

TencentDB for MongoDB

OPS Guide

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

OPS Guide

- Sharding Cluster Commands

- DynamoDB Cluster Commands

- DynamoDB Expressions

- MongoDB Official Commands

- Command Support in v3.6

- Development OPS

 - Database Operations in MongoDB 3.6

 - High CPU Utilization

 - Connection Problems

OPS Guide

Sharding Cluster Commands

Last updated : 2019-08-12 12:00:07

Sharding strategy

1. Ranged sharding is supported.
2. Shard keys or indexed compound fields are supported.
3. Sharding is required for all data sets in a sharding instance. It is recommended to place non-sharded data in a separate replica set instance.

Authentication method

Fully compatible with two methods: SCRAM-SHA-1 and MONGODB-CR.

Supported sharding cluster commands

	Commands	Subcommands	Supported
CRUD basic commands	find	filter	Yes
		sort	Yes
		projection	Yes
		hint	Yes
		skip	Yes
		limit	Yes
		batchSize	Yes
		singleBatch	Yes
		comment	Yes
		maxScan	Yes
		maxTimeMS	No
		readConcern	Yes
		max	Yes

	min	Yes
	returnKey	Yes
	showRecordId	Yes
	snapshot	No
	tailable	No
	oplogReplay	No
	noCursorTimeout	Yes
	awaitData	No
	allowPartialResults	No
insert	Must include the shardkey field, the shard keys must be consistent during batch insert operations.	Yes
update	The update field cannot be shardkey.	Yes
delete		Yes
findandmodify		Yes
count		Yes
distinct	Must include shard key.	Yes
aggregate		No
group		No
mapReduce		No
getmore		Yes
getLastError		No
getPrevError		No
resetError		No
eval		No
geoNear		No

	geoSearch		No
	parallelCollectionScan		No
Diagnostic commands	collStats		Yes
	dbstats		Yes
	explain		Yes
	listDatabases		Yes
	serverStatus		No
	top		No
Sharding commands	enableSharding		Yes
	shardCollection		Yes
Management commands	listCollections		Yes
	dropDatabase		Yes
	drop		Yes
	createIndexes		Yes
	listIndexes		Yes
	dropIndexes		Yes
	logout		Yes
	renameCollection		No
	copydb		No
	create		No
	clone		No
	cloneCollection		No
	cloneCollectionAsCapped		No
	convertToCapped		No
	filemd5		No

	fsync		No
	clean		No
	connPoolSync		No
	connectionStatus		No
	compact		No
	collMod		No
	reIndex		No
	setParameter		No
	getParameter		No
	repairDatabase		No
	repairCursor		No
	touch		No
	shutdown		No
	logrotate		No
	killop		No
User management commands			No
Role management commands			No
Replica set commands			No

DynamoDB Cluster Commands

Last updated : 2019-08-12 12:00:19

DynamoDB Protocol Support

API Name	Parameter	Supported
CreateTable	Required input parameter: TableName	Yes
	Required input parameter: AttributeDefinitions	Yes
	Required input parameter: KeySchema	Yes
	Required input parameter: ProvisionedThroughput	Yes
	Optional input parameter: GlobalSecondaryIndexes	Yes
	Optional input parameter: LocalSecondaryIndexes	Yes
	Optional input parameter: StreamSpecification	No
BatchWriteItem	Required input parameter: RequestItems	Yes
	Required input parameter: PutRequest	Yes
	Required input parameter: DeleteRequest	Yes
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ReturnItemCollectionMetrics	No
BatchGetItem	RequestItems	Yes
	ProjectionExpression	Yes
	ExpressionAttributeNames	Yes
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ConsistentRead	No
DeleteItem	ConditionExpression	Yes
	Key	Yes
	TableName	Yes
	Optional input parameter: Expected	Yes

