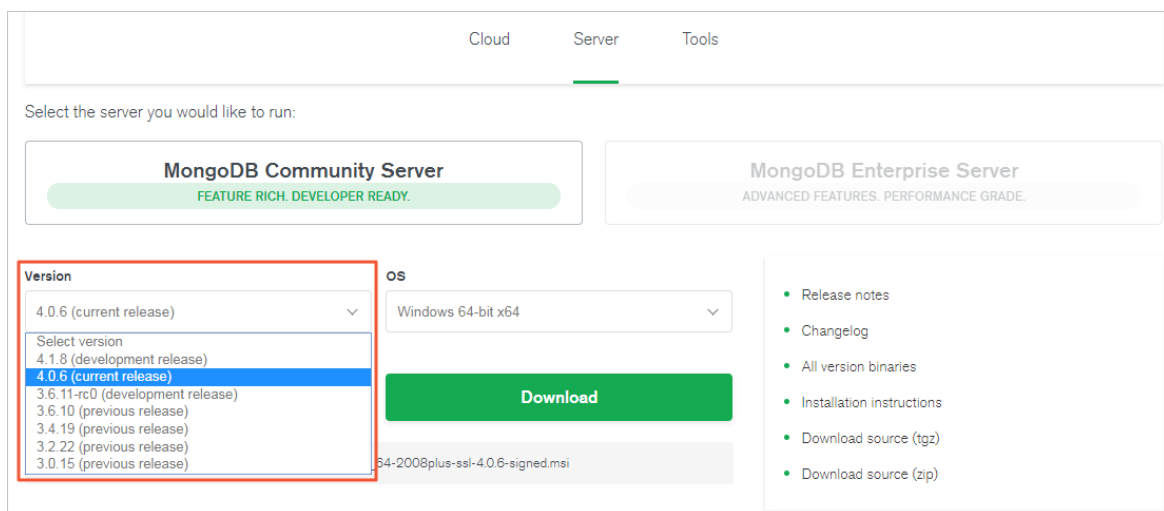
> ⑦ **Note**   To be authenticated when you connect to an ApsaraDB for MongoDB instance,
> select MongoDB 3.0 or later.

