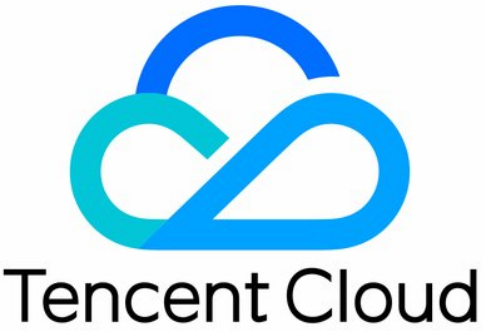
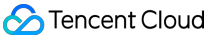
TencentDB for MongoDB

Quick Start



Copyright Notice

Trademark Notice



Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Quick Start
Creating TencentDB for MongoDB Instance

Last updated: 2025-02-08 10:09:12

Overview

The TencentDB for MongoDB console provides a simple and easy-to-use operation method. You can directly use the service through the console with no need to write code or run programs. Based on the introduction in this document, you can purchase and configure your first TencentDB for MongoDB instance.

Prerequisites

- You have registered a Tencent Cloud account and completed identity verification.
  - To register a Tencent Cloud account: Sign Up for Tencent Cloud.
  - To complete the identity verification: Go to Identity Verification.
- You have determined a region and AZ for the instance. For more information, see Region and Availability Zone.
- You have determined the specification and performance requirements of the instance. For more information, see Product Specifications and Service Performance.
- You have planned the VPC and security group for the database instance. For more information, see Create a VPC and Configuring a Security Group. Currently does not support external network access.
- You have understood the billing details of the database instance. For more information, see Billing Overview. When creating a pay-as-you-go database, one hour of database fees will be frozen. Ensure your account balance is sufficient before purchasing.
- You have planned the business project to which the database instance belongs. To create a new project, go to **account center** and select **Project Management**.
- You have understood the differences between various versions of the database. For more information, see Storage Engine and Version.

Operation Steps

1. Log in to the TencentDB for MongoDB purchase page with a Tencent Cloud account.
2. See the table below to configure instance specifications based on your actual needs.

Parameter Name	Mandatory or Not	Parameter Description
Billing Mode	Yes	Supports **annual and monthly subscription** and **pay-as-you-go**. For details on how to choose a billing method, see Billing Overview.
Region	Yes	Select a region where your instance resides. You should select a region closest to you to reduce access latency. • Note that the region cannot be changed after the instance is successfully created. • We recommend you select the same region as the CVM instance for private network communication.
Availability Zone	Yes	Specify whether to **enable multi-AZ deployment**. Multi-AZ means deploying instances in different AZs within the same region. Compared to single-AZ, where instances are deployed in the same AZ, multi-AZ instances have higher availability and disaster recovery capabilities. • If the instance is deployed in the same AZ, select the AZ for the primary node in the dropdown list behind **primary node**. • If the instance is deployed in different AZs, i.e., multi-AZ deployment is enabled, select the AZ for the primary node in the dropdown list of **primary node** and designate an AZ for each secondary node n (n=1,2,3,4,5,6) in the dropdown list. • If the number of read-only nodes is configured, please also configure the AZ for each read-only node.

		After creating the instance, Modifying Instance AZ is supported.
Specification Type	Yes	To select the specification type of the database instance, for more information, please see Specification Type. • **General Edition**: It is a long-term stable database service, deployed on high-performance servers with adequate bandwidth, allocating resource isolation to different instances; providing high-performance, high-reliability, and easy-to-manage MongoDB managed service. • **Cloud Disk Edition**: It is a cloud-native architecture database service built on various cloud services such as CVM and CBS. It fully integrates the capabilities of Tencent Cloud's underlying services, allowing to quickly adjust computing specifications, quickly back up and restore through cloud disk snapshots, providing more flexibility and convenience for product use.
Database Version	Yes	Select the database version. For how to choose a version, see Feature. • General Edition (High IO 10G): 7.0, 6.0, 5.0, 4.4, 4.2, 4.0. • Cloud Disk Edition: 5.0, 4.4, 4.2, 4.0. • After purchasing the instance, version upgrade is supported. For specific operations, see Version Upgrade.
Architecture Type	Yes	Select the system architecture of the instance cluster, supporting **replica set**, **sharded cluster**, and **single-node**. The single-node architecture has been discontinued. Please understand the use cases of different architectures and select the appropriate architecture based on the actual business data volume. For specific information, see System Architecture. Architecture upgrade is currently not supported.
Storage Engine	Yes	The default storage engine is **WiredTiger**.
Mongod Specification	Yes	Select the computing specifications of the database instance from the drop-down list. Higher specifications result in higher IOPS (requests per second). For supported specifications, see Product Specifications. After creating the instance, you can adjust the computing specifications of the instance. For specific operations, see Adjusting Instance Configuration.