	Optional input parameter: ConditionOperator	No
	Optional input parameter: ReturnItemCollectinMetricst	No
	Optional input parameter: ReturnConsumedCapacity	No
DeleteTable	Required input parameter: TableName	Yes
DescribeTable	Required input parameter: TableName	Yes
	Response parameter: Backfilling	No
	Response parameter: ProvisionedThroughput	No
	Response parameter: LatestStreamArn	No
	Response parameter: LatestStreamLabel	No
	Response parameter: StreamSpecification	No
GetItem	Required input parameter: TableName	Yes
	Required input parameter: Key	Yes
	Optional input parameter: ConsistentRead	No
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ProjectionExpression	No
ListTables	Required input parameter: TableName	Yes
	Optional input parameter: Limit	Yes
	Optional input parameter: ExclusiveStartTableName	Yes
PutItem	Required input parameter: TableName	Yes
	Required input parameter: Item	Yes
	Optional input parameter: ConditionExpression	Yes
	Optional input parameter: Expected	No
	Optional input parameter: ConditionOperator	No
	Optional input parameter: ReturnItemCollectinMetrics	No

	Optional input parameter: ReturnConsumedCapacity	No
	Response parameter: ConsumedCapacity	No
	Response parameter: ItemCollectionMetrics	No
Query	Required input parameter: TableName	Yes
	Optional input parameter: FilterExpression	Yes
	Optional input parameter: KeyConditions	Yes
	Optional input parameter: ConditionExpression	Yes
	Optional input parameter: Select	No
	Optional input parameter: ReturnItemCollectinMetrics	No
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ScanIndexForward	No
	Optional input parameter: ConditionOperator	No
	Optional input parameter: ConsistentRead	No
	Optional input parameter: Expected	No
	Optional input parameter: IndexName	No
	ProjectionExpression	Yes
UpdateItem	Required input parameter: Key	Yes
	Required input parameter: TableName	Yes
	Optional input parameter: Expected	No
	Optional input parameter: ConditionOperator	No
	Optional input parameter: ReturnConsumedCapacity	No
	Optional input parameter: ReturnItemCollectinMetrics	No
	UpdateExpression	Yes
	Response parameter: ConsumedCapacity	No
	Response parameter: .ItemCollectionMetrics	No

UpdateTable	Required input parameter: TableName	Yes
	Optional input parameter: TGlobalSecondaryIndexUpdates	Yes
	Optional input parameter: ProvisionedThroughput	No
	Optional input parameter: StreamSpecification	No
Scan		No
TagResource		No
UntagResource		No
ListTagsOfResource		No
DescribeLimits		No

DynamoDB Expressions

Last updated : 2019-08-12 12:00:29

DynamoDB supports three expressions: ConditionExpression, UpdateExpression and ProjectExpression. Their compatibilities with TencentDB are described below.

ConditionExpression

DynamoDBConditionExpressionThe syntax is as follows:

```
condition-expression ::=
| operand comparator operand
| operand BETWEEN operand AND operand
| operand IN ( operand (',' operand (, ...)) )
| function
| condition AND condition
| condition OR condition
| NOT condition
| ( condition )
```

comparator ::=

```
| =
| <>
| <
| <=
| >
| >=
```

function ::=

```
| attribute_exists (path)
| attribute_not_exists (path)
| attribute_type (path, type)
| begins_with (path, substr)
| contains (path, operand)
| size (path)
```

Notes:

1. For operand1 comparator operand2 syntax, operand1 must be an attribute field, while operand2 must be a value field.
2. For operand1 BETWEEN operand2 AND operand3 syntax, operand1 must be an attribute field, while operand2 and operand3 must be value fields.
3. For operand1 IN (operand2 (',' operand3 (, ...))) syntax, operand1 must be an attribute field, while operand2 and operand3 must be value fields.

4. For function syntax, `attribute_exists (path)`, `attribute_not_exists (path)`, `begins_with (path, substr)`, and `contains (path, operand)` are fully compatible. `attribute_type (path, type)` supports type-> S, N, B, BOOL, NULL, but not type-> SS, NS, BS, L, M. `size (path)` is not supported.

