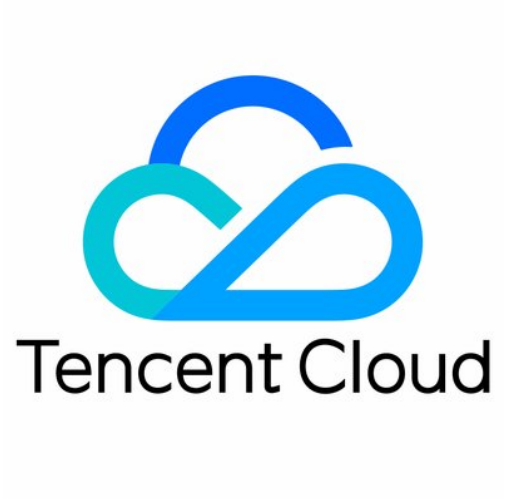
TencentDB for MongoDB

Instance Connection

Product Documentation



Contents

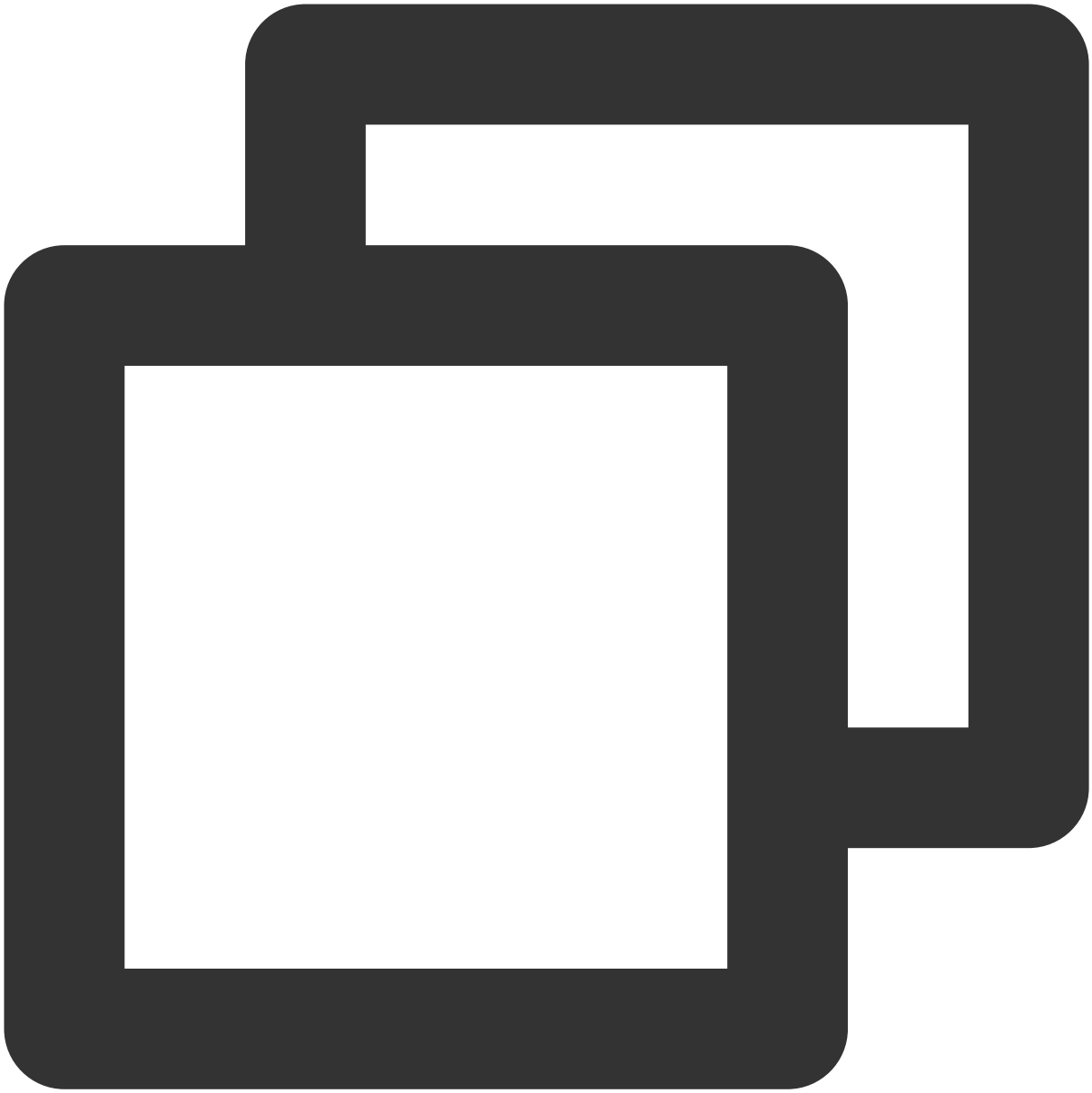
Instance Connection
Shell Connection Sample

Last updated：2024-01-15 14:49:55

You can use the MongoDB shell client (please see the installation documentation) on a CVM instance to connect to TencentDB for MongoDB for data management. Be sure to use the latest version of MongoDB client suite.

Quick start

A typical connection command is as follows:



```
mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
```

Note:

To access TencentDB for MongoDB via a connection string, special characters in the password need to be converted to URL encoded characters so that they can be correctly identified. For example, "@" should be converted to "%40". See the figure below:

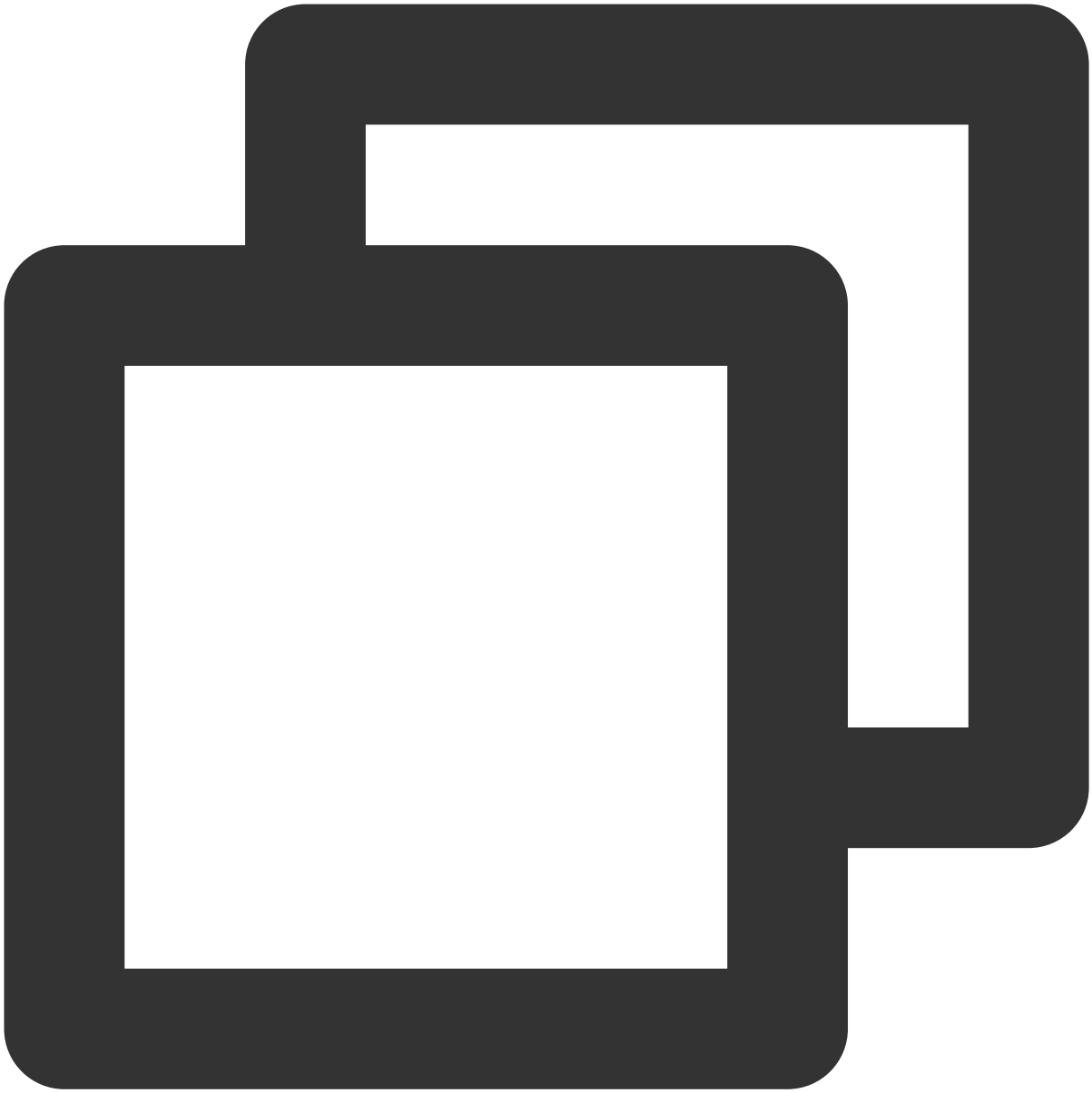
```
#: ./mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/admin
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> show dbs
admin   0.031GB
local   3.030GB
testdb  0.031GB
mongos> use testdb
switched to db testdb
mongos> show collections
system.indexes
testcollection
mongos> db.testcollection.find().limit(2)
```

Connection in different authentication methods

As described in the Connection Sample, TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. For those two authentication methods, the shell parameters are not the same. See below for more information.

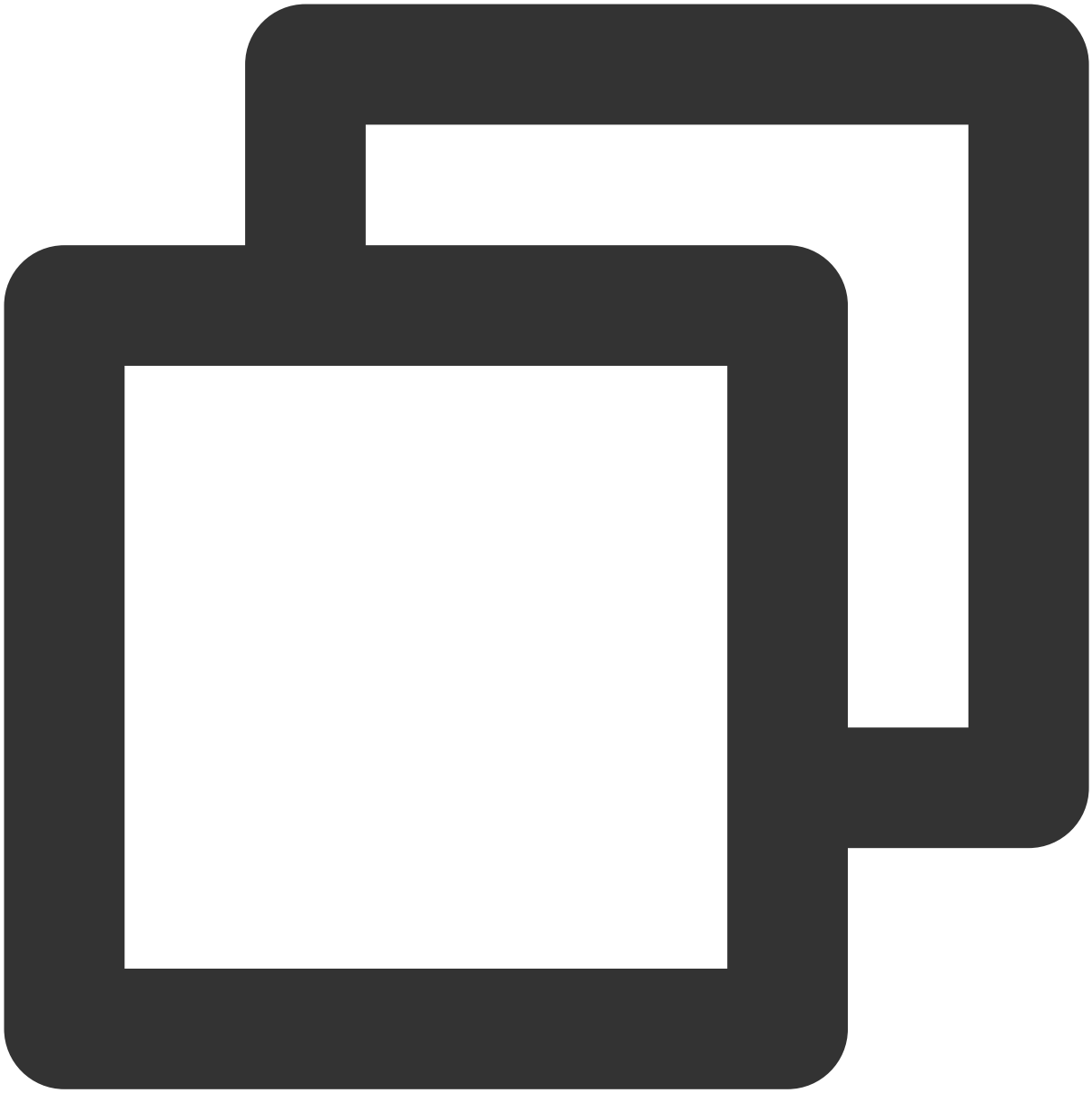
SCRAM-SHA-1 authentication (mongouser)

