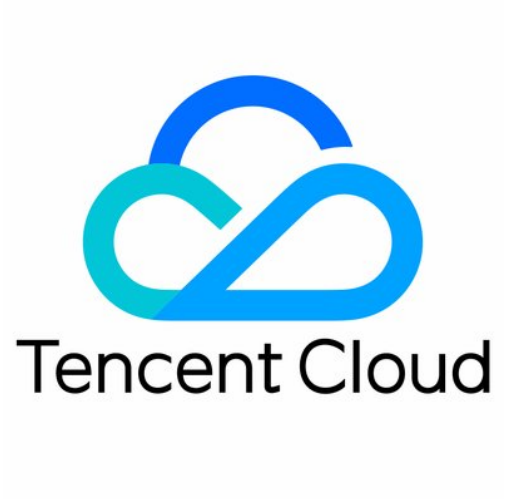
TencentDB for MongoDB

Best Practices

Product Documentation



Contents

Best Practices
Sample of Reading and Writing Data in MongoDB Instance

Last updated：2019-08-12 11:58:46

Below is the sample (Python) code illustrates basic data read/write operations 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]# python demo.py
insert docs
list all docs
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'pretty', u'id': 1, u'name': u'R9'}
{u'_id': ObjectId('589c62e99d89702a48ebb10e'), u'des': u'nice', u'id': 3, u'name': u'cat'}
{u'_id': ObjectId('589c62e99d89702a48ebb10f'), u'des': u'clever', u'id': 4, u'name': u'dog'}
{u'_id': ObjectId('589c62e99d89702a48ebb10d'), u'des': u'handsome', u'id': 2, u'name': u'BOY'}
update R9 and delete BOY
print R9
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'good', u'id': 1, u'name': u'R9'}
list all docs
{u'_id': ObjectId('589c62e99d89702a48ebb10c'), u'des': u'good', u'id': 1, u'name': u'R9'}
{u'_id': ObjectId('589c62e99d89702a48ebb10e'), u'des': u'kind', u'id': 3, u'name': u'cat'}
{u'_id': ObjectId('589c62e99d89702a48ebb10f'), u'des': u'clever', u'id': 4, u'name': u'dog'}
```

Sample of Reading and Writing Data in DynamoDB Instance

Last updated：2019-08-12 11:59:19

The following example shows how to access and perform read/write operations in DynamoDB instances with Python.

Sample code:

```
#!/usr/bin/python

from __future__ import print_function # Python 2/3 compatibility
import boto3
import json
import decimal
import time
from boto3.dynamodb.conditions import Key, Attr

#dynamodb = boto3.resource('dynamodb', region_name='us-west-2', endpoint_url="http://10.247.10.101:9000")
dynamodb = boto3.resource('dynamodb', region_name='us-west-2', endpoint_url="http://10.112.112.228:8000")

table = dynamodb.create_table(
TableName='music',
KeySchema=[
{
'AttributeName': 'Artist',
'KeyType': 'HASH' #Partition key
},
{
'AttributeName': 'SongTitle',
'KeyType': 'RANGE' #Sort key
},
],
AttributeDefinitions=[
{
'AttributeName': 'Artist',
'AttributeType': 'S'
},
{
'AttributeName': 'SongTitle',
'AttributeType': 'S'
```

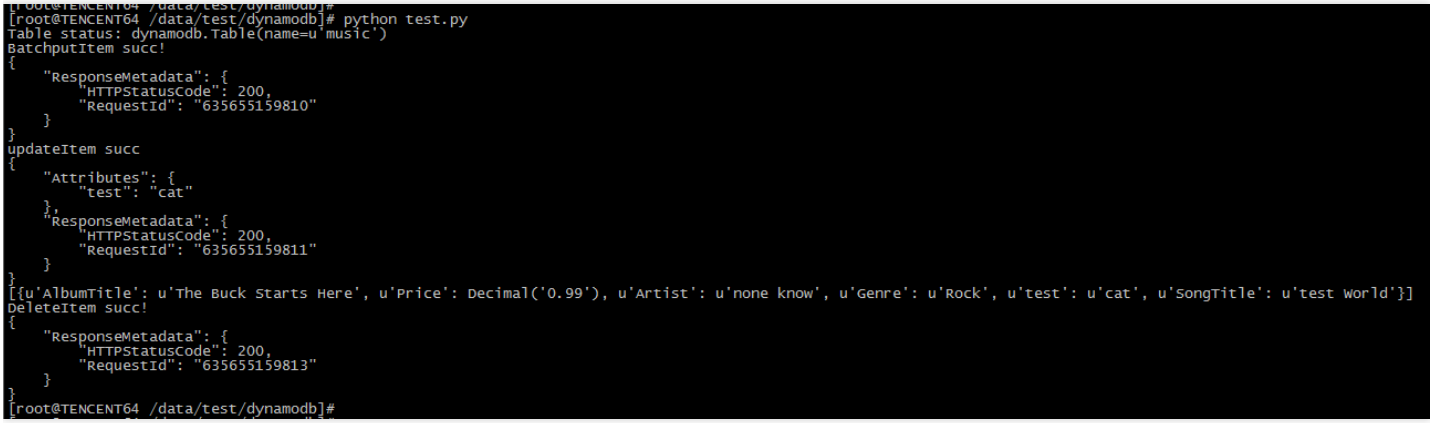
```
},