3. Double-click the downloaded software package to start the installation.

   Select **I accept the terms in the License Agreement** and click **Next**.
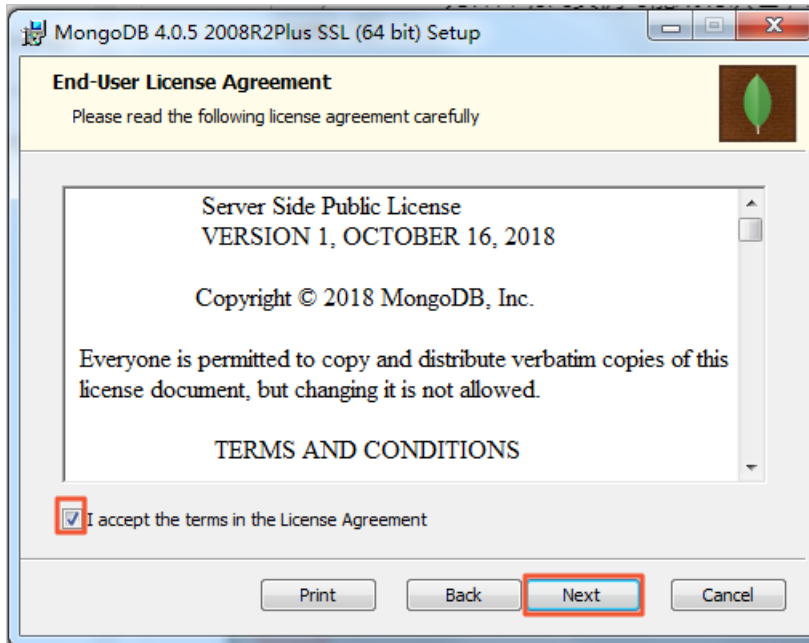
4. Click **Custom**.
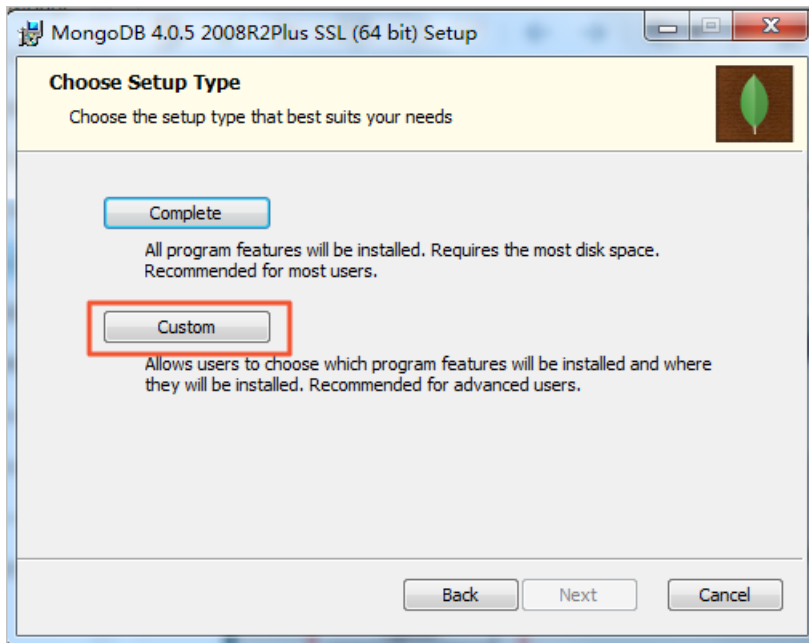
5. Only select **Client** and click **Next**.
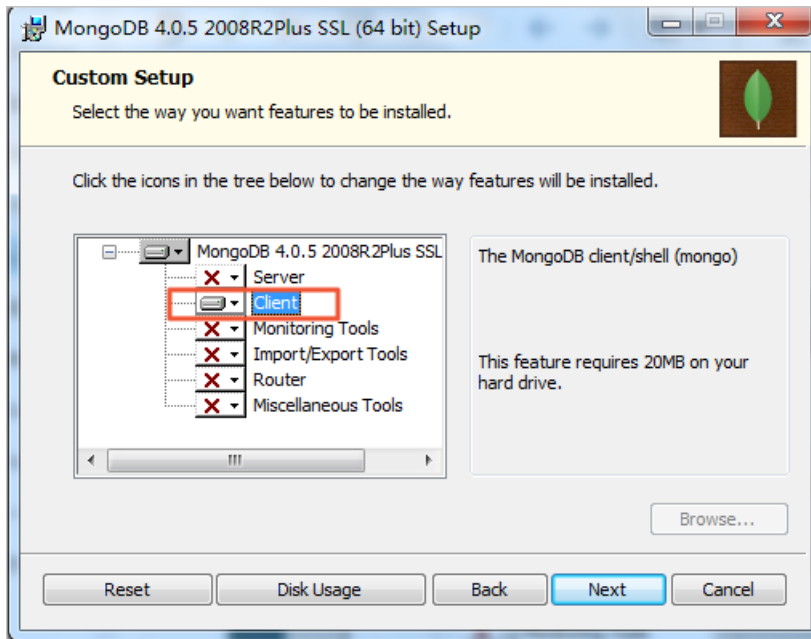
6. Wait until the mongo shell is installed.

Use the mongo shell

Open the command prompt window and go to the directory where the mongo.exe program is stored.

Example:

```
cd C:\Program Files\MongoDB\Server\4.0\bin
```

> ⑦ **Note**   After you open the directory of mongo.exe, you can use the mongo shell.

References

Connect to a replica set instance

3.Account and permission management

3.1. What permissions must the root account be granted to create an instance?

This topic describes the permissions of the root account.

After you create an ApsaraDB for MongoDB instance, the system automatically creates a root user in the admin database. This user has the root permissions of ApsaraDB for MongoDB. You can create accounts as needed. For more information, see Create an account for an ApsaraDB for MongoDB instance

3.2. Create an account for an ApsaraDB for MongoDB instance

After you create an ApsaraDB for MongoDB instance, the system automatically creates a root user in the admin database. This user has the root permissions of the ApsaraDB for MongoDB instance. You can log on to a database as the root user to create other users and grant them permissions.

You can create a user by running the `db.createUser()` command. For more information, visit db.createUser().

3.3. How to configure RAM user permissions on ApsaraDB for MongoDB

To implement fine-grained access control and improve account security, you can use Resource Access Management (RAM) to grant the management permissions on ApsaraDB for MongoDB to RAM users. In this way, RAM users can manage ApsaraDB for MongoDB instances.