SCRAM-SHA-1 authentication is used for the default user `mongouser` and all new users created in the console. Shell connection parameters are the same as those described in Quick Start without additional parameters required. See the example below:



```
mongo 10.66.187.127:27017/admin -u mongouser -p thepasswordA1
```

If you want to enter a specific `db` directly such as "singer", after connecting to MongoDB, proceed as described below:



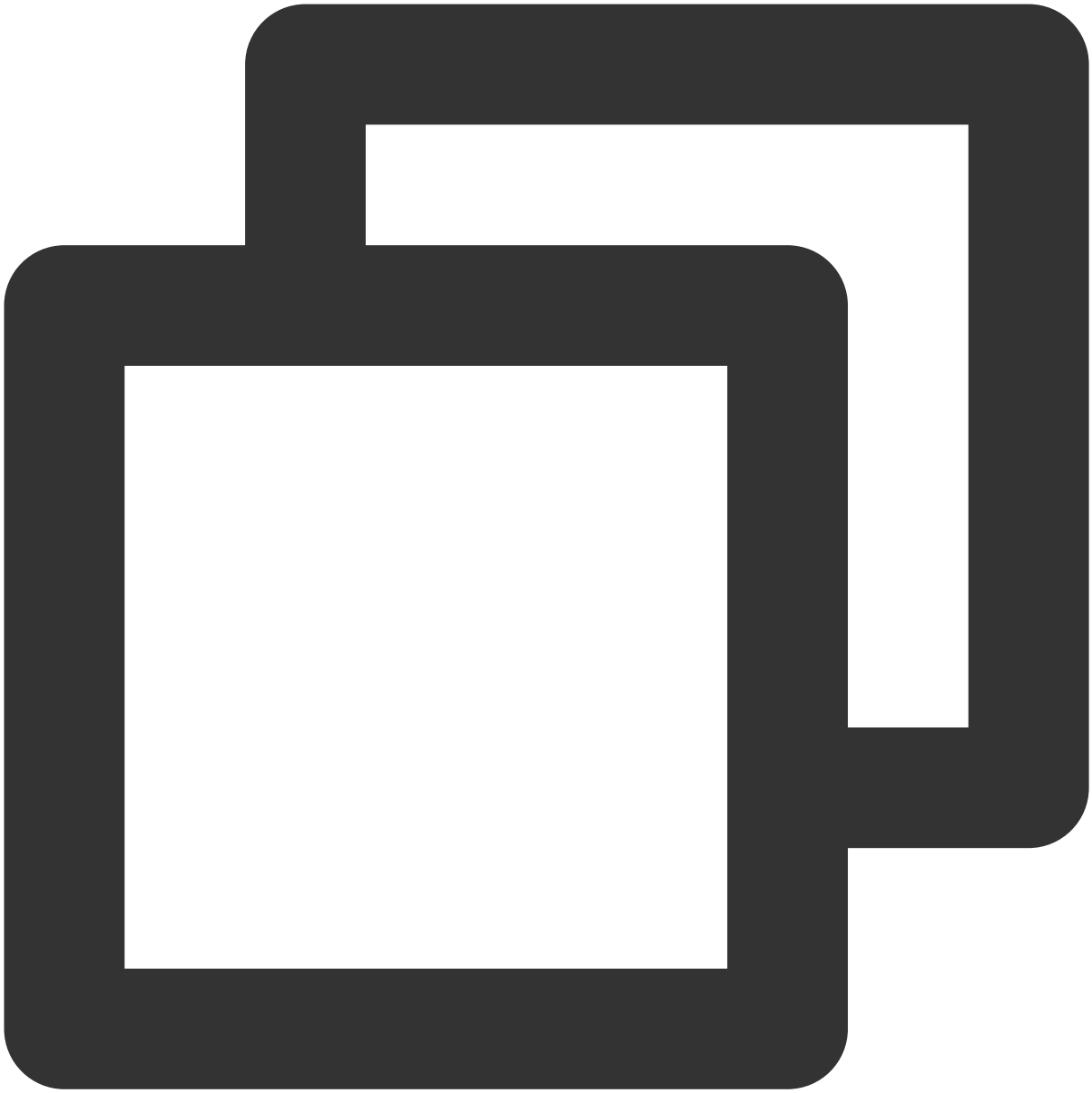
```
mongo 10.66.187.127:27017/singer -u mongouser -p thepasswordA1 --authenticationData
```

See the figure below:

```
#: ./mongo 10.66.187.127:27017/singer -u mongouser -p thepasswordA1 --authenticationDatabase admin
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/singer
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> db
singer
mongos>
```

MONGODB-CR authentication (rwuser)

Please note that MONGODB-CR authentication is used only for the default user `rwuser` , and the authentication method of MONGODB-CR should be expressly specified in the shell connection parameters. See the example below:

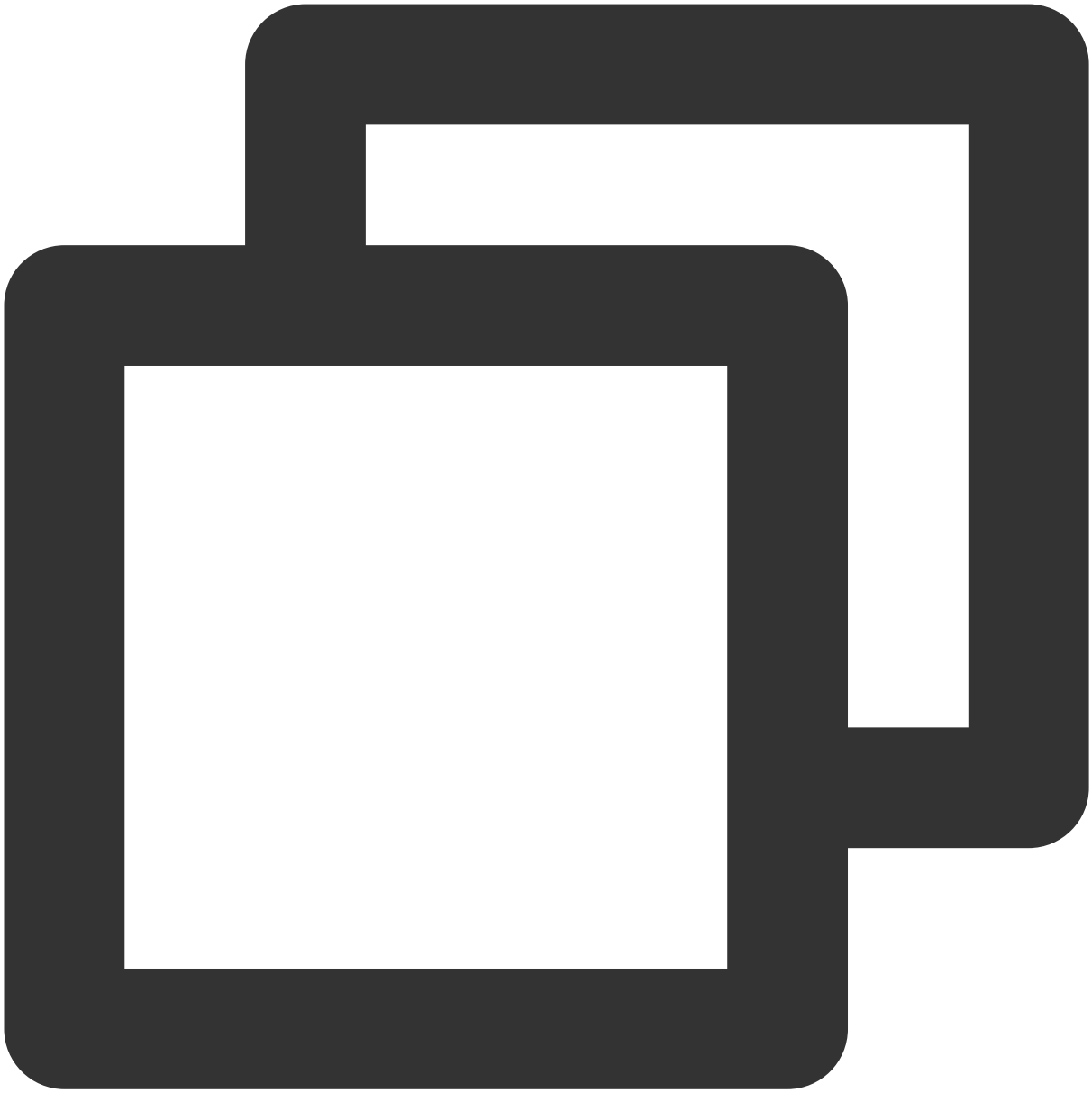


```
mongo 10.66.187.127:27017/admin -u rwuser -p thepasswordA1 --authenticationMechanis
```

See the figure below:

```
#: ./mongo 10.66.187.127:27017/admin -u rwuser -p thepasswordA1 --authenticationMechanism=MONGODB-CR
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/admin
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> show dbs
admin   0.031GB
local   3.030GB
testdb  0.031GB
mongos>
```

If you want to enter a specific `db` directly such as "singer", after connecting to MongoDB, proceed as described below:



```
mongo 10.66.187.127:27017/singer -u rwuser -p thepasswordA1 --authenticationMechani
```

See the figure below:

```
#: ./mongo 10.66.187.127:27017/singer -u rwuser -p thepasswordA1 --authenticationMechanism=MONGODB-CR --
authenticationDatabase admin
MongoDB shell version: 3.2.3
connecting to: 10.66.187.127:27017/singer
Server has startup warnings:
tencent cloud mongodb platform 2.0.4
mongos> db
singer
mongos>
```

Using shell to import and export data

For both authentication methods, you can use the shell to import and export data. For more information, please see Export and Import.

PHP Connection Sample

Last updated：2024-01-15 14:49:56

Notes

TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. The connecting URIs for the two authentication methods are formed differently. For more information, please see Connecting to TencentDB for MongoDB Instance.

For PHP, there is a driver that can be used to connect to and manipulate a MongoDB database, namely, MongoDB driver. The MongoDB driver is officially recommended by MongoDB, but it requires PHP 5.4 or above.

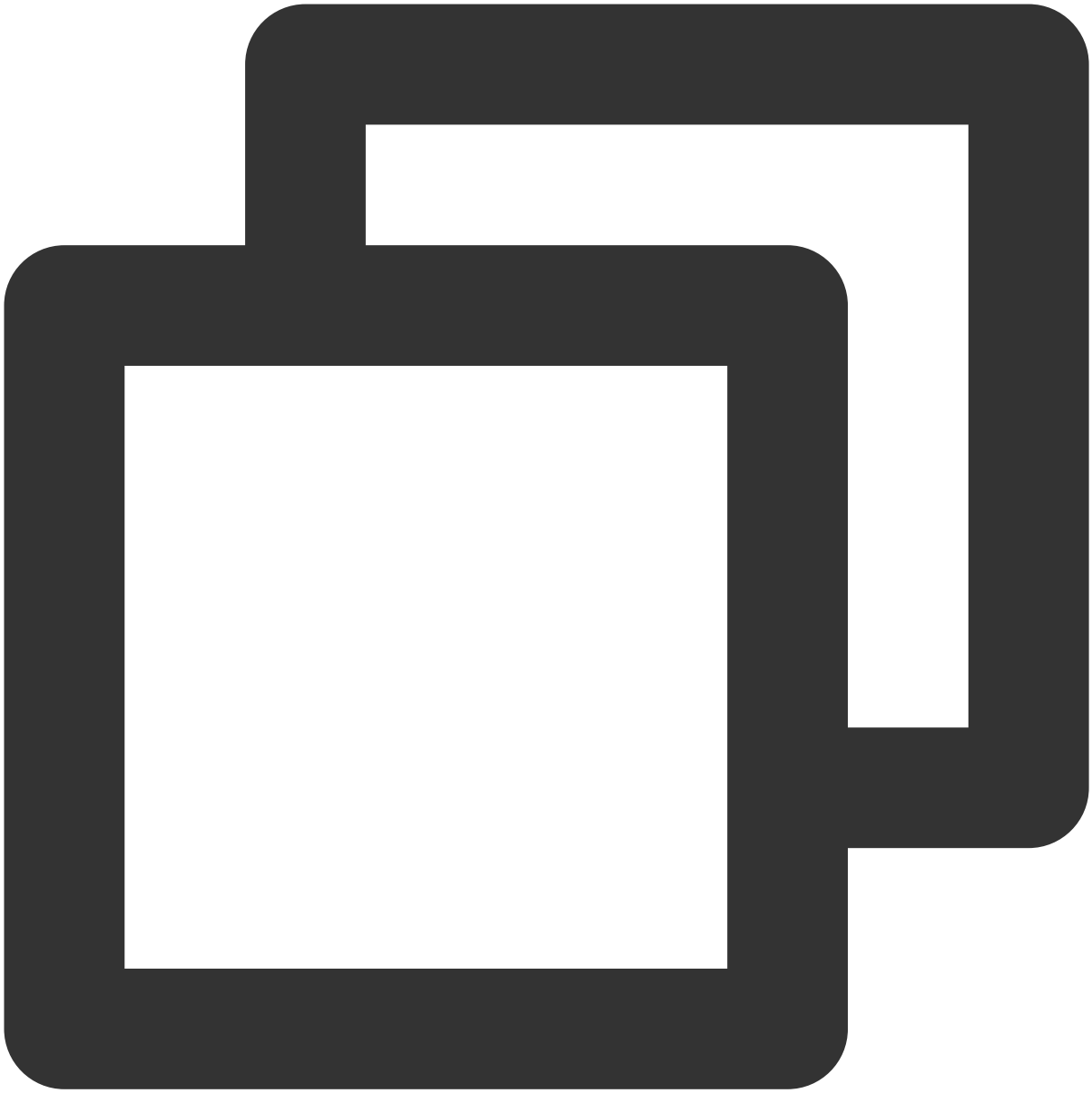
The following shows you how to connect to TencentDB for MongoDB and read/write data by using the aforementioned driver.