Number of Mongod Shards	Yes	**Architecture Type**: When selecting **Sharded Cluster**, this parameter is displayed. It is used to set the number of shards in the sharded cluster, with a range of [1,20]. Each shard is a replica set. Increasing the number of shards can increase the storage capacity of the cluster. Select as needed. After creating the instance, you can adjust the shard count of Mongod. For specific operations, see Adjusting instance configuration.
Disk Capacity	Yes	Select the storage capacity of the database instance on the slider. • The disk capacity range varies with different Mongod specifications. For details, see Product Specifications. • The system defaults the oplog storage space to 10% of the selected storage capacity. The oplog size can be adjusted in the instance list in the console. For specific operations, see Adjusting Oplog Capacity. • After creating the instance, you can adjust the disk capacity of the instance. For specific operations, see Adjusting instance configuration.
Number of Primary and Secondary Nodes	Yes	, to display this parameter. • The default is three nodes (one primary, two secondary), forming a one-primary-two-secondary architecture with three storage nodes. Currently, you cannot customize the number of replicas. You can select five nodes (one primary, four secondary) or seven nodes (one primary, six secondary) from the dropdown list. • After creating an instance, you can increase the number of secondary nodes. For specific operations, see Adding Secondary Node.
Number of Primary and Secondary Nodes per Shard	Yes	**Architecture type**: When selecting **Sharded Cluster**, this parameter is displayed. It is used to set the number of nodes for each shard in the sharded cluster. • The system defaults to 3 nodes (1 primary and 2 secondary nodes), meaning each shard has a 3-node architecture with 1 primary and 2 secondary nodes. You can select 5 nodes (1 primary and 4 secondary nodes) or 7 nodes (1 primary and 6 secondary nodes) from the dropdown list. Customizing the number of nodes is not supported at this time. • After creating an instance, you can increase the number of secondary nodes for each shard. For specific operations, see Adding Secondary Node.

Number of Read-Only Nodes	Yes	Set the number of read-only nodes. • Supports no read-only nodes, 1-5 read-only nodes. • Only versions 6.0, 5.0, 4.0, 4.2, and 4.4 support configuring the number of read-only nodes, while version 3.6 does not. • After creating an instance, you can increase the number of read-only nodes. For specific operations, see Adding Read-Only Node.
Configuration Note	-	Calculate the maximum connections per instance based on the configured Mongod specifications to help you predict whether the current specifications meet your expectations.
Mongos Specs	Yes	**Architecture Type** selection **Sharded Cluster**, displays this parameter. Used to configure the specifications for Mongos. • After configuring the Mongod specifications, Mongos will have a default specification adaptation. For example, if Mongod is set to 2-core 4 GB, Mongos will be set to 1-core 2 GB by default. Upgrading mongos specifications will incur charges. For details on how to charge, refer to Product Pricing. • The connection limit of the sharded cluster will be determined by the mongos specifications and quantity you choose. You can view the maximum connections per instance in the configuration instructions. • After creating an instance, you can adjust the mongos configuration. For specific operations, refer to Adjust the mongos node specification.
Number of Mongos	Yes	**Architecture Type** selection **Sharded Cluster**, displays this parameter. Used to configure the number of mongos, instances are deployed in the same AZ, and the number of mongos ranges from [3,32]. If **AZ** is checked with **Enable Multi-AZ Deployment**, instances are deployed in different AZs, and the number of mongos ranges from [6,32]. • Increasing the number of mongos will incur charges. For how to charge, refer to Product Pricing. • After creating an instance, you can adjust the number of mongos. For specific operations, refer to Adding Mongos Node.
Network Type	-	Only supports selecting **VPC**.
IPv4 Network	Yes	Select the specific Virtual Private Cloud and its subnet. • Use a CVM instance to connect to the internal address automatically assigned to the cloud database. This connection