Grant permissions to RAM users

1. Log on to the RAM console by using an Alibaba Cloud account.

2. Create a RAM user.

3. In the left-side navigation pane, click **Users** under **Identities**.

4. In the **User Logon Name/Display Name** column, find the target RAM user.

5. Click **Add Permissions** in the Actions column.
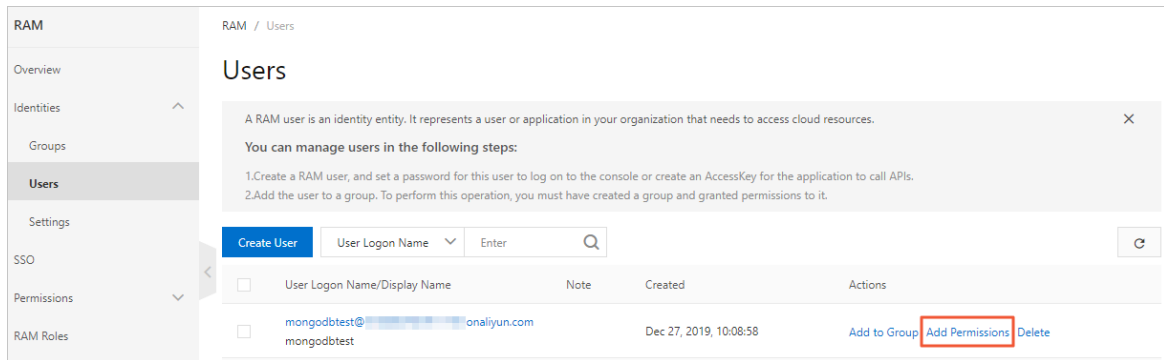
6. In the **Add Permissions** dialog box that appears, select permission policies as needed.
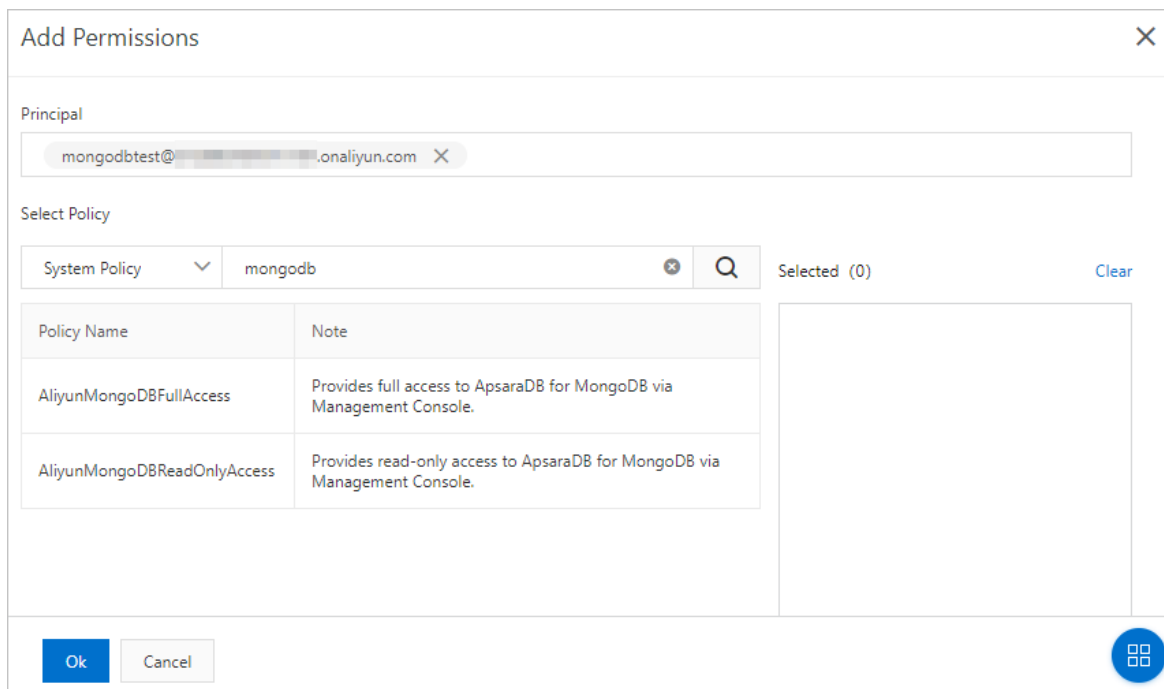
    i. Enter **mongodb** in the search box to display related permission policies.