Using MongoDB driver

For more information on how to install the MongoDB driver, please see Installation.

The MongoDB driver can use both the MONGODB-CR and SCRAM-SHA-1 authentication methods. For more information, please see Connection Sample.

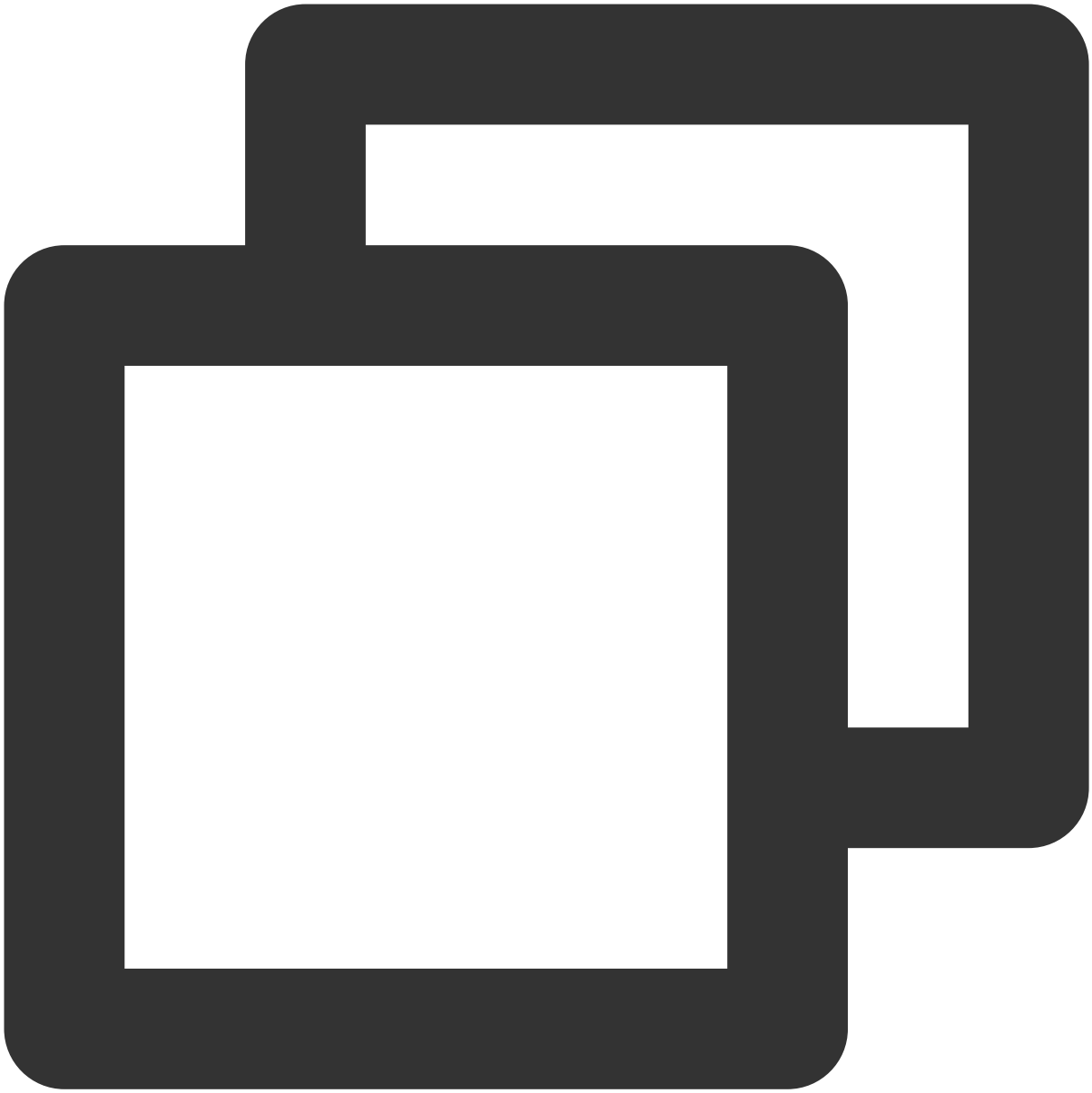
Sample code:



```
<?php
// Splice the connection URI
$uri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin';
$manager = new MongoDB\\Driver\\Manager($uri);

// Prepare to write data
$document1 = [
    'username' => 'lily',
    'age'      => 34,
    'email'    => 'lily@qq.com'
];
```

```
// Preprocess the data with the driver. Here, you can see that `_id` of MongoDB is
$bulk = new MongoDB\\Driver\\BulkWrite;
$_id1 = $bulk->insert($document1);

$result = $manager->executeBulkWrite('tsdb.table1', $bulk);

// You can also use the following code as needed to ensure that data is written to
// $writeConcern = new MongoDB\\Driver\\WriteConcern(MongoDB\\Driver\\WriteConcern:
// $result = $manager->executeBulkWrite('testdb.testcollection', $bulk, $writeConce

// Query
$filter = ['_id' => $_id1];
$query = new MongoDB\\Driver\\Query($filter);
$rows = $manager->executeQuery('tsdb.table1', $query); // You can also select to re
foreach($rows as $r){
   print_r($r);
}
```

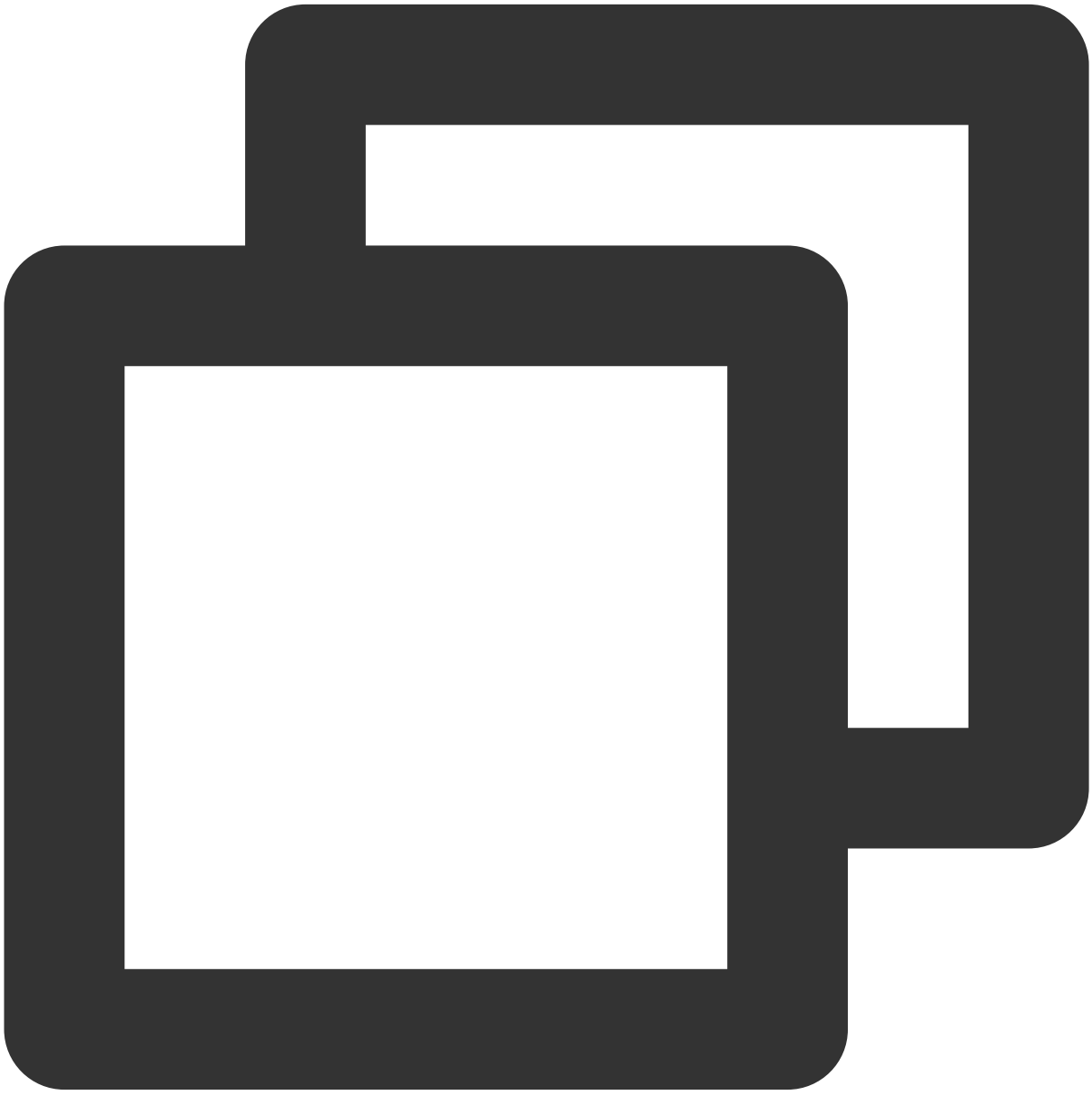
Output:



```
stdClass Object
(
    [_id] => MongoDB\\BSON\\ObjectID Object
        (
            [oid] => 582c001618c90a16363abc31
        )

    [username] => lily
    [age] => 34
    [email] => lily@qq.com
)
```

Using PHPLIB library (encapsulated based on MongoDB driver)