		method uses a high-speed internal network with low latency. The CVM instance and the database must be under the same account and within the same VPC (ensuring the same region). For more information, see Connecting to TencentDB for MongoDB Instance. • A virtual private cloud has a region attribute (such as Guangzhou), while a subnet has an availability zone attribute (such as Guangzhou Zone 1). A virtual private cloud can be divided into one or more subnets. Different subnets in the same virtual private cloud are communicable on the private network by default, and different virtual private clouds (regardless of whether they are in the same region) are isolated on the private network by default. • After purchasing an instance, you can switch the VPC. For specific operations, see Switching Networks. • You can also click **Create VPC** or **Create Subnet** to recreate the required network environment. For specific operations, see Create Virtual Private Cloud.
IPv6 Network	–	The current region and subnet do not support IPv6 address access.
Security Group	Yes	Set security group rules for an instance to control inbound traffic to the database. You can select existing security groups in the select existing security group drop-down list, or click **Customize Security Groups** to set new security group inbound rules. For more information, see Configuring a Security Group.
Specified Projects	Yes	Allocate a project to an instance. You can manage instances by project.
Tag	No	Set a tag to an instance. You can manage instances by tag. Click **Add** to select the tag key and tag value.
Parameter Template	Yes	Parameter template ID. A parameter template is a set of pre-configured parameter values for MongoDB. Save a set of parameters and values with the same requirements as a template, which can be directly referenced to the new instance when creating an instance. The system will accordingly set a default parameter template for each version. When purchasing an instance, after selecting the database version, the parameter template will accordingly select the default parameter template. You can select

		the newly defined parameter template from the drop-down list. For more information, see Creating a Parameter Template.
Instance Name	Yes	Set the instance name, default is 500. Please set a recognizable name. Supports Chinese characters, English letters, or digits with a length less than 60, en dash "-", and underscore "_" only.
Set Password	Yes	Select the instance password authentication method. • **Password Authentication**: You need to set the instance access password to access the database through password authentication. For more information, see Connecting to TencentDB for MongoDB Instance. • **Password-free Access**: No need to set a password. For more information, see Accessing Instance Without Authentication.
Password	No	When selecting **Password Authentication** in **Setting Password**, this parameter is displayed. It is used to set an access password for the instance. The password complexity requirements are as follows: • The number of characters is [8, 32]. • Characters can be in the range of [A, Z], [a, z], and [0, 9]. • Allowed special characters include: exclamation mark "!", at "@", hash "#", percent "%", caret "^", asterisk "*", parentheses "()", and underscore "_". • Cannot set a single letter or digit.
Confirm Password	No	Confirm the username and password again.
Purchase Quantity	Yes	• The maximum quota for annual and monthly subscription per instance is 10, i.e., the value range is [1,10]. • The maximum quota for pay-as-you-go per region is 30 instances, and the maximum quota per instance is 10 instances.
Purchase Duration	No	When choosing the **monthly subscription** billing mode, you need to select the purchase duration of instances. The longer the duration, the greater the discount. You can choose based on actual business needs.
Automatic Renewal	No	When choosing the **monthly subscription** billing mode, you can choose whether to enable the **auto-renewal** feature. After the fee expires, it will be automatically deducted monthly from the Tencent Cloud account.

Total Fees	Yes	• Choose monthly subscription to display the total fees for the purchased duration. • Choose pay-as-you-go to display the hourly fees. Click **Billing Details** to refer to Product Pricing.