> ⑦ Note
>
> ■ **AliyunMongoDBFullAccess**: grants RAM users full management permissions on ApsaraDB for MongoDB.
>
> ■ **AliyunMongoDBReadOnlyAccess**: grants RAM users the read-only permissions on ApsaraDB for MongoDB.

    ii. Click a policy name to add it to the **Selected** section.

7. Click **OK**.

8. Click **Finished**.

Customize permission policies in the RAM console

You can use system permission policies to grant RAM users permissions on all ApsaraDB for MongoDB resources. You can also customize permission policies as needed to grant RAM users specific operation permissions on specific instances. For information about the syntax of custom permission policies, see Policy structure and syntax.

Use RAM to grant permissions on ApsaraDB for MongoDB resources

You can only use RAM to grant permissions on ApsaraDB for MongoDB instances of the dbinstance type. When granting permissions using RAM, you can describe resources in the `Resource` field of the policy as follows.

Resource type	Resource description in the permission policy
dbinstance	- acs:dds:$regionid:$accountid:dbinstance/$dbinstanceid - acs:dds:$regionid:$accountid:dbinstance/ - acs:dds:::dbinstance/

Parameter description

Parameter	Description
`$regionid`	The region ID, which can be an asterisk (`*`).
`$dbinstanceid`	The instance ID, which can be an asterisk (`*`).
`$accountid`	The ID of your Alibaba Cloud account, which can be an asterisk (`*`).

Actions that you can authorize

In the RAM console, you can authorize RAM users to perform the following actions on a single ApsaraDB for MongoDB resource.

Action	Description
CreateDBInstance	Creates an instance.
ModifyDBInstanceSpec	Modifies instance specifications.
DeleteDBInstance	Deletes an instance.
DescribeDBInstances	Queries instances.
RestartDBInstance	Restarts an instance.
DescribeSecurityIps	Queries IP addresses in the whitelist.
ModifySecurityIps	Modifies IP addresses in the whitelist.
ResetAccountPassword	Resets the password of an account.
DescribeBackupPolicy	Queries the backup policy.

Action	Description
ModifyBackupPolicy	Modifies the backup policy.
CreateBackup	Creates a backup.
RestoreDBInstance	Restores an instance.
DescribeAccounts	Queries account information.
DescribeDBInstancePerformance	Queries the instance status.
DescribeReplicaSetRole	Queries the primary/secondary attribute of an instance.
ModifyDBInstanceDescription	Modifies the description of an instance.
ModifyAccountDescription	Modifies information about an account.
DescribeDBInstanceAttribute	Queries attributes of an instance.
RenewDBInstance	Renews an instance.
ModifyDBInstanceNetworkType	Modifies the network type of an instance.

3.4. How can I change the password of a database in an ApsaraDB for MongoDB instance?

You can set the password of the root user when you create an instance.

You can also change or reset the password in the ApsaraDB for MongoDB console. For more information, see (Optional) Reset a password.

4.Product features

4.1. How do I understand the names and sources of an IP address whitelist for ApsaraDB for MongoDB?

This topic describes the names and sources of an IP address whitelist for ApsaraDB for MongoDB. After you create an ApsaraDB for MongoDB instance, the instance has the default IP address whitelist. When you configure data migration or perform operations, more whitelists are generated.
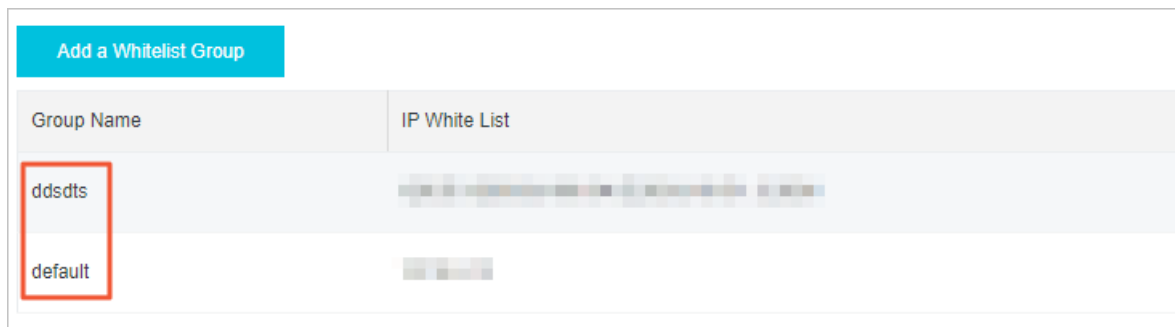
> ⑦ **Note**   You can open the **Whitelist Settings** page by following the instructions provided in Configure a whitelist or an ECS security group for an ApsaraDB for MongoDB instance.