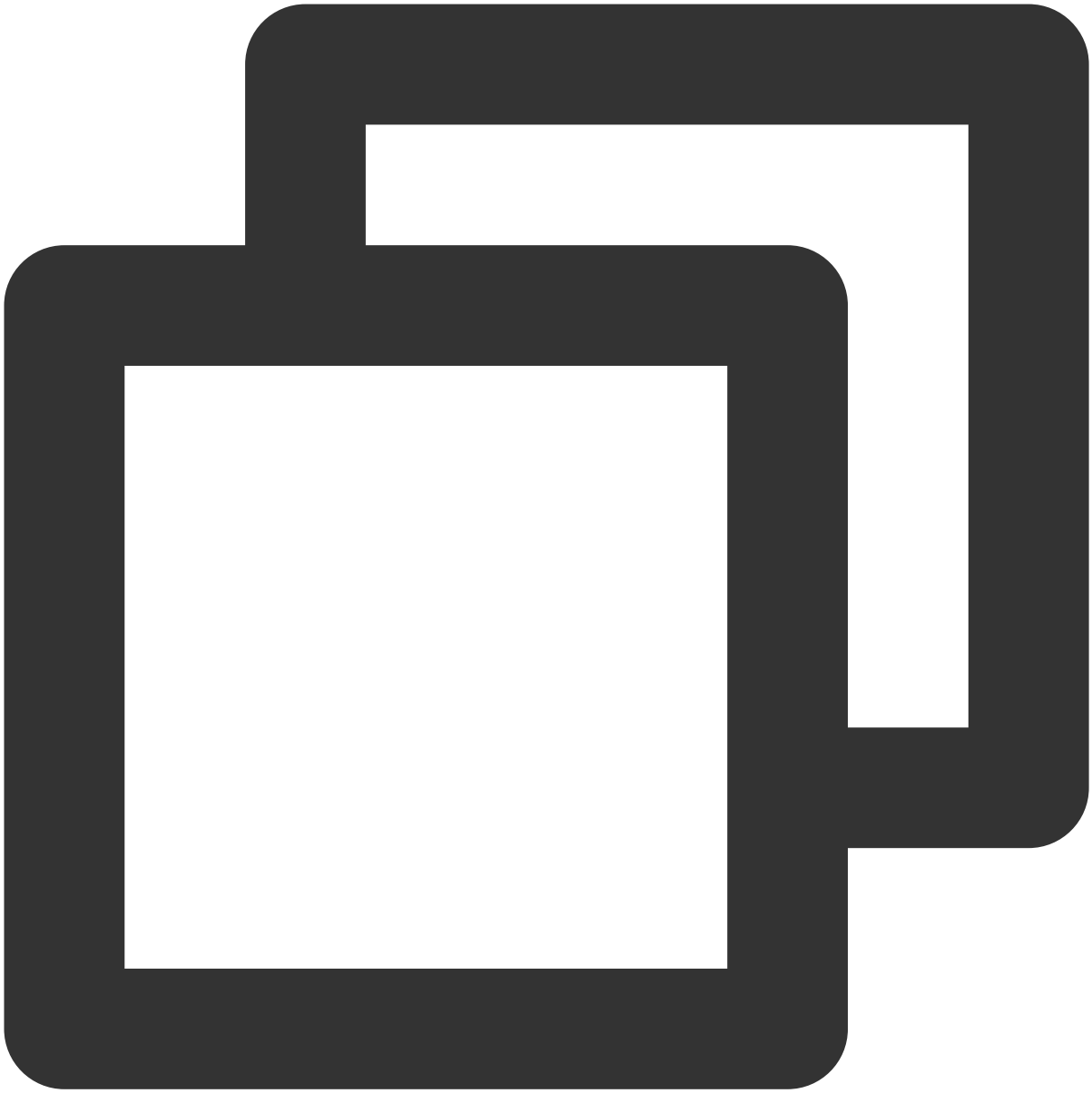
We recommend you use PHPLIB with the MongoDB driver. For more information, please see CRUD Operations.
For more information on how to install PHPLIB, please see Install the MongoDB PHP Library. Please note that PHPLIB depends on the MongoDB driver.
Sample code:



```
<?php
require_once __DIR__ . "/vendor/autoload.php";
```

```
// Initialize
$mongoClient = new MongoDB\\Client('mongodb://mongouser:thepasswordA1@10.66.187.127

// Use the `users` collection under the `demo` library
$collection = $mongoClient->demo->users;

// Write a data entry
$insertOneResult = $collection->insertOne(['name' => 'gomez']);

printf("Inserted %d document(s)\\n", $insertOneResult->getInsertedCount());
var_dump($insertOneResult->getInsertedId());

// Query data
$document = $collection->findOne(['name' => 'gomez']);

var_dump($document);
```

Output:



```
Inserted 1 document(s)
object(MongoDB\\BSON\\ObjectID)#11 (1) {
  ["oid"]=>
  string(24) "57e3bf20bf605714a53e69c1"
}
object(MongoDB\\Model\\BSONDocument)#16 (1) {
  ["storage":"ArrayObject":private]=>
  array(2) {
    ["_id"]=>
    object(MongoDB\\BSON\\ObjectID)#14 (1) {
      ["oid"]=>
```

```
        string(24) "57e3bf20bf605714a53e69c1"
      }
      ["name"]=>
      string(5) "gomez"
    }
}
```

Node.js Connection Sample

Last updated：2024-01-15 14:49:56

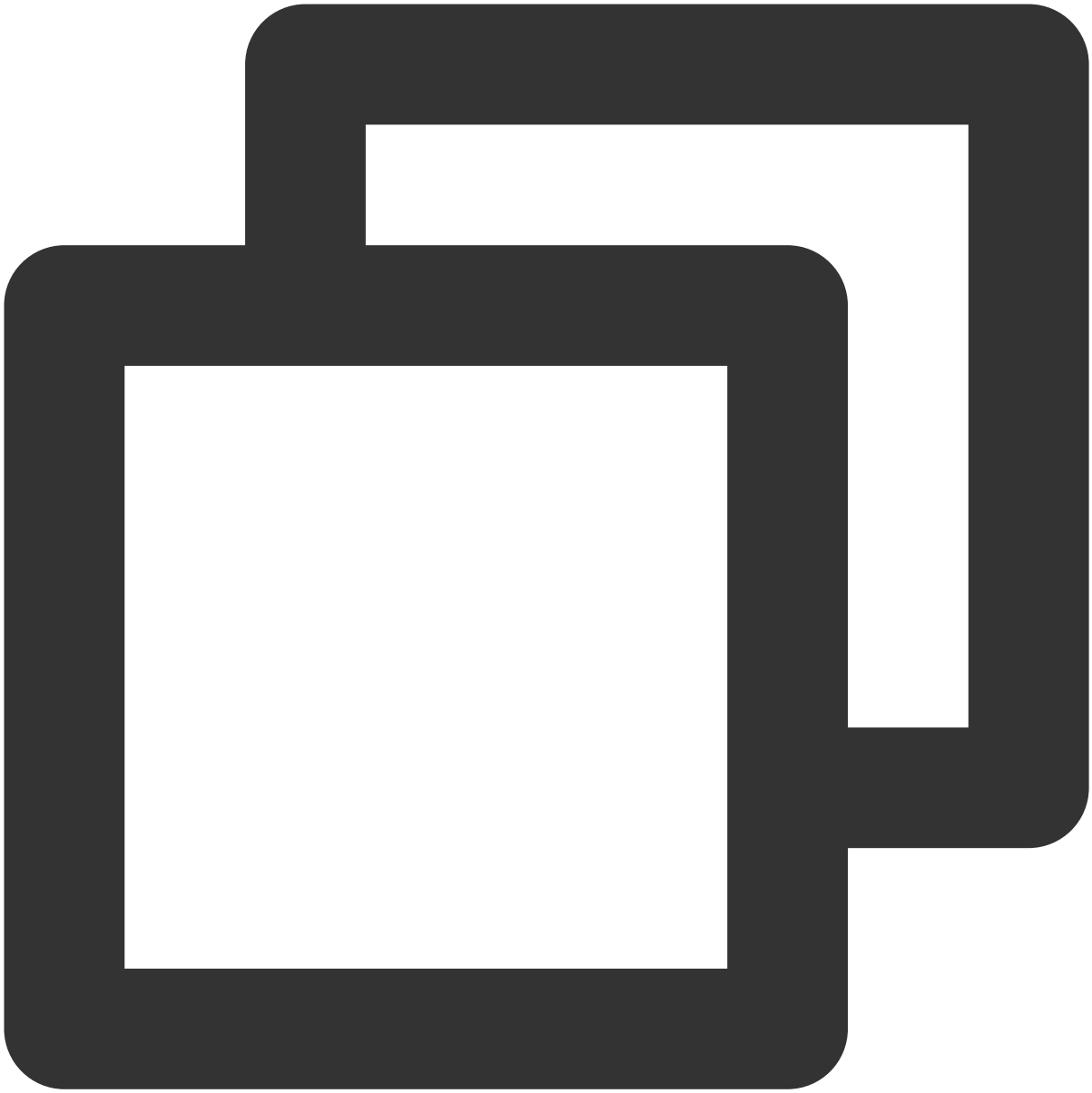
Notes

TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. The connecting URIs for the two authentication methods are formed differently. For more information, please see Connection Sample.
Documentation of MongoDB Node.js Driver

Quick Start

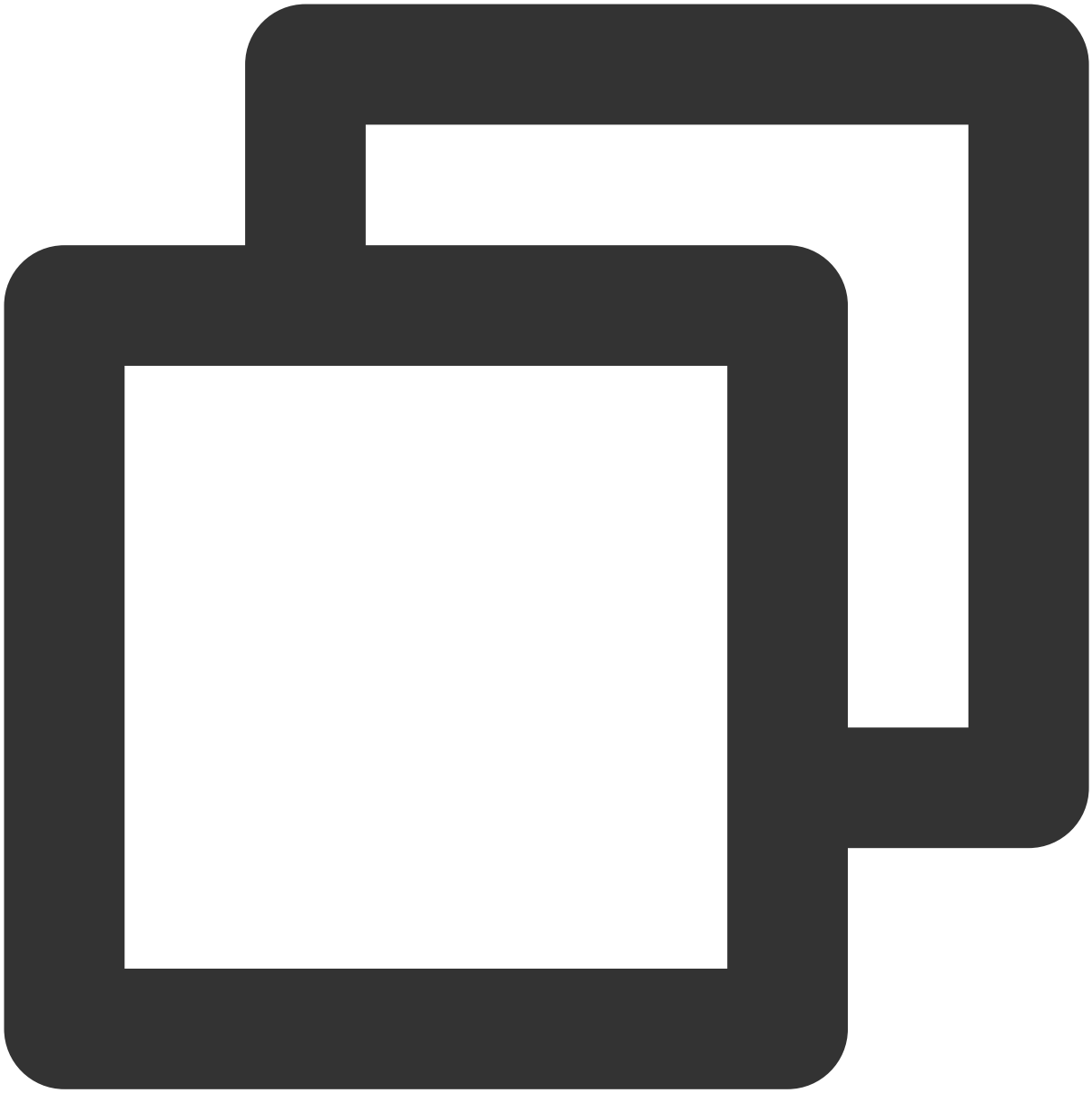
Native Node.js sample code

Install the driver package through shell:



```
npm install mongodb --save
(If the installation failed, you can try another source, such as `npm config set re
npm init
```