UpdateExpression

update-expressionThe syntax is as follows:

update-expression ::=

| **SET set-action** , ...

| **REMOVE remove-action** , ...

| **ADD add-action** , ...

| **DELETE delete-action** , ...

set-action ::=

path = value

value ::=

| operand

| operand '+' operand

| operand '-' operand

operand ::=

path | function

remove-action ::=

path

add-action ::=

path value

delete-action ::=

path value

function ::=

`if_not_exists (path, operand) | list_append (operand, operand)`

Notes:

[set-action]

1. For SET key=value, key must be an attribute field, while value must be a value field.

SET a =:b //Supported

SET #a = #b //Not Supported

- For SET key1=key2 + value, key1 and key2 must be attribute fields and have the same name, while value must be a value field.

```
SET a = a + :b //Supported
SET a = b + :b //Not supported
```

- For SET key1[NUM]=value, key1 must be an attribute field, while value must be a value field, and num must be a non-negative integer.
- SET key1[NUM]=key2[NUM] + value, not supported.
- SET key1[NUM]=key2[NUM] - value, not supported.
- SET key1=if_not_exists(key2, value), not supported.
- For SET key1=list_append (value, key2), key1 and key2 must be of value type and have the same name, while value must be of attribute type.
- For SET key1=list_append (key2, value), key1 and key2 must be of value type and have the same name, while value must be of attribute type.

[remove-action]

- For REMOVE key, key must be an attribute field, fully compatible.
- REMOVE key[NUM], not supported.

[delete-action]

- For DELETE key value, key must be of attribute type, while value must be of value type, fully compatible.

[add-action]

- For ADD key value, key must be of attribute type, while value must be of value type, fully compatible.

ProjectExpression

```
Projec-expression The syntax is as follows:
stat ::= sections
sections ::= section, sections
section ::= key| key[num]
```

Note: All the above syntax is supported except for the two-dimensional array section ::= key[num][num].

MongoDB Official Commands

Last updated : 2019-08-12 12:00:41

To learn more about MongoDB official commands, see the [link](#).

TencentDB for MongoDB 3.6 does not support the following commands:

Command Type	Unsupported Commands
Sharding Commands	addShard
	removeShard
Query and Write Operation Commands	getPrevError
Role Management Commands	dropAllRolesFromDatabase
Replication Commands	replSetAbortPrimaryCatchUp
	replSetFreeze
	replSetGetConfig
	replSetGetStatus
	replSetInitiate
	replSetMaintenance
	replSetReconfig
	replSetResizeOplog
	replSetStepDown
	replSetSyncFrom
Administration Commands	resync
	cloneCollection
	cloneCollection
	cloneCollectionAsCapped
	compact

	connPoolSync
	logRotate
	setParameter
	shutdown
Diagnostic Commands	availableQueryOptions
	dbHash
	getCmdLineOpts
	getLog
	shardConnPoolStats
System Events Auditing Commands	logApplicationMessage

Command Support in v3.6

Last updated : 2019-08-29 10:56:09

For a list of MongoDB's commands, please see MongoDB's [command documentation](#).

TencentDB for MongoDB v3.6 does not support the following commands:

Command Type	Unsupported Commands
Sharding Commands	addShard
	removeShard
Query and Write Operation Commands	getPrevError
Role Management Commands	dropAllRolesFromDatabase
Replication Commands	replSetAbortPrimaryCatchUp
	replSetFreeze
	replSetGetConfig
	replSetGetStatus
	replSetInitiate
	replSetMaintenance
	replSetReconfig
	replSetResizeOplog
	replSetStepDown
	replSetSyncFrom
Administration Commands	resync
	cloneCollection
	cloneCollection
	cloneCollectionAsCapped
	compact
	connPoolSync

	logRotate
	setParameter
	shutdown
Diagnostic Commands	availableQueryOptions
	dbHash
	getCmdLineOpts
	getLog
	shardConnPoolStats
System Events Auditing Commands	logApplicationMessage