IP address whitelist name	Source
default	The default IP address whitelist, which cannot be deleted.
ddsdts	The IP address whitelist automatically generated when you migrate your ApsaraDB for MongoDB instance. It contains the IP addresses of DTS servers. ⑦ **Note** When you migrate data, do not delete this IP address whitelist. Otherwise, data migration fails.

4.2. Does ApsaraDB for MongoDB support password-free access?

ApsaraDB for MongoDB supports password-free access if your instance is in a VPC.

For more information, see Enable or disable password-free access over a VPC for an ApsaraDB for MongoDB instance.

4.3. Does ApsaraDB for MongoDB support access over the Internet?

ApsaraDB for MongoDB supports access over the Internet.

For more information, see (Optional) Apply for a public endpoint for an ApsaraDB for MongoDB instance.

4.4. Does ApsaraDB for MongoDB support KMS encryption?

ApsaraDB for MongoDB supports Key Management Service (KMS) encryption.

You can enable Transparent Data Encryption (TDE) in the ApsaraDB for MongoDB console to encrypt and decrypt I/O flow for data files in real time. This way, data is encrypted before being written to disks and decrypted before being read from disks to memory. For more information, see Configure TDE for an ApsaraDB for MongoDB instance.

5.Performance and storage space

5.1. How frequent is monitoring data collected in ApsaraDB for MongoDB?

You can change the frequency for monitoring data of ApsaraDB for MongoDB by using the console.

The collection frequency for monitoring data is 60 seconds, which you can view on the **Monitoring Info** page in the ApsaraDB for MongoDB console. You cannot change Monitoring Info.

5.2. What do I do if my ApsaraDB for MongoDB instance is heavily loaded?

While an ApsaraDB for MongoDB instance is running, it may be overloaded, which slows down its read/write processing for your application. Analyze and adjust your application for troubleshooting.

For more information, see Troubleshoot the high CPU utilization of ApsaraDB for MongoDB.

5.3. What do I do if my ApsaraDB for MongoDB instance is locked due to exhausted disk space?

If the disk space of an ApsaraDB for MongoDB instance is exhausted, the instance is locked and you cannot write data to or delete data from it.

Symptoms

- You can read but cannot write data in the instance.
- The administrator enters the following command in the mongo shell, and `not authorized on xxxx to execute command` is returned.

```
db.customer.insert({"name":"zhangsan"})
WriteCommandError({
        "operationTime" : Timestamp(1563437183, 1),
        "ok" : 0,
        "errmsg" : "not authorized on db1 to execute command { insert: \"customer\", orde
red: true, lsid: { id: UUID(\"8d43461c-5c51-49ef-b9b3-9xxxxxxxxf\") }, $clusterTime: { cl
usterTime: Timestamp(1563437183, 1), signature: { hash: BinData(0, 0C3FAAE747xxxxxx), key
Id: 668293399xxxxxx } }, $db: \"db1\" }",
        "code" : 13,
        "codeName" : "Unauthorized",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1563437183, 1),
                "signature" : {
                        "hash" : BinData(0,"DD+q50dPTuIQKTzytT5SiTPYX4Q="),
                        "keyId" : NumberLong("66xxxxxxxx")
                }
        }
})
```

- The administrator finds that the instance status is **Locked** in the ApsaraDB for MongoDB console.

> ⑦ **Note**   Sharded cluster instances are a special case. When the disk space of a sharded
> cluster instance is exhausted, the instance does not enter the **Locked** state.

Check whether disk space is exhausted

1. Log on to the ApsaraDB for MongoDB console.

2. In the upper-left corner of the page, select the resource group and region to which the instance belongs.

3. In the left-side navigation pane, click or based on the instance type.

4. On the page that appears, find the instance that you want to manage and click its ID.
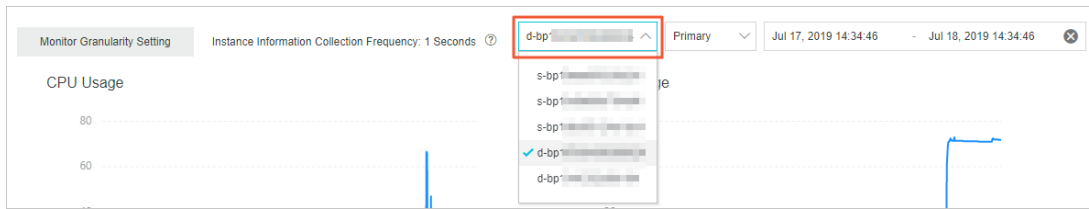
5. Check whether disk space is exhausted.

