

# **TencentDB for MongoDB**

## **Shard Feature**

### **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

Shard Feature

Shard Description

Code Example

# Shard Feature

## Shard Description

Last updated : 2018-09-14 15:45:55

### Sharding strategy

1. Supports ranged sharding.
2. You can combine the shard keys of fields.
3. Sharding is required for all data sets under a sharding instance. It is recommended to place non-sharded data in a separate replica set instance.

### Authentication mechanism

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

### Supported sharding cluster commands

	Commands	Subcommands	Support
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	Not Supported
		readConcern	Yes
		max	Yes

	min	Yes
	returnKey	Yes
	showRecordId	Yes
	snapshot	Not Supported
	tailable	Not Supported
	oplogReplay	Not Supported
	noCursorTimeout	Yes
	awaitData	Not Supported
	allowPartialResults	Not Supported
insert	Must include 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		Not Supported
group		Not Supported
mapReduce		Not Supported
getmore		Yes
getLastError		Not

		Supported
	getPrevError	Not Supported
	resetError	Not Supported
	eval	Not Supported
	geoNear	Not Supported
	geoSearch	Not Supported
	parallelCollectionScan	Not Supported
Diagnostic commands	collStats	Yes
	dbstats	Yes
	explain	Yes
	listDatabases	Yes
	serverStatus	Not Supported
	top	Not Supported
Sharding commands	enableSharding	Yes
	shardCollection	Yes
Management commands	listCollections	Yes
	dropDatabase	Yes
	drop	Yes
	createIndexes	Yes
	listIndexes	Yes
	dropIndexes	Yes

logout		Yes
renameCollection		Not Supported
copydb		Not Supported
create		Not Supported
clone		Not Supported
cloneCollection		Not Supported
cloneCollectionAsCapped		Not Supported
convetToCapped		Not Supported
filemd5		Not Supported
fsync		Not Supported
clean		Not Supported
connPoolSync		Not Supported
connectionStatus		Not Supported
compact		Not Supported
collMod		Not Supported
reIndex		Not Supported
setParameter		Not

			Supported
	getParameter		Not Supported
	repairDatabase		Not Supported
	repairCursor		Not Supported
	touch		Not Supported
	shutdown		Not Supported
	logrotate		Not Supported
	killop		Not Supported
User management commands			Not Supported
Role management commands			Not Supported
Replica set commands			Not Supported



# Code Example

Last updated : 2018-09-14 15:45:10

Python code is taken as an example to demonstrate basic data read/write in a MongoDB sharding cluster.

Sample Code:

```
#!/usr/bin/python
import pymongo
import random

mongodbUri = 'mongodb://mongouser:1234567a@10.66.153.111:27017/admin'

client = pymongo.MongoClient(mongodbUri)
db = client.test

if 'num' in db.collection_names():
    db.drop_collection('num')

#create database and shardkey,shardkey is name
db_admin=client.admin
db_admin.command('enableSharding', 'test')
db_admin.command('shardCollection', 'test.num', key = {'name':1})

#insert data
print 'insert docs'
db.num.insert_one({'id':1, 'name':'R9', 'des':'pretty'})
db.num.insert_one({'id':2, 'name':'BOY', 'des':'handsome'})
db.num.insert_one({'id':3, 'name':'cat', 'des':'nice'})
db.num.insert_one({'id':4, 'name':'dog', 'des':'clever'})
print 'list all docs'
for i in db.num.find(): print i

#insert update doc
print 'update R9 and delete BOY'
db.num.update_one({"name":"R9"}, {"$set":{"des":"good"}})
db.num.delete_one({"name":"BOY"})
db.num.update_one({"id":3}, {"$set":{"des":"kind"}})

print 'print R9'
for i in db.num.find({"name":"R9"}): print i
print 'list all docs'
```

```
for i in db.num.find(): print i
}
```

## Execution results

```
[root@vm_63_228_centos distribute_test]#
[root@vm_63_228_centos distribute_test]# python demo.py
insert docs
list all docs
{'_id': ObjectId('589c62e99d89702a48ebb10c'), 'des': u'pretty', 'id': 1, 'name': u'R9'}
{'_id': ObjectId('589c62e99d89702a48ebb10e'), 'des': u'nice', 'id': 3, 'name': u'cat'}
{'_id': ObjectId('589c62e99d89702a48ebb10f'), 'des': u'clever', 'id': 4, 'name': u'dog'}
{'_id': ObjectId('589c62e99d89702a48ebb10d'), 'des': u'handsome', 'id': 2, 'name': u'BOY'}
update R9 and delete BOY
print R9
{'_id': ObjectId('589c62e99d89702a48ebb10c'), 'des': u'good', 'id': 1, 'name': u'R9'}
list all docs
{'_id': ObjectId('589c62e99d89702a48ebb10c'), 'des': u'good', 'id': 1, 'name': u'R9'}
{'_id': ObjectId('589c62e99d89702a48ebb10e'), 'des': u'kind', 'id': 3, 'name': u'cat'}
{'_id': ObjectId('589c62e99d89702a48ebb10f'), 'des': u'clever', 'id': 4, 'name': u'dog'}
[root@vm_63_228_centos distribute_test]#
```