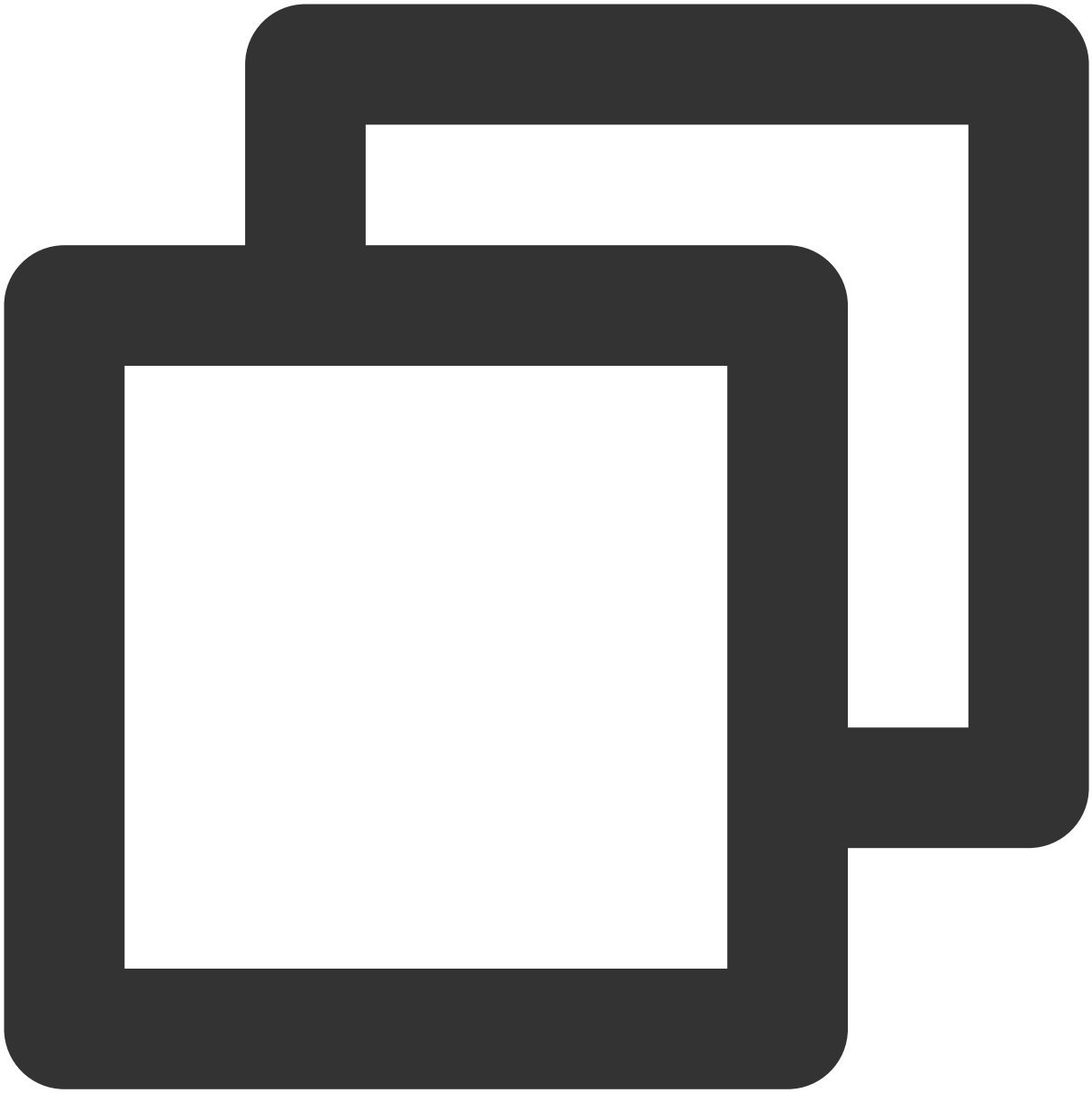
Program code:



```
'use strict';

var mongoClient = require('mongodb').MongoClient,
    assert = require('assert');

// Form the URI
var url = 'mongodb://mongouser:thepasswordA1@10.66.161.177:27017/admin';

mongoClient.connect(url, function(err, db) {
    assert.equal(null, err);
    var db = db.db('testdb'); // Select a database
```

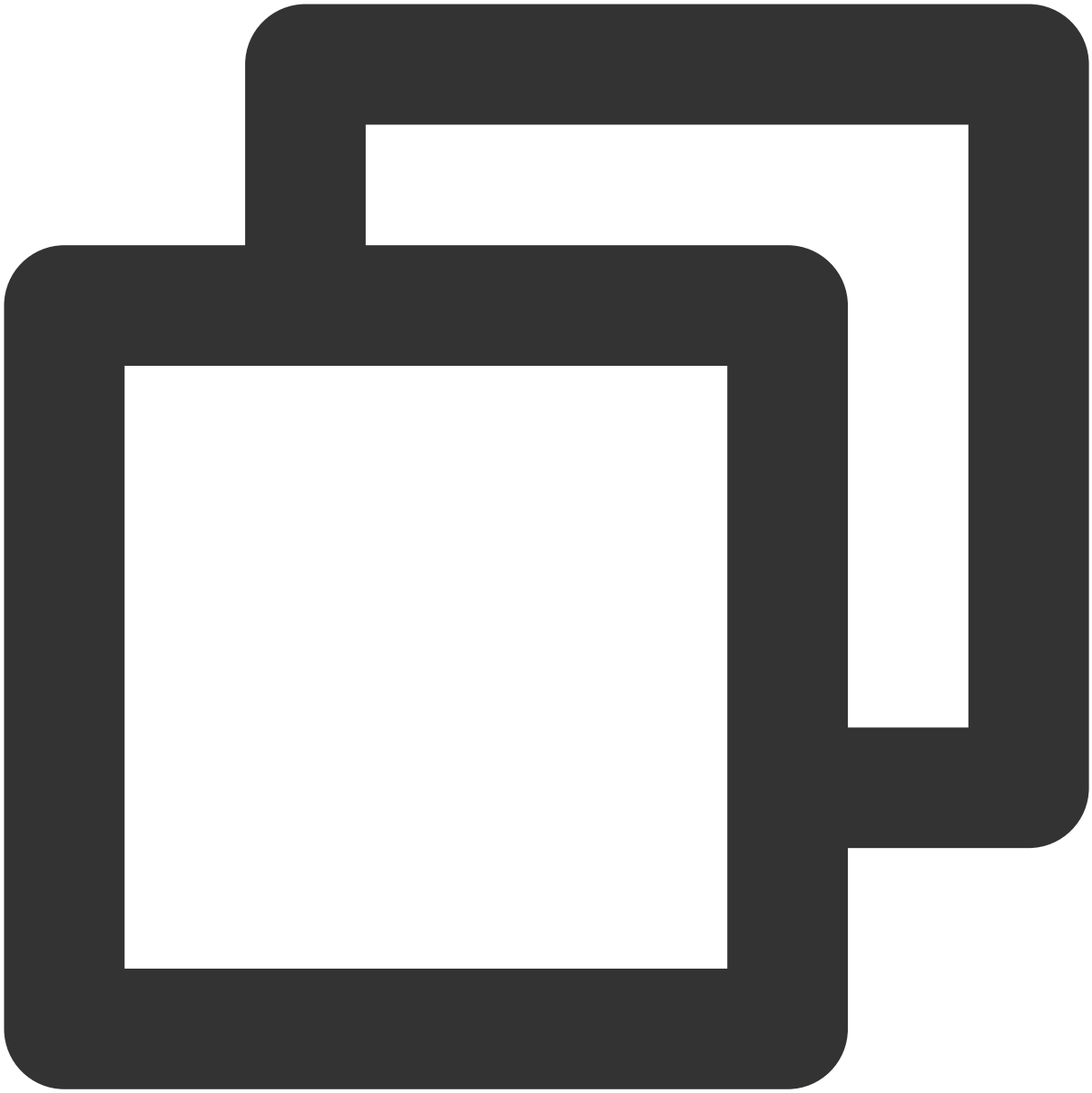
```
    var col = db.collection('demoCol'); // Select a collection (table)
    // Insert data
    col.insertOne(
        {
            a: 1,
            something: "yy"
        },
        // Optional parameters
        //{
        //    w: 'majority' // Enable the "Majority" mode to ensure that data are w
        //},
        function(err, r) {
            console.info("err:", err);
            assert.equal(null, err);
            // Assertion is written successfully
            assert.equal(1, r.insertedCount);
            // Query data
            col.find().toArray(function(err, docs) {
                assert.equal(null, err);
                console.info("docs:", docs);
                db.close();
            });
        }
    );
});
```

Output:



```
[root@VM_2_167_centos node]# node index.js
docs: [ { _id: 567a1bf26773935b3ff0b42a, a: 1, something: 'yy' } ]
```

Sample Code for Connecting to Node.js mongoose



```
var dbUri = "mongodb://" + user + ":" + password + "@" + host + ":" + port + "/" +
var opts = {
    auth: {
        authMechanism: 'MONGODB-CR', // This parameter is not required if SCRAM-SHA
        authSource: 'admin'
    }
};
var connection = mongoose.createConnection(dbUri, opts);
```

Java Connection Sample

Last updated：2024-01-15 14:49:55

Notes

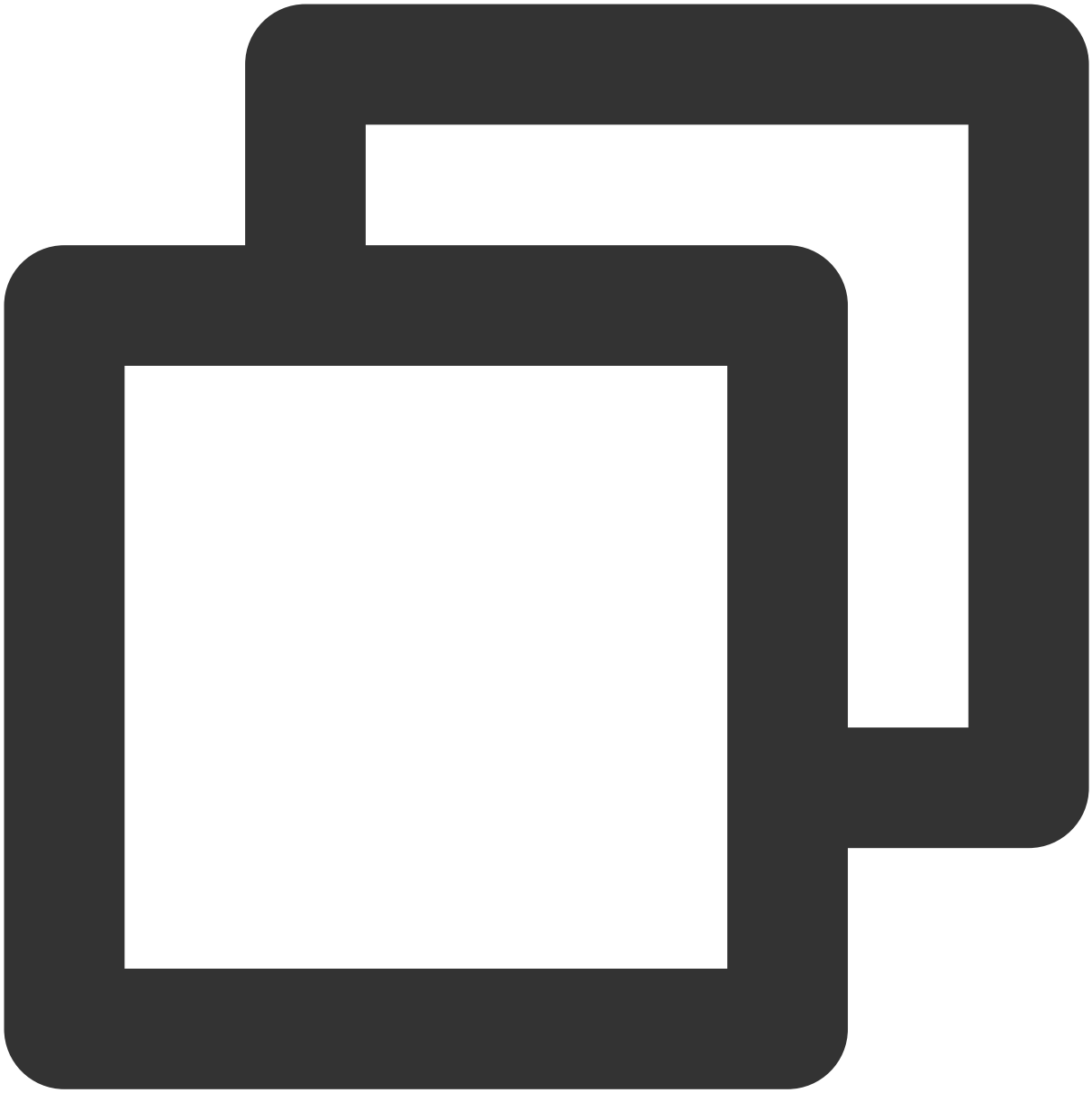
TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. The connecting URIs for the two authentication methods are formed differently. For more information, see Instance Connection.

Documentation of the MongoDB Java Driver

Download the JAR package and select a version above 3.2.