> ⑦ **Note**   The system collects data on disk space usage at intervals of five minutes.

- Standalone or replica set instances

  On the **Basic Information** page, view the instance status and its disk space usage. In this example, the instance status is **Locked**, and the disk space usage of the instance exceeds 100%, which indicates that the disk space is exhausted.
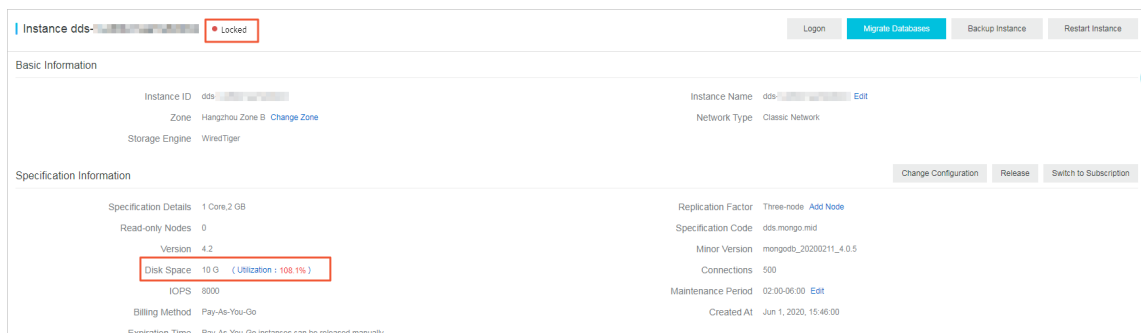
- Sharded cluster instances

  a. In the left-side navigation pane, click **Monitoring Data**.

  b. On the **Monitoring Data** page that appears, select the target shard.

> ⑦ **Note** If an ID starts with the letter `d`, it indicates a shard. If an ID starts with the letter `s`, it indicates a mongos.

  c. View disk space usage. In this example, the disk space usage of the shard exceeds 100%, which indicates that the disk space is exhausted.
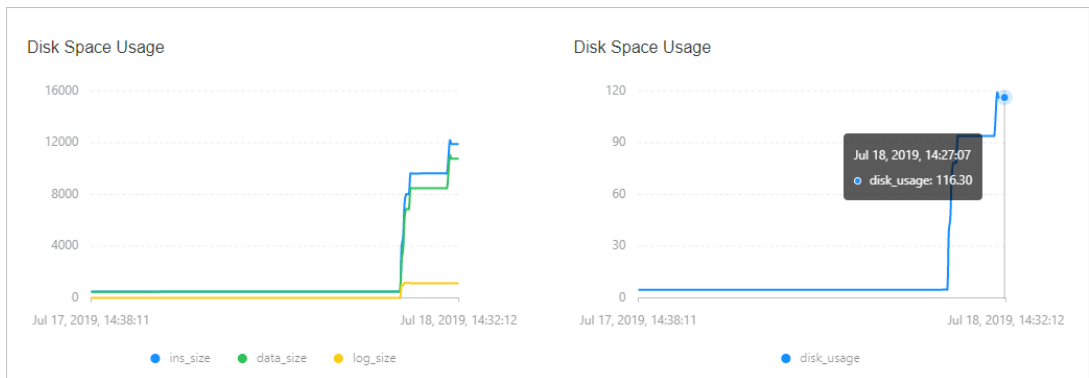
Solutions

- Expand the disk space of the instance. For more information, see Overview.

  > ⑦ **Note** Maximum disk space varies with instance types. For more information, see Instance types.

- A replica set instance supports up to 3,000 GB of disk space. If you need greater disk space, we recommend that you deploy a sharded cluster instance where you can add shards to expand the disk space up to 96,000 GB.

  > ⑦ **Note** You can use Data Transmission Service (DTS) to migrate data from a source instance to a new sharded cluster instance. For more information, see Migrate data from a replica set instance to a sharded cluster instance.