3. Confirm that the parameter configuration is correct. Click **Purchase Now**, a success message will appear. Click **Go to Console**, in the instance list, wait for the instance status to display as **Running**. Now you can use it normally.

Subsequent Operations

- Use a CVM instance to directly access the intranet address of the cloud database. For more information, see Connecting to TencentDB for MongoDB Instance.
- View the instance list information and instance details. For more information, see Viewing Instance Details.

Relevant API

API Name	API Feature
CreateDBInstance	Creates a monthly subscribed TencentDB instance
CreateDBInstanceHour	Creates a pay-as-you-go TencentDB instance

Connecting to TencentDB for MongoDB Instance

Last updated: 2025-02-08 10:09:37

After the instance creation is successful, you can access the database through MongoDB shell or various language drivers to perform read, write, query, and other operations.

Basic Introduction

Shell Method

MongoDB Shell is an interactive JavaScript command-line management tool provided by MongoDB, which encapsulates many commonly used commands. You can install the MongoDB Shell tool on the Cloud Virtual Machine (CVM) and then use Shell commands to connect to the MongoDB instance to perform read, write, update, or query operations on the database.

> **Note:**
> Note: Using the Cloud Virtual Machine (CVM) to connect to the intranet address automatically assigned to the cloud database. This connection method uses the internal high-speed network and has low latency. The CVM and database must be in the same account and within the same VPC (ensuring the same region) or within the basic network.

URI Method

URI is a Uniform Resource Identifier, a unique identifier for each available resource on the Web. MongoDB official recommendation is to use URI form connection to MongoDB, and most drivers also support URI form connection.
A typical URI connection example is as follows:

```
mongodb://username:password@IP:27017/admin
```

```
mongodb://username:password@IP:27017/somedb?authSource=admin
```

```
mongodb://username:password@IP:27017/somedb?
authSource=admin&readPreference=secondaryPreferred
```

```
mongodb://mongouser:****@10.X.X.X:27017,10.X.X.X:27017/test?
replicaSet=cmgo-*****&authSource=admin
```

The meaning explanation of each part of the URI is shown in the table below. For more details, please refer to the MongoDB official documentation.

Components	Meaning	Required
mongodb://	A specific string representing the mongodb protocol	Yes
username	Username for logging into MongoDB	Yes, for more information, see default user
password	Password for logging into MongoDB	Yes
hostX:portX	IP and port of MongoDB	Yes
/admin	The database to be authenticated, TencentDB for MongoDB is fixed as admin	Yes, for more information, see authentication database
authMechanism=MONGODB-CR	Authentication Mechanism	Yes, for more information, see authentication mechanism
authSource=admin	The library used for identity authentication, TencentDB for MongoDB is fixed as admin	Yes, for more information, see authentication database
readPreference=secondaryPreferred	Can set to prefer to read from secondary	Yes, for more information, see Primary-Secondary Priority of Read Operations
replicaSet	Replica Set ID	Yes

Using Shell to Connect to Database

Introduce the operation method to connect to the database via mongo shell.

Prerequisites

- Register a Tencent Cloud account and complete real-name authentication.

- Create a Linux CVM instance in the same VPC and the same region as the cloud database MongoDB instance.
- A TencentDB for MongoDB instance has been created and its status is **running**.
- You have obtained the username and password information for database instance access on the **Database Management** page under the **Account Management** tab. For detailed directions, see Account Management.
- You have obtained the private IP and port for database instance access in the **Instance List**. For detailed directions, see Viewing Instance Details.

Operation Steps

Step 1: Logging in to the CVM

1. Log in to the Cloud Virtual Machine Console.
2. In the left sidebar, select **Instance**.
3. At the top of the instance management page, select the region.
4. In the instance list, find the requested CVM, and click **Log-in** in the **Operation** column on the right.
5. Enter the username and password set when applying for the CVM to log in to the cloud server.

Step 2: Downloading and extracting MongoDB Shell

1. Go to the path where MongoDB Shell is installed, use the `mkdir` command to create a separate folder for easy management.
2. In the path of the created folder, use the `wget` command to download MongoDB Shell. Example:

> **Note:**
> In the example below, l70-XX.XX.XX needs to be replaced with the actual version information and cannot be directly copied.