Development OPS

Database Operations in MongoDB 3.6

Last updated : 2019-08-12 12:00:50

Problem Description

If you repeatedly drop a database and then create a new database with the same name in for a MongoDB 3.6 instance, an error "database does not exist" may occur when you try to read, write to or drop this database, as shown below:

```
mongos> show dbs
admin    0.000GB
config  0.001GB
local    0.209GB
scrm     0.001GB
mongos> use scrm
switched to db scrm
mongos> db.dropDatabase()
{
  "info" : "database does not exist",
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1546858044, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAA
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1546858044, 2)
}
mongos> █
```

Solution

It is a common problem, which may be caused by mongos not refreshing its metadata cache. For more information, see official [explanation](#), as shown below:

WARNING:

If you drop a database and create a new database with the same name, you must either restart all `mongos` instances, or use the `flushRouterConfig` command on all `mongos` instances before reading or writing to that database. This action ensures that the `mongos` instances refresh their metadata cache, including the location of the `primary shard` for the new database. Otherwise, the `mongos` may miss data on reads and may write data to a wrong shard.

Select one of the following two solutions for troubleshooting:

1. Restart `mongos` on the console.
2. Run the `flushRouterConfig` command. Click the link to see detailed description.

High CPU Utilization

Last updated : 2019-08-12 12:00:59

If you find the CPU utilization is high when using MongoDB, you can troubleshoot the problem as follows.

1. Check if your database operates too frequently.

You can check the monitoring metrics on the console. You may evaluate whether the instance needs to be upgraded if QPS is high. If the QPS is not high, check if there is slow operation log.

2. Check if there is a slow operation log on MongoDB.

Log in to the [Console of TencentDB for MongoDB](#) and view the slow operation log of the instance by using "Query statistics".

Please pay attention to keywords such as command, COLLSCAN, IXSCAN, keysExamined and docsExamined.

- command indicates the operations recorded in a slow operation log.
- COLLSCAN indicates a full table scan is performed. IXSCAN indicates an index scan is performed. For more information, see [MongoDB official website](#).
- keysExamined the number of index entries scanned. docsExamined indicates the number of documents scanned. Larger keysExamined and docsExamined values mean that no index is created or the index is less distinctive. Please confirm the fields for which an index is created.

For more log descriptions, please see [MongoDB official website](#).

Connection Problems

Last updated : 2019-08-12 12:01:27

You may encounter two types of connection problems when using MongoDB. You can troubleshoot the problems as follows.

High Connection Usage

If you find that the connection usage of the instance is too high, you can refer to the following procedure for troubleshooting.

1. Check whether a connection pool is used in the business.

The MongoDB service is provided in such a mode that each network connection is processed by a single thread (one-thread-per-connection). Too many network connections generate too many threads, which will increase context switch overhead and memory usage. Establishing connections and performing authentication for each request greatly affect performance. Therefore, it is recommended to use a connection pool for business to limit the number of connections of the instance and to release the connections that are no longer in use. All language versions of MongoDB Drivers generally encapsulate an object, so you need to construct a global object and use it in subsequent requests to send requests to the instance when using MongoDB. You can use the default connection pool size for the object in Driver, or configure the connection pool size by specifying the `maxPoolSize` option when constructing an object.

2. If you have set a connection pool, check your business for any unexpected exceptions.

- Check whether there is any business release change or code logic defect that leads to a large number of connections.
- Check whether there is any connection leak, i.e., check whether the number of connections on relevant CVM is exceptional.
- Check whether the business really has a large number of sudden requests.

3. Check whether there is any slow log that causes the connection to be occupied.

- If the business is confirmed to be normal, check if there is index exception. For example, check if a previously established index was deleted by mistake.
- If the index is normal, check whether there are a large number of slow logs, which occupy the connections and lead to establishment of more connections.