Suggestions

If you have executed the `db.collection.remove` command to delete a large volume of data or you have not defragmented your disk, you can still increase disk space by performing defragmentation. For more information, see Defragment the disk space to improve disk usage.

5.4. How can I check slow requests in an ApsaraDB for MongoDB instance?

ApsaraDB for MongoDB records the details of slow requests in the system.profile collection of each database. It enables slow request analysis by default.

You can view the system.profile collection to check slow requests. For more information, see Analyze slow requests in ApsaraDB for MongoDB.

5.5. Can I forcibly terminate a request which is being processed for a very long period?

If a request is being processed for a very long period, you can forcibly terminate the request.

For more information, see View and manage requests being processed.

6.errors and exception handling

6.1. Why does ApsaraDB for MongoDB trigger primary/standby switchovers?

ApsaraDB for MongoDB uses high availability (HA) architecture. When an ApsaraDB for MongoDB instance detects that one of its nodes is unavailable, it triggers a primary/standby switchover. It also sends a Short Message Service (SMS) message or internal notice to inform you of the switchover.

Internal notice

[Alibaba Cloud] Dear ******: Your ApsaraDB for MongoDB instance dds-bpxxxxxxxx (name: xxxxxx) has an error. A switchover is triggered to ensure stable running of your instance. Please check whether your application is still connected to your instance. We recommend that you configure your application to reconnect to the instance after it is disconnected.

Why do I receive the internal notice?

ApsaraDB for MongoDB uses HA architecture. A replica set instance of ApsaraDB for MongoDB contains three nodes by default, and a sharded cluster instance of ApsaraDB for MongoDB has each of its shards contain three nodes. Primary and secondary nodes are used for you to connect your application, and hidden nodes are used for primary/standby switchovers, which ensures HA. For more information, see Architecture of replica set instances or Architecture of sharded cluster instances.

ApsaraDB for MongoDB supports node health monitoring. When the monitoring results show an unavailable node, a primary/standby switchover is triggered.

Impact of primary/standby switchovers

- A primary/standby switchover causes a brief disconnection of less than 30 seconds.
- If you connect your application to a primary node, the read/write operations of your application are affected as a result of the primary/secondary switchover.

Suggestions

- We recommend that you configure your application to reconnect to an ApsaraDB for MongoDB instance after it is disconnected.
- If your application runs in a production environment, we recommend that you use a connection string URI to connect your application to the instance. In this way, the read/write operations of your application remain available even if a node fails as a result of a primary/secondary switchover. For more information, see Connect to a replica set instance or Connect to a sharded cluster instance.

ApsaraDB for MongoDB

Product Usage·How do I back up an
d restore an ApsaraDB for MongoDB
instance?

7.How do I back up and restore an ApsaraDB for MongoDB instance?

You can back up an ApsaraDB for MongoDB instance in either the automatic or manual mode, and restore it from backup files or to a point in time.

Backup

ApsaraDB for MongoDB stores its backup files in Object Storage Service (OSS) to reduce the storage space usage of its instances. You can perform an automatic or manual backup in the ApsaraDB for MongoDB console. For more information, see Configure automatic backup for an ApsaraDB for MongoDB instance or Manually back up an ApsaraDB for MongoDB instance.

> ⑦ **Note** If you choose automatic backup, only physical backup is supported.

Backup methods supported by ApsaraDB for MongoDB

Instance	Backup method	Impact
Standalone instances	Snapshot backup ⑦ **Note** The status of disk data at a specific point in time is retained.	Snapshot backup affects the I/O performance of standalone instances.
• Replica set instances • Sharded cluster instances	Physical backup ⑦ **Note** Physical data files of an instance are backed up. Logical backup ⑦ **Note** mongodump is used to logically back up each database.	Physical backup runs on a hidden node, which does not affect the read/write performance of the primary and secondary nodes. Backing up a large volume of data may take a long time.

Restoration

For more information, see Restoration solution overview.

Product Usage·What is the log dele tion policy of an ApsaraDB for Mong oDB instance?

ApsaraDB for MongoDB

8.What is the log deletion policy of an ApsaraDB for MongoDB instance?

Logs are automatically deleted when their size reaches a specific threshold.

In emergencies, you can run the `compact` command to delete oplogs. For more information, see Defragment the disk space to improve disk usage.

> ⑦ **Note**
> - You can only run this command on replica set instances.
> - Before running `compact` , you must log on to the local database where the oplog.rs collection is stored.