],

GlobalSecondaryIndexes=[
{
"IndexName": "t1Index",
"KeySchema": [
{
"AttributeName": 'Artist',
"KeyType": "HASH"
},
{
"AttributeName": 'SongTitle',
"KeyType": "HASH"
},
],
"Projection": {
"ProjectionType": "KEYS_ONLY"
},
"ProvisionedThroughput": {
'ReadCapacityUnits': 3,
'WriteCapacityUnits': 3
}
}
],

ProvisionedThroughput={
'ReadCapacityUnits': 3,
'WriteCapacityUnits': 3
}
)

print("Table status:", table)
#print "Table status: ",table

time.sleep(30)
#write data
response = dynamodb.batch_write_item(
RequestItems={
'music': [
{
'PutRequest': {
'Item': {
"Artist": "The Acme Band 2",
```

```
"SongTitle": "Look Out, World",
"AlbumTitle":"The Buck Starts Here",
"Price": decimal.Decimal('0.99'),
"Genre": "Rock",
}
}
},
{
'PutRequest': {
'Item': {
"Artist": "none know",
"SongTitle": "test World",
"AlbumTitle":"The Buck Starts Here",
"Price": decimal.Decimal('0.99'),
"Genre": "Rock",
}
}
},
{
'PutRequest': {
'Item': {
"Artist": "The Acme Band 4",
"SongTitle": "Look Out, World",
"AlbumTitle":"The Buck Starts Here",
"Price": decimal.Decimal('3.99'),
"Genre": "Rock 4",
}
}
},
{
'PutRequest': {
'Item': {
"Artist": "The Acme Band 5",
"SongTitle": "Look Out, World",
"AlbumTitle":"The Buck Starts Here 5",
"Price": decimal.Decimal('5.99'),
"Genre": "Rock",
}
}
},

]
},
ReturnConsumedCapacity='TOTAL',
ReturnItemCollectionMetrics='SIZE',
```

```
)

print("BatchputItem succ!")
print(json.dumps(response, indent=4))


#update data
t_name='music'
table = dynamodb.Table(t_name)

response = table.update_item(
Key={
"Artist": "none know",
"SongTitle": "test World"
},

UpdateExpression="SET test = :incr REMOVE Tags.del",
ExpressionAttributeValues={
":incr": 'cat'
},
ReturnValues="UPDATED_NEW"
)

print("updateItem succ")
print(json.dumps(response, indent=4))


#query data
t_name='music'
table = dynamodb.Table(t_name)

response = table.query(
KeyConditionExpression=Key('SongTitle').eq('test World')
)

items = response['Items']
print(items)


#delete Item
test_table='music'
table = dynamodb.Table(t_name)
response = table.delete_item(
Key={
'Artist': 'none know',
```

```
    "SongTitle": "test World",
    }
    )

    print("DeleteItem succ!")
    print(json.dumps(response, indent=4))
```

Execution results



Reconnection Mechanism

Last updated：2018-09-14 16:27:06

Description

Instead of simply allowing you to access mongod, TencentDB for MongoDB database service provides a load balancer IP for access. You can use this IP to connect to a range of route access layers similar to mongos.
The client driver establishes a persistent connection with an access server using a load balancer IP. If the connection is active for a long period of time, we will not impose an intervention on this status. However, if the persistent connection is inactive for more than one day (this period will be adjusted with optimized version), the route access layer will terminate the connection.
Generally, the client driver will implement an automatic reconnection. However, this process cannot be implemented by some language drivers. For the language drivers that cannot implement automatic reconnection, if you attempt to communicate with the TencentDB for MongoDB service using a terminated connection, an error message such as "Remote server has closed the connection" will be returned. So manual reconnection is required. Here is a demo for PHP reconnection.

Reconnection Based on PHP Mongo Driver