Getting Started

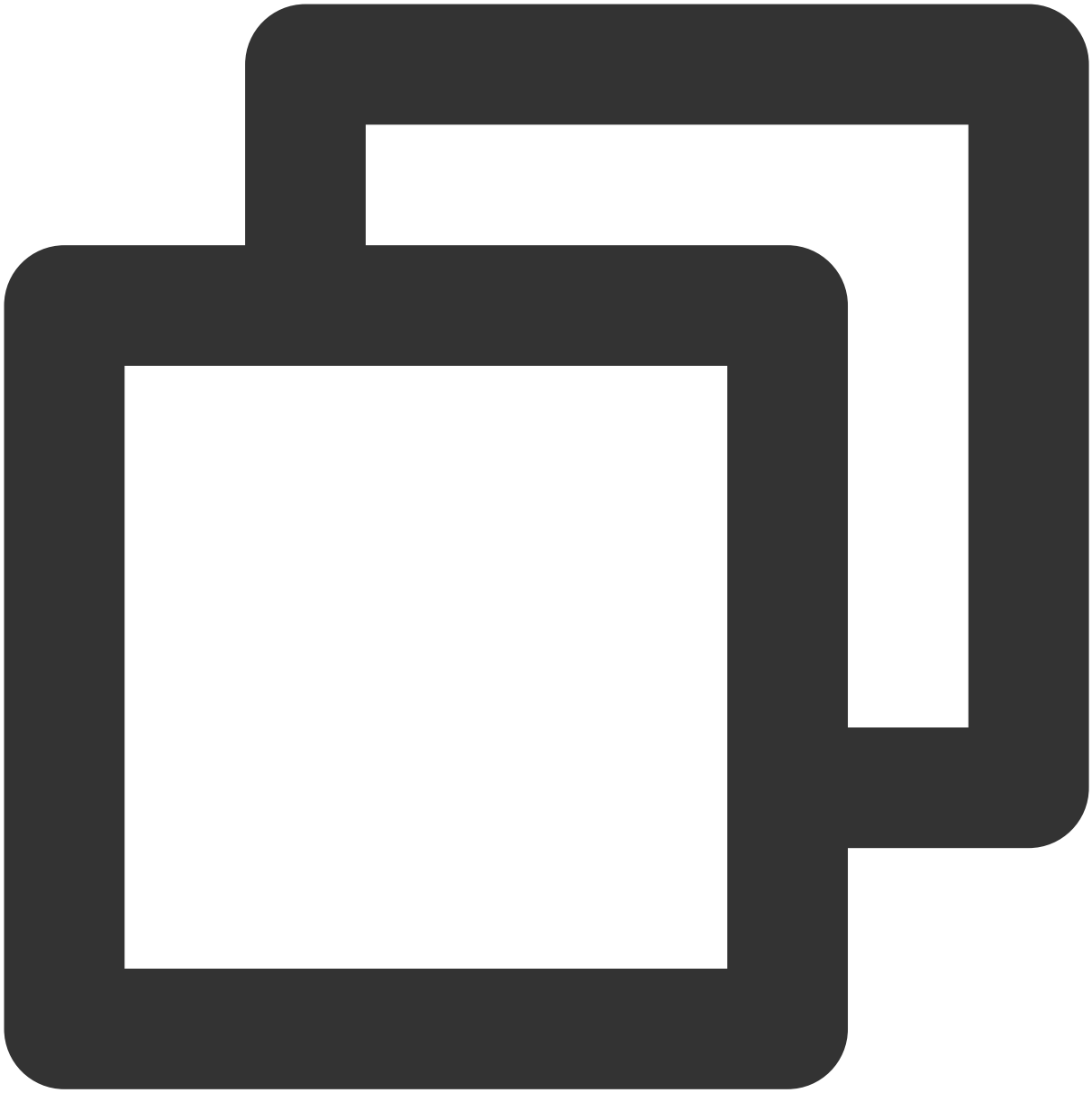
Native Java sample code



```
package mongodbdemo;

import org.bson.*;
import com.mongodb.*;
import com.mongodb.client.*;

public class MongodbDemo {

    public static void main(String[] args) {
        String mongoUri = "mongodb://mongouser:thepasswordA1@10.66.187.127:27017/ad
        MongoClientURI connStr = new MongoClientURI(mongoUri);
```

```
        MongoClient mongoClient = new MongoClient(connStr);
        try {
            // Use the database `someonedb`
            MongoDatabase database = mongoClient.getDatabase("someonedb");
            // Get the handle of the collection/table `someonetable`
            MongoCollection<Document> collection = database.getCollection("someonet

            // Prepare to write data
            Document doc = new Document();
            doc.append("key", "value");
            doc.append("username", "jack");
            doc.append("age", 31);

            // Write data
            collection.insertOne(doc);
            System.out.println("insert document: " + doc);

            // Read data
            BsonDocument filter = new BsonDocument();
            filter.append("username", new BsonString("jack"));
            MongoCursor<Document> cursor = collection.find(filter).iterator();
            while (cursor.hasNext()) {
                System.out.println("find document: " + cursor.next());
            }
        } finally {
            // Close the connection
            mongoClient.close();
        }
    }
}
```

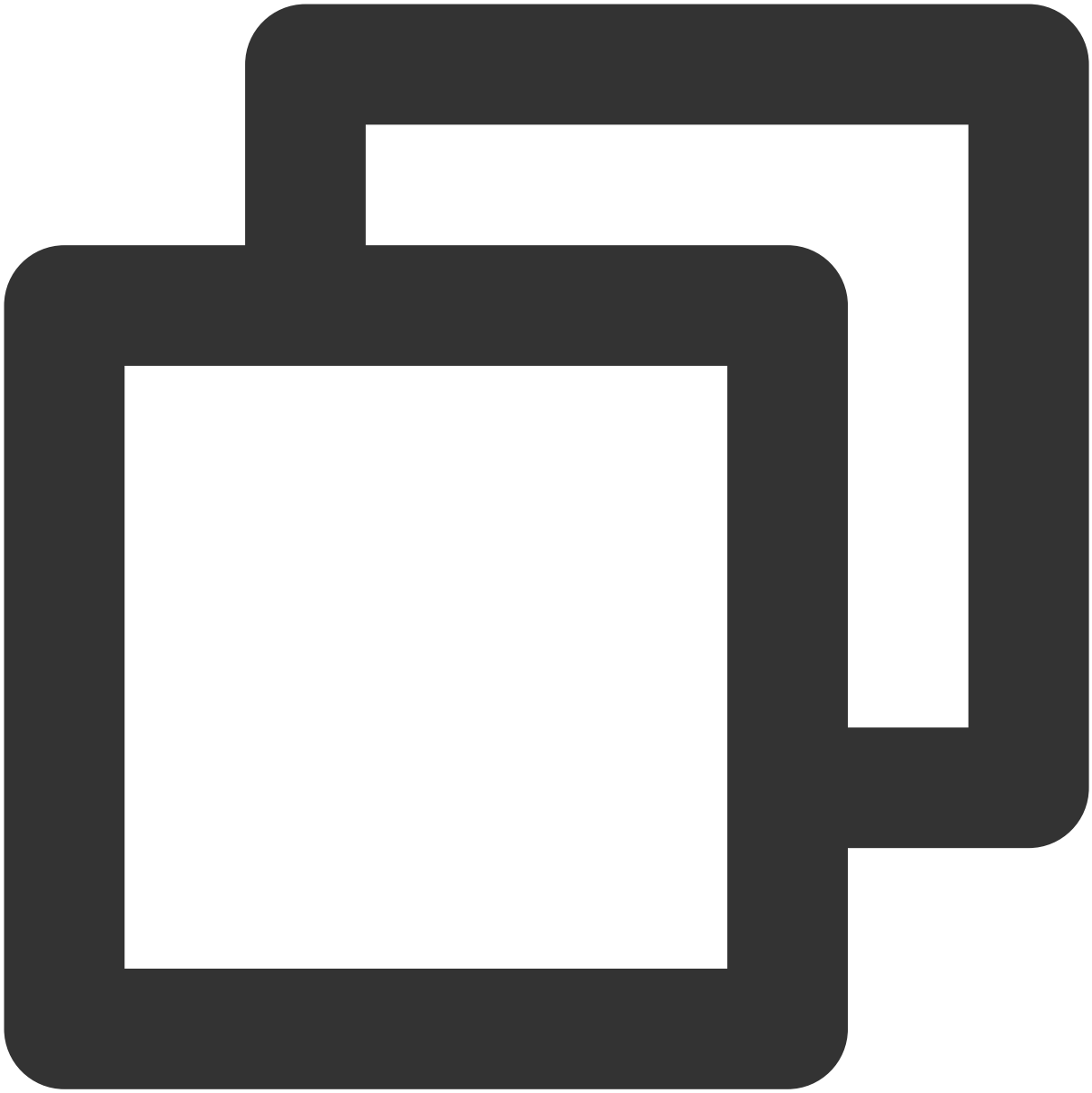
Output:



```
INFO: Opened connection [connectionId{localValue:2, serverValue:67621}] to 10.66.12
insert document: Document{{key=value, username=jack, age=31, _id=56a6ebb565b33b771f
find document: Document{{_id=56a3189565b33b2e7ca150ba, key=value, username=jack, ag
Jan 26, 2016 11:44:53 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:67621}] to 10.66.12
```

Configuration sample for Spring Data MongoDB

This sample demonstrates how to configure the authentication database admin, which depends on the versions of Spring and Spring Data MongoDB you use.



```
<bean id="mongoTemplate" class="org.springframework.data.mongodb.core.MongoTemplate
    <constructor-arg name="mongoDbFactory" ref="mongoDbFactory" />
</bean>
<bean id="mongoDbFactory" class="org.springframework.data.mongodb.core.SimpleMongoD
    <constructor-arg name="mongo" ref="mongo" />
    <constructor-arg name="databaseName" value="your target database" />
    <constructor-arg name="credentials" ref="userCredentials" />
    <constructor-arg name="authenticationDatabaseName" value="admin" />
</bean>
<bean id="userCredentials" class="org.springframework.data.authentication.UserCrede
    <constructor-arg name="username" value="username" />
```

```
    <constructor-arg name="password" value="password" />
</bean>
```

Python Connection Sample

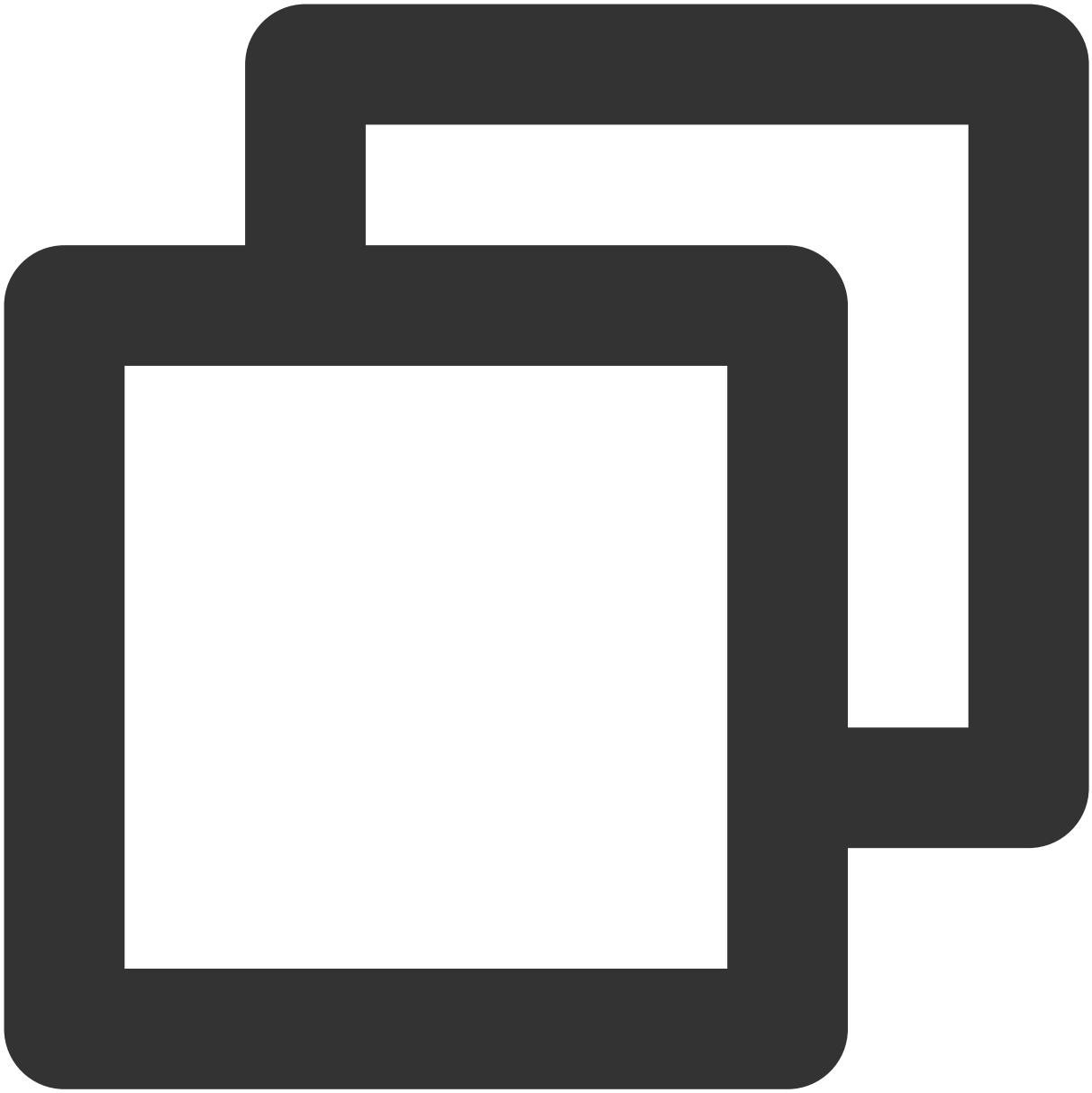
Last updated：2024-01-15 14:49:55

Notes

TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. The connecting URIs for the two authentication methods are formed differently. For more information, see Connecting to TencentDB for MongoDB Instance.
Download pymongo 3.13.0. For more information, see MongoDB Python Drivers.