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-XX.XX.XX.tgz
```

> **Note:**
> When selecting the MongoDB Shell version, make sure to choose a version that matches the TencentDB for MongoDB service and the CVM operating system. For specific download information, see the Download Address.

3. Use the `tar` command to extract the downloaded MongoDB Shell installation package. Example:

```
tar zxvf mongodb-linux-x86_64-rhel70-XX.XX.XX.tgz
```

Step 3: Connecting to MongoDB

1. Use the cd command to navigate to the extracted path of MongoDB Shell. Example:

```
cd mongodb-linux-x86_64-rhel70-XX.XX.XX
```

2. Run the following command to enter MongoDB. Example:

```
./bin/mongo -umongouser -plxh***** 172.xx.xx.xx:27017/admin
```

Here, -u specifies the username for connecting to the database, -p specifies the password for the username, and 172.xx.xx.xx and 27017 specify the IP address and port of the primary node or secondary node of the MongoDB instance, respectively. Please replace them with the actual configuration information.

- If you forgot the username and password, refer to Account Management to view and modify account password information.
- You can get the IP address and port of the primary node or secondary node of the database instance in Node Management.
- For multiple IP access, configure multiple IPs separated by english commas, for example: `--host 172.XX.XX.XX:27017,172.XX.XX.XX:27017,172.30.XX.XX:27017`.

The connection success message is shown below. For more connection examples, refer to Shell Connection Sample.

```
MongoDB shell version v4.2.16
connecting to: mongodb://172.x.x.X:27017/admin?
compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("aeb18f32-6413-49da-864a-
5123b4d2****") }
MongoDB server version: 4.2.11
Welcome to the MongoDB shell.
```

Note:

- In a replica set, you can connect to the address of the Primary node, Secondary1 node, or Secondary2 node.

- Primary node: the primary node in a replica set instance, connecting to this node allows you to perform read and write operations on the database.
- Secondary node: the secondary node in a replica set instance, connecting to this node only allows you to perform read operations on the database.