```
<?php

function getConnection() {
$connection = false;
$uri = 'mongodb://rwuser:1234567a@10.66.148.142:27017/admin?authMechanism=MONGODB-CR';
$maxRetries = 5;
for( $counts = 1; $counts <= $maxRetries; $counts++ ) {
try {
$connection = new MongoClient($uri);
} catch( Exception $e ) {
// Or use the catch code line below as required. Please note that, "\" is needed when some frameworks use namespace.
// } catch( \Exception $e ) {
continue;
}
break;
}
return $connection;
```

```
}

$connection = getConnection();

if($connection) {
$db = $connection->testdb;
$collection = $db->testcollection;

$one = $collection->findOne();

var_dump($one);
}
```

Import and Export

Last updated：2019-08-12 11:59:41

> The local database store stores metadata of a replica set, including configuration information and oplog. The admin database manages information about users and roles. To prevent data corruption, authentication failure and other issues, TencentDB for MongoDB does not allow to import local and admin databases into instances.

In CVM, you can use the MongoDB shell client to connect to TencentDB for MongoDB service for data import and export. Please use the latest MongoDB client suite. For more information, see **Operation Guide -> Connection Example**.

Import Commands

mongodump and mongorestore

MongoDB provides two sets of data import and export tools. If you need to import and export the entire database, we recommend mongodump and mongorestore. The data should be formatted in BSON, which facilitates massive data "dump" and "restore".
The mongodump import command is as follows:

```
mongodump --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase
=admin --db=testdb -o /data/dump_testdb
```

```
#: ./mongodump --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase=admin
 --db=testdb -o /data/dump_testdb
2016-11-16T12:12:06.114+0800    writing testdb.system.indexes to
2016-11-16T12:12:06.116+0800    done dumping testdb.system.indexes (1 document)
2016-11-16T12:12:06.116+0800    writing testdb.testcollection to
2016-11-16T12:12:06.118+0800    done dumping testdb.testcollection (3 documents)
```

The mongorestore import command is as follows:

```
mongorestore --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase
=admin --dir=/data/dump_testdb
```

```
#: ./mongorestore --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase=ad
min --dir=/data/dump_testdb
2016-11-16T12:13:23.654+0800    building a list of dbs and collections to restore from /data/dump_test
db dir
2016-11-16T12:13:23.678+0800    reading metadata for testdb.testcollection from /data/dump_testdb/test
db/testcollection.metadata.json
2016-11-16T12:13:23.678+0800    restoring testdb.testcollection from /data/dump_testdb/testdb/testcoll
ection.bson
2016-11-16T12:13:23.740+0800    restoring indexes for collection testdb.testcollection from metadata
2016-11-16T12:13:23.740+0800    finished restoring testdb.testcollection (3 documents)
2016-11-16T12:13:23.741+0800    done
#:
```

Export Commands

mongoexport and mongoimport

When you import and export a single collection, we recommend mongoexport and mongoimport . The data should be in JSON format for higher readability.

The mongoexport export command is as follows:

```
mongoexport --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase
=admin --db=testdb --collection=testcollection -o /data/export_testdb_testcollection.json
```

In addition, you can add the "-f" into the parameter to specify a desired field, and "-q" to specify a query condition that applies restriction on data export.
The mongoimport export command is as follows:

```
mongoimport --host 10.66.187.127:27017 -u mongouser -p thepasswordA1 --authenticationDatabase
=admin --db=testdb --collection=testcollection2 --file=/data/export_testdb_testcollection.json
```

Parameters of Authentication Methods

As described in the Connection Example, TencentDB for MongoDB provides two user names "rwuser" and "mongouser" by default to support the "MONGODB-CR" and "SCRAM-SHA-1" authentication respectively.

- For "mongouser" and all new users created in the console, they can simply follow the example above to use the import and export tools.
- For "rwuser", the parameter "--authenticationMechanism=MONGODB-CR" should be included in each command.

Example of mongodump:

```
mongodump --host 10.66.187.127:27017 -u rwuser -p thepasswordA1 --authenticationDatabase=admin --authenticationMechanism=MONGODB-CR --db=testdb -o /data/dump_testdb
```