Getting Started

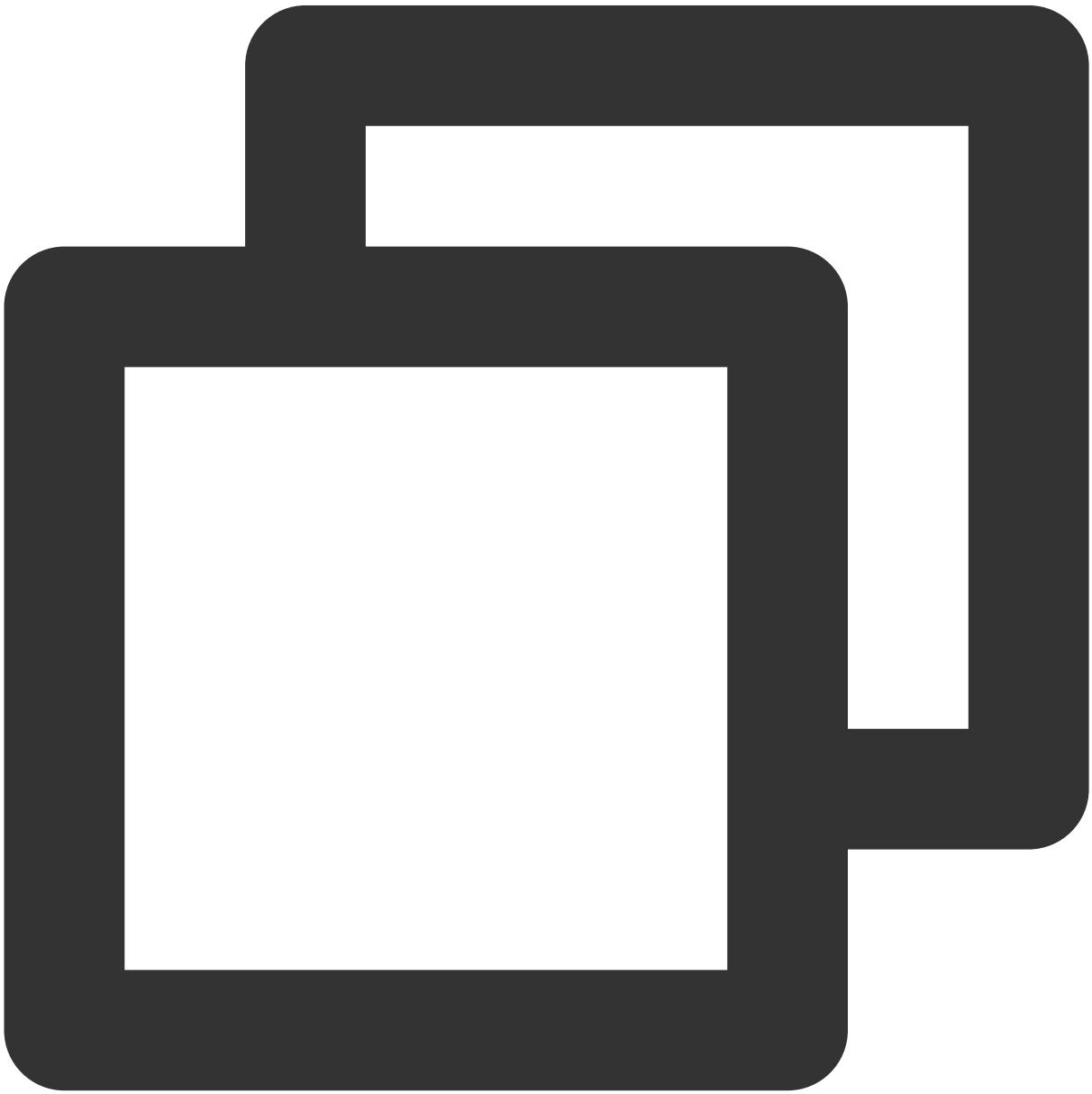
Python sample code 1



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

mongodbUri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin'

client = pymongo.MongoClient(mongodbUri)
db = client.somedb
db.user.drop()
element_num=10
for id in range(element_num):
```

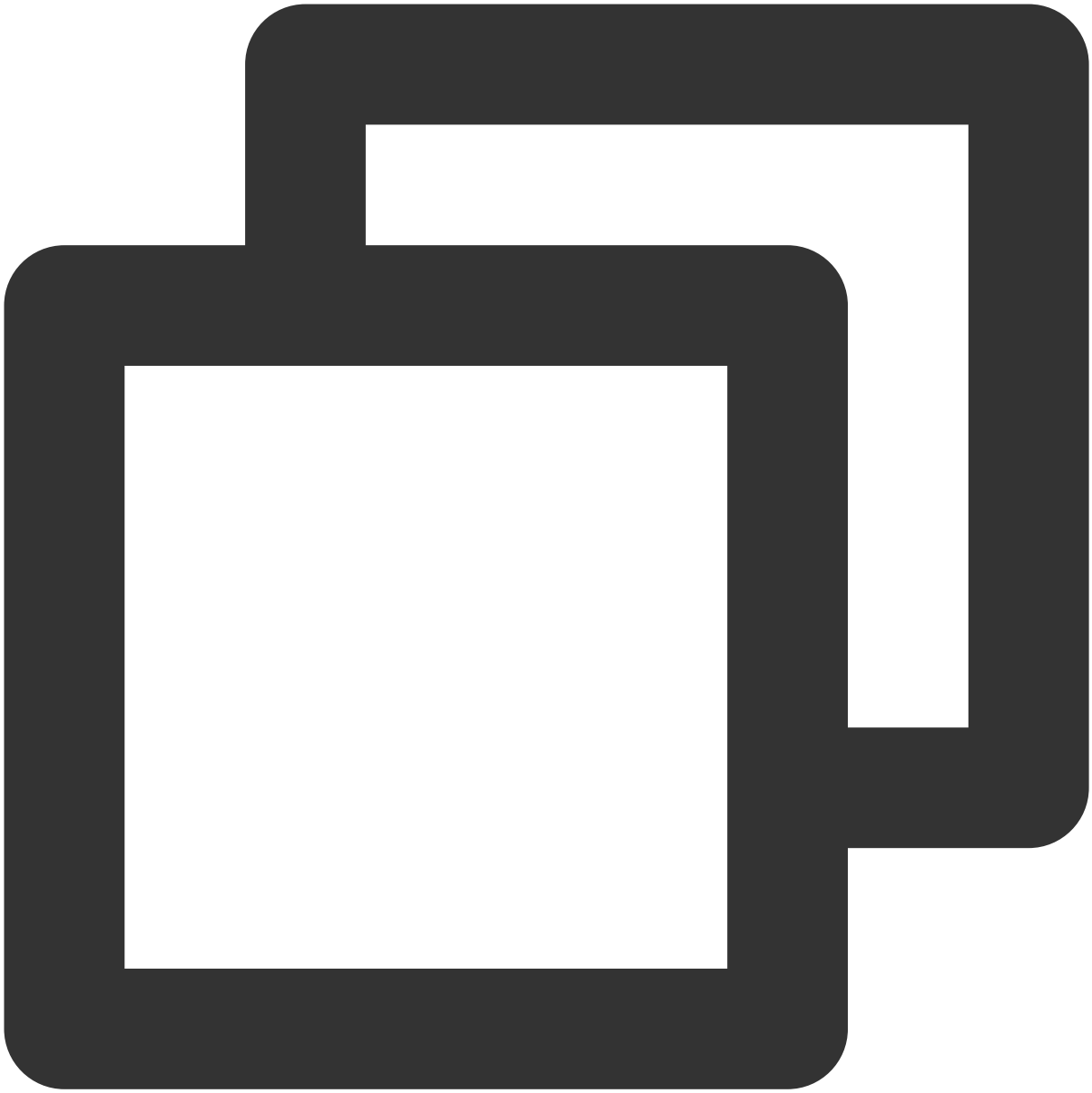
```
    name = random.choice(['R9','cat','owen','lee','J'])
    sex = random.choice(['male','female'])
    db.user.insert_one({'id':id, 'name':name, 'sex':sex})

content = db.user.find()
for i in content:
    print i
```

Python sample code 2



```
#!/usr/bin/python
import pymongo
mongodbUri = 'mongodb://mongouser:thepasswordA1@10.66.187.127:27017/admin'
client = pymongo.MongoClient(mongodbUri)
db = client.someonedb

inserted_id = db.somecoll.insert_one({"somekey":"yiqihapi"}).inserted_id
print inserted_id

for doc in db.somecoll.find(dict(_id=inserted_id)):
        print doc

for doc in db.somecoll.find({"somekey":"yiqihapi"}):
        print doc
```

The output information is as follows:

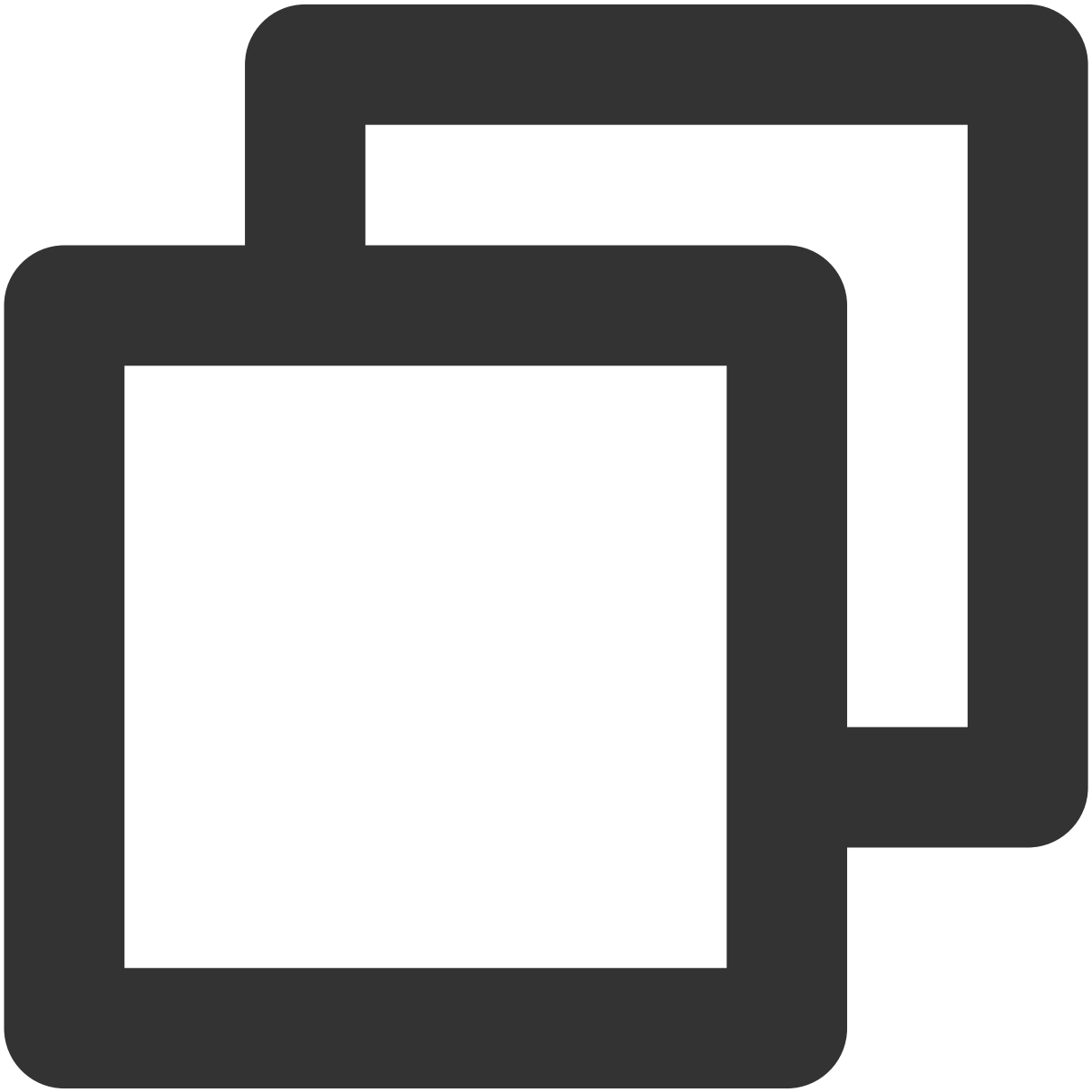