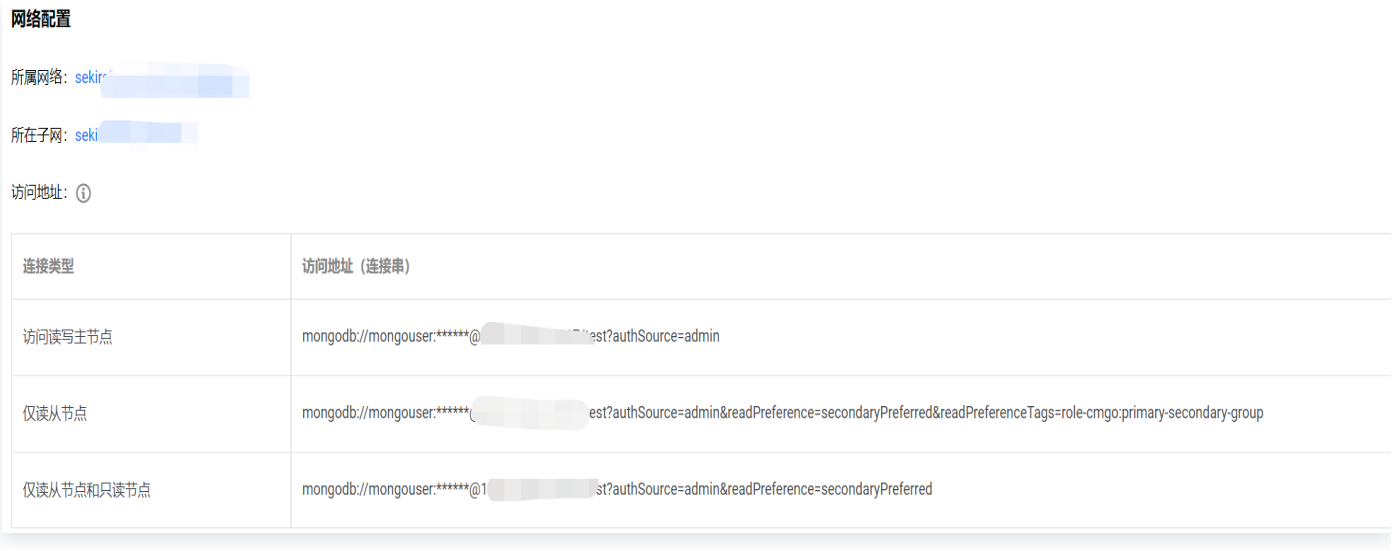
- In a sharded cluster, you can connect to the address of any mongos node.

Connecting to Database Using URI

Introduce the specific method for multilingual SDK client to connect to TencentDB for MongoDB using URI.

Prerequisites

- Register a Tencent Cloud account and complete real-name authentication.
- Prepare the multilingual SDK runtime environment.
- Obtain the URI connection string: Log in to the MongoDB console, find the target instance in the instance list, and copy it from the **access address** in the **Network Configuration** area of the **instance details** page.


网络配置

所属网络: seki

所在子网: seki

访问地址:

连接类型	访问地址（连接串）
访问读写主节点	mongodb://mongouser:******@ est?authSource=admin
仅读从节点	mongodb://mongouser:****** est?authSource=admin&readPreference=secondaryPreferred&readPreferenceTags=role-cmgo:primary-secondary-group
仅读从节点和只读节点	mongodb://mongouser:******@1 st?authSource=admin&readPreference=secondaryPreferred

Connection Sample

The minimum driver version required to connect to TencentDB for MongoDB is version 3.2. It is recommended to use the latest client driver to ensure the best compatibility, including shell suite, Java jar package, PHP extension, Node.js module, etc. For details, please refer to the MongoDB official website driver introduction. Below are specific examples of multilingual SDKs supported by TencentDB for MongoDB. You can concatenate the URI based on the examples, try to connect to the database, and perform read and write operations.

- PHP Connection Example
- Node.js Connection Sample

- mongoose Connection Example
- Java Connection Sample
- Python Connection Example
- Go Connection Example
- PHP Reconnection Mechanism

More References

default user

TencentDB for MongoDB has a built-in default user mongouser. Instances of version 3.2 support another built-in user rwuser. You can view system accounts and manage privileges to meet business needs on the MongoDB Console's **Database Management** page.

- **rwuser** is the only user using MONGODB-CR authentication. Example URI:

```
mongodb://rwuser:password@10.66.100.186:27017/admin?
authMechanism=MONGODB-CR
```

```
mongodb://rwuser:password@10.66.100.186:27017/somedb?
authMechanism=MONGODB-CR&authSource=admin
```

- **mongouser** and users created in the MongoDB Console use SCRAM-SHA-1 authentication. Example URI:

```
mongodb://mongouser:password@10.66.100.186:27017/admin
```

```
mongodb://mongouser:password@10.66.100.186:27017/somedb?
authSource=admin
```

authentication database

TencentDB for MongoDB uses the admin database for login authentication, so "/admin" must be added after the port in the URI to specify the authentication database. After authentication, switch to the specific business database for read and write operations. Example URI:

```
mongodb://username:password@IP:27017/admin
```

Alternatively, you can directly specify the read and write target database and additional authentication library parameters (authSource=admin) to directly access the target database.

Example URI:

```
mongodb://username:password@IP:27017/somedb?authSource=admin
```

> **Note:**
> Users created in the official website console have admin as the authentication database, so they need to specify the authentication database as admin when logging in. Users created via the command line, such as users created under the test database, need to specify the authentication database as test when logging in.

Therefore, you must choose a method to substitute admin as the authentication database into the URI.

Authentication Mechanism

TencentDB for MongoDB supports two authentication methods, MONGODB-CR and SCRAM-SHA-1, and has built-in two default users, rwuser and mongouser. It also supports creating other users in the TencentDB for MongoDB Console. Different users use different authentication mechanisms.

Username	Authentication Mechanism	URI Handling
rwuser	MONGODB-CR	You must add the parameter "authMechanism=MONGODB-CR"
mongouser and users created in the console	SCRAM-SHA-1 (recommended)	No parameters needed

Primary-Secondary Priority of Read Operations

TencentDB for MongoDB provides multiple connection methods to access the instance. To specify read access from the secondary database, set the readPreference parameter in the URI. The specific value meanings are as follows:

> **Note:**
> - When business needs read-write separation, it is more recommended to use secondaryPreferred for higher availability.
> - If there are business operations that only access read-only nodes, it is recommended to configure two or more read-only nodes to achieve read request load balancing with high availability guarantee. The connection string for read-only

> nodes can be directly obtained from the network configuration on the instance details page.

Value	Meaning
primary	Read-only primary node
primaryPreferred	Primary node preferred, if the primary node is unavailable, then read from secondary/read-only nodes
secondary	Secondary/read-only nodes, if the secondary node is unavailable, an error will be reported
secondaryPreferred	Secondary/read-only nodes preferred, if the secondary/read-only nodes are unavailable, then read from the primary node

To set the preference to read from secondary nodes, concatenate the URI according to the following example:

```
mongodb://username:password@IP:27017/admin?
readPreference=secondaryPreferred
```

Unable to connect

If the connection fails, refer to Troubleshooting MongoDB Connection Failures for step-by-step troubleshooting. For more issues, refer to Connection-related issues.

Reading/Writing Database

Last updated: 2025-02-08 10:10:02

After connecting to the database, you can create a database and write data.

Creating a Database

The syntax format for creating a database in MongoDB is as follows:

```
use DATABASE_NAME
```

Create a database named myfirstdb and insert a document.

```
> use myFirstDB
switched to db myFirstDB
> db.myFirstDB.insert({"test":"myFirstDB"})
WriteResult({ "nInserted" : 1 })
```

Query the created database.

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.041GB
myFirstDB  0.000GB
```

Creating a collection

In MongoDB, the createCollection() method is used to create a collection.
Syntax format:

```
db.createCollection(name, options)
```

Parameter description:

- name: Name of the collection to be created.
- options: Optional parameters specifying options related to memory size and indexes.

Options Field	Type	Description

capped	BOOL	Indicates whether to set the maximum byte size of the collection. If true, the size parameter must be set. Default is false.
autoIndexId	BOOL	Set whether to automatically create an index. If true, an index is automatically created on the _id field. Default is false.
size	Value	Set the maximum number of bytes for the collection.
max	Value	Set the maximum number of documents in the collection.

Example of creating the firstcol collection in the myfirstdb database:

```
> use myFirstDB
switched to db myFirstDB
> db.createCollection("FirstCol")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1634821900, 2),
                "signature" : {
                        "hash" :
BinData(0,"WFu7yj8wjeUBWG3b+oT84Q8wIw8="),
                        "keyId" : NumberLong("6990600483068968961")
                }
        },
        "operationTime" : Timestamp(1634821900, 2)
}
```

View the created collection:

```
> show collections
FirstCol
```

Create the firstcol collection with a maximum size of 6142800B and a maximum of 10000 documents, example as follows:

```
> db.createCollection("FirstCol", { capped : true, autoIndexId : true,
size : 6142800, max : 10000 } )
{
        "note" : "the autoIndexId option is deprecated and will be
removed in a future release",
```

```
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1634821879, 1),
                "signature" : {
                        "hash" :
BinData(0,"EuIbp2fu9Yh38HOBHLgYqljdKaE="),
                        "keyId" : NumberLong("6990600483068968961")
                }
        },
        "operationTime" : Timestamp(1634821879, 1)
}
```

Inserting Documents

- MongoDB uses the insert() or save() method to insert documents into the collection, example as follows:

```
> db.FirstCol.insert({name:"Li Si",sex:"Female",age:25,status:"A"})
WriteResult({ "nInserted" : 1 })
```