Log in to the [Console of TencentDB for MongoDB](#), and view the slow log of the instance by using "Query statistics".

!(<https://main.qcloudimg.com/raw/19a7b1568cf38f6b493cb5088cfdf93.png>)

Pay attention to keywords such as command, COLLSCAN, IXSCAN, keysExamined and docsExamined. For more log descriptions, see [MongoDB official website](#).

The keywords are described as follows:

- i. command indicates the operations recorded in a slow log.
- ii. COLLSCAN indicates a full table scan is performed. IXSCAN indicates an index scan is performed. For descriptions of other fields, see [MongoDB official website](#).
- iii. keysExamined refers to the number of index entries scanned. docsExamined refers to the number of documents scanned. Larger keysExamined and docsExamined values mean that no index is created or the index is less distinctive. Please confirm the fields for which an index is created.

4. Confirm whether the request is locked due to an index created at the foreground.

If there is no problem with the index used for business queries, check if an index is created at the foreground during peak business hours. An index is created at the foreground by default for a collection (the Background option is false), which will block all the other operations until the index is created at the foreground. If you select to create an index at the background, MongoDB can still provide read and write services during the creation of the index. However, it takes longer time to create an index in this way. For options to create an index, see [MongoDB official website](#).

You can view the progress of index creation using the currentOp command as shown below.

```
db.currentOp(  
{  
  $or: [  
    { op: "command", "query.createIndexes": { $exists: true } },  
    { op: "insert", ns: /\.system\.indexes\b/ }  
  ]  
}
```

The returned result is shown as follows. The msg field indicates the progress of index creation. The locks field indicates the lock type of the operation. For more information on locks, see [MongoDB official website](#).

```

"msg" : "Index Build Index Build: 3795542/30590193 12%',
"progress" : {
  "done" : 3795542,
  "total" : 30590193
},
"numYields" : 0.
"locks" : {
  "Global" : "w",
  "Database" : "W",
  "Collection" : "w"
},
"waitingForLock" : false,
"lockStats" : {
  "Global" : {
    "acquireCount" : {

```

Lock Mode	Description
R	Represents Shared (S) lock.
W	Represents Exclusive (X) lock.
r	Represents Intent Shared (IS) lock.
w	Represents Intent Exclusive (IX) lock.

5. Evaluate whether the instance configuration meets business requirements.

If the business connection usage shown on the console is still high after a connection pool is set and highly-distinctive indexes are created, it may indicate that the instance configuration cannot meet the actual business needs any longer. In such case, you need to evaluate the instance specification you actually need based on your business model, peak traffic, QPS, and TPS, and then upgrade your instance.

Connection Rejected

If a connection is rejected, you can troubleshoot the problem as follows.

1. Check whether the connection usage of the instance is 100%.

Log in to the [Console of TencentDB for MongoDB](#), and view the monitoring metrics "Number of Connections" and "Connection Usage" on the console.

If the connection usage of the instance is 100%, check whether your business is exceptional. If necessary, you can quickly release the connections by restarting mongos on the console.

All instance connections will be interrupted at the moment of restarting mongos, but the business can be directly reconnected. So restarting mongos will not continuously affect the business. If the number of business connections increases rapidly and the connection usage reaches 100% again after the restart, it indicates that the business does have a large number of valid connections and there is no connection leak. In such case, you need to find why there is such a large number of connections in the business by referring to the troubleshooting procedure for high connection usage.

2. Check whether the user name and the password are correct.

Check whether the user name and the password are correct. If they are incorrect, log in to the console and go to **Management** -> **Account Management** to modify them.

3. Check whether the Mongoshell version is correct.

To ensure successful authentication, install Mongo Shell 3.0 and above. For specific installation procedure, see the [official documentation](#).

4. Check whether the authentication database is correct.

The authentication database for users created on the console is the "admin" database, so the users need to specify "admin" as the authentication database on login. The users created with the command line, such as a user created under the "test" database, need to specify "test" as the authentication database.