```
5734431e101e2f6d699b37ef
{u'somekey': u'yiqihapi', u'_id': ObjectId('5734431e101e2f6d699b37ef')}
{u'somekey': u'yiqihapi', u'_id': ObjectId('5734431e101e2f6d699b37ef')}
```

Python Read/Write Sample

Last updated：2024-01-15 14:49:55

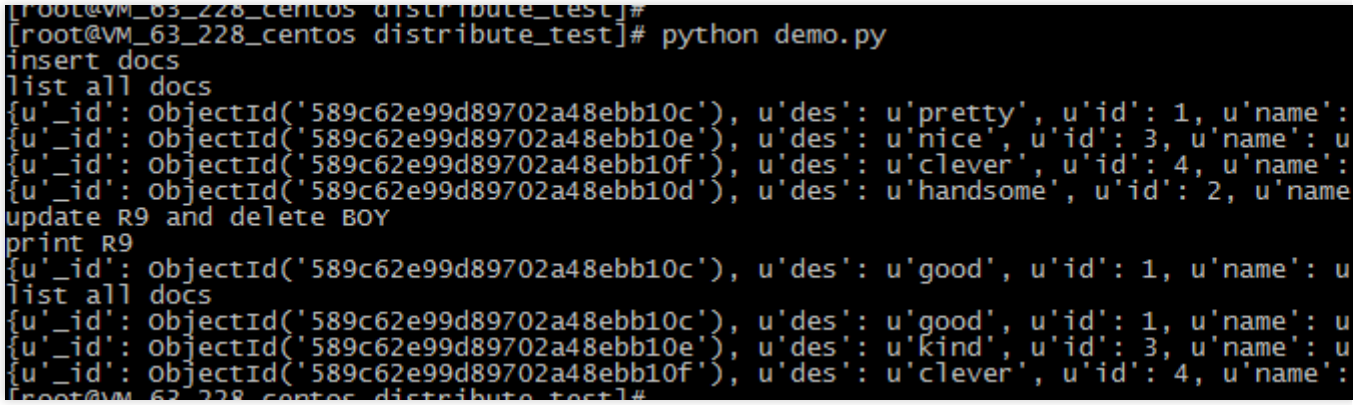
This document uses Python sample code to demonstrate the basic data read/write operations in a TencentDB for MongoDB sharded cluster. The following sample code is written based on Python 2.7 syntax, and the actual code may vary by version. For more information, see MongoDB Python Drivers.



```
#!/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 result:



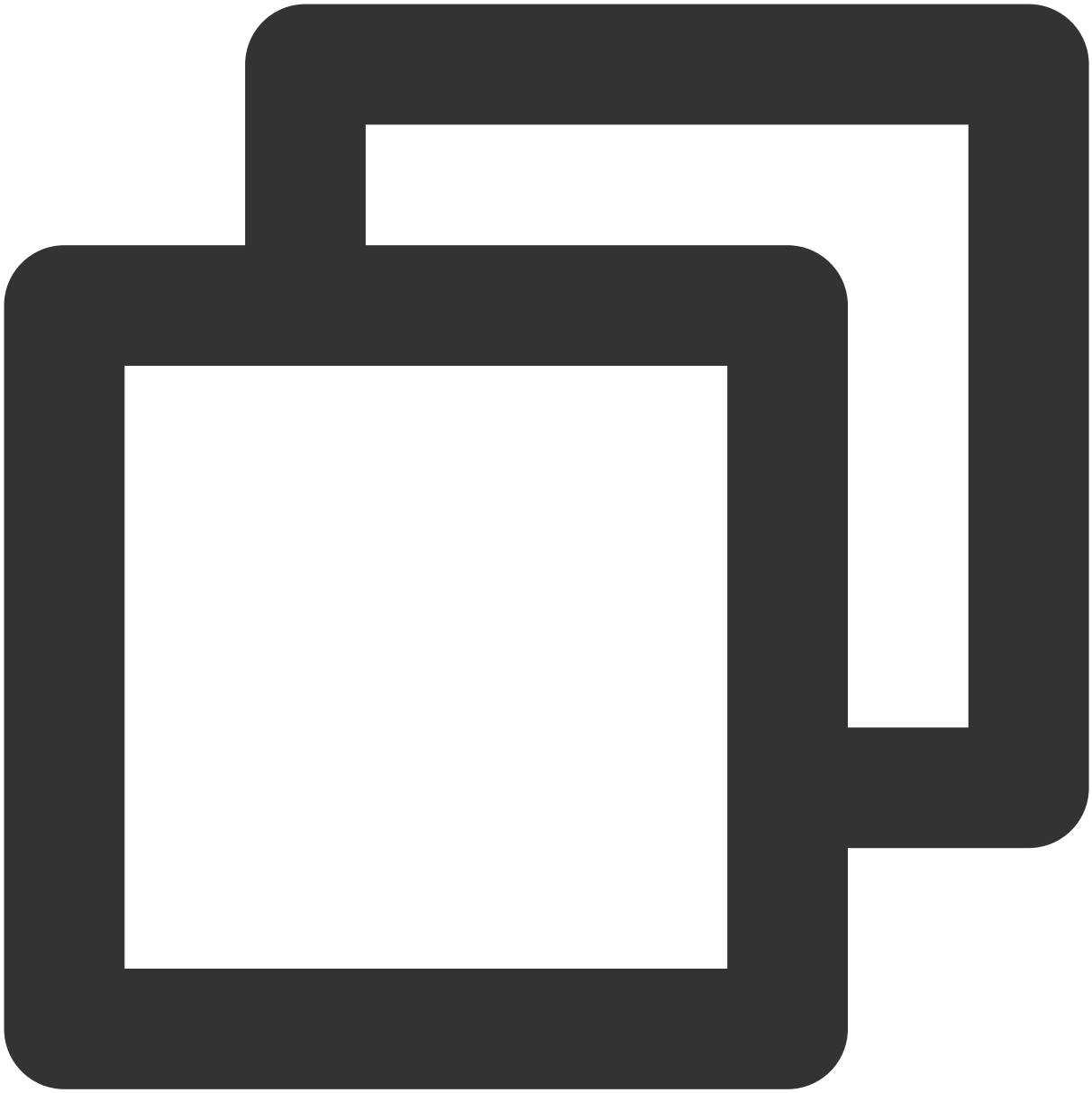
Go Connection Sample

Last updated：2024-01-15 14:49:56

Notes

TencentDB for MongoDB provides two usernames `rwuser` and `mongouser` by default to support the MONGODB-CR and SCRAM-SHA-1 authentication methods, respectively. The connecting URIs for the two authentication methods are formed differently. For more information, please see Connecting to TencentDB for MongoDB Instance.
Download mgo driver and MongoDB Go driver.

Sample Code for mgo



```
func GetMgoURL(ip, user, password string, port int) string {
    urlString := ""
    if user == "" && password == "" {
        urlString = fmt.Sprintf("mongodb://%s:%d/admin", ip, port)
    }else {
        urlString = fmt.Sprintf("mongodb://%s:%s@%s:%d/admin", url.QueryEscape(user
    }

    return urlString
}
```

```
url := service.GetMgoURL(reqPara.Ip, reqPara.User, reqPara.Password, reqPara.Port)
    session, err := mgo.Dial(url)
```

Sample Code for MongoDB Go

Please see MongoDB's official document.

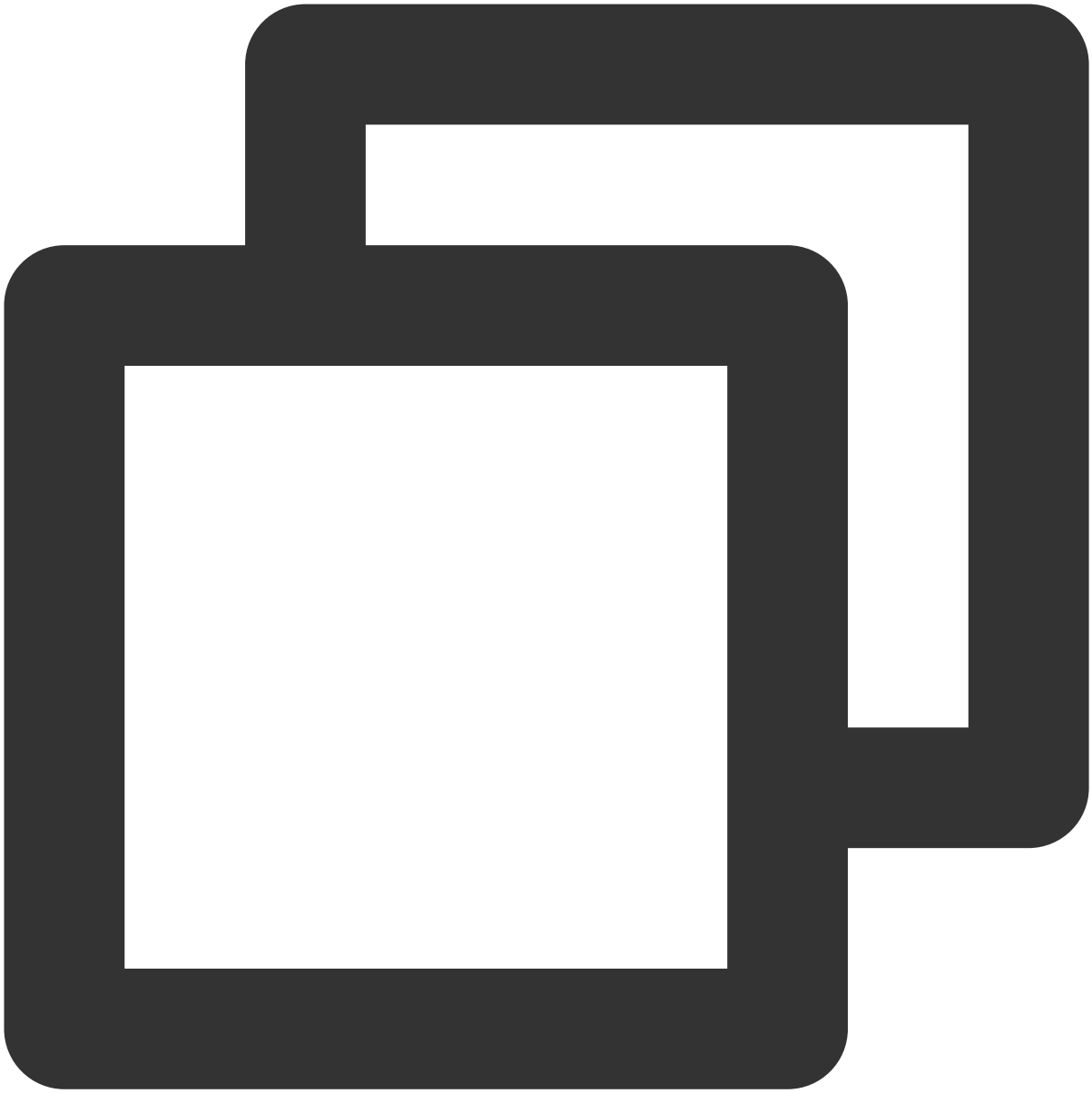
PHP Reconnection Sample

Last updated：2024-01-15 14:49:55

Notes

Instead of simply allowing you to access mongod, the 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 through the load balancer IP. If the connection is active for a long period of time, no intervention will be imposed on this status. However, if the persistent connection is inactive for more than one day (this period will be adjusted with version optimization), the route access layer will terminate the connection.
Generally, the client driver will implement an automatic reconnection process. However, this process cannot be implemented by certain language drivers. For such language drivers, if you attempt to communicate with the TencentDB for MongoDB service through a terminated connection, an error message such as "Remote server has closed the connection" will be returned, and manual reconnection will be required. The document provides 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=MONGO
    $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 ne
```

```
      // } catch( \\Exception $e ) {
            continue;
        }
        break;
    }
    return $connection;
}

$connection = getConnection();

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

        $one = $collection->findOne();

        var_dump($one);
}
```