Viewing Inserted Documents in the Collection:

```
> db.FirstCol.find()
{ "_id" : ObjectId("61716957a6fe1ef4d7eae979"), "name" : "Li Si",
"sex" : "Female", "age" : 25, "status" : "A" }
```

- db.collection.insertMany() is used to insert one or more documents into the collection. The syntax format is as follows:

```
db.collection.insertMany(
  [ <document 1> , <document 2>, ... ]
  )
```

For example:

```
> db.FirstCol.insertMany([{name:"Li
San",sex:"Female",age:25,status:"A"},{name:"Wang
Liu",sex:"Male",age:26,status:"B"},{name:"Harry
Smith",sex:"Male",age:26,status:"A",groups:["news","sports"]}])
{
```

```
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("617282a3a4bb72d733b5c6d7"),
                ObjectId("617282a3a4bb72d733b5c6d8"),
                ObjectId("617282a3a4bb72d733b5c6d9")
        ]
}
```

Updating the Database

MongoDB uses update() to update documents in the collection.
Example of updating data in the FirstCol collection where name is Li San:

```
> db.FirstCol.update({name:"Li San", sex:"Female", age:25, status:"A"},
{$set:{'age':28}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Query modification results:

```
> db.FirstCol.find().pretty()
{
        "_id" : ObjectId("618904b6258a6c38daf13abd"),
        "name" : "Li San",
        "sex" : "Female",
        "age" : 28,
        "status" : "A"
}
{
        "_id" : ObjectId("618904b6258a6c38daf13abe"),
        "name" : "Wang Liu",
        "sex" : "Male",
        "age" : 26,
        "status" : "B"
}
{
        "_id" : ObjectId("618904b6258a6c38daf13abf"),
        "name" : "Wang Wu",
        "sex" : "Male",
        "age" : 26,
        "status" : "A",
        "groups" : [
                "news",
```

```
                "sports"
        ]
}
```

Delete Database

MongoDB uses remove() to delete documents in the collection. Example:

```
> db.FirstCol.remove({name:"Li San",sex:"Female",age:28,status:"A"})
WriteResult({ "nRemoved" : 1 })
```

Query delete results:

```
> db.FirstCol.find().pretty()
{
        "_id" : ObjectId("618904b6258a6c38daf13abe"),
        "name" : "Wang Liu",
        "sex" : "Male",
        "age" : 26,
        "status" : "B"
}
{
        "_id" : ObjectId("618904b6258a6c38daf13abf"),
        "name" : "Wang Wu",
        "sex" : "Male",
        "age" : 26,
        "status" : "A",
        "groups" : [
                "news",
                "sports"
        ]
}
```

More information

For more operation methods, please see MongoDB official documentation.

iptables Forwarding

Last updated: 2025-02-08 10:10:34

Overview

TencentDB for MongoDB does not currently support direct external access. You can achieve external access to MongoDB instances by using port forwarding through a CVM with a public IP.

> **Notes:**
> Because iptables-based forwarding may be unstable, we recommend that you do not access instances over the public network in the production environment.

Operation Steps

1. Log in to CVM and enable the CVM IP forwarding feature.

   > **Notes:**
   > The CVM and TencentDB instances must be under the same account and in the same VPC in the same region, or both in the classic network.

   ```
   echo 1 > /proc/sys/net/ipv4/ip_forward
   ```

2. Configure the forwarding rules. For example, forward access from 26.xx.x.2:10001 (public network address of the CVM, port can be chosen) to the intranet MongoDB instance at 10.0.0.5:6379.

   ```
   iptables -t nat -A PREROUTING -p tcp --dport 10001 -j DNAT --to-
   destination 10.0.0.5:6379
   iptables -t nat -A POSTROUTING -d 10.0.0.5 -p tcp --dport 6379 -j
   MASQUERADE
   ```

3. Configure CVM security groups to allow access to the external port of the CVM. It is recommended to configure security group rules to allow only the necessary source addresses.
4. On the client side, connect to the intranet MongoDB instance via the public network address (in this example, 26.xx.xx.2:10001). The connection method is the same as for intranet connections. Please refer to Connecting to TencentDB for MongoDB Instance.