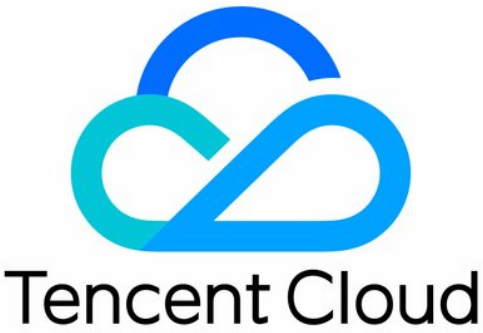
TencentDB for Tendis

Operation Guide

Product Documentation



©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2023 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

Tencent Cloud

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

Operation Guide
- Instance Connection Using Programming Languages
  - .Net Connection Sample
  - C Connection Sample
  - Go Connection Sample
  - Java Connection Sample
  - Node.js Connection Sample
  - PHP Connection Sample
  - Python Connection Sample
- Instance Maintenance and Management
  - Specifying Projects for Instances
  - Upgrading Instance Specification
  - Terminating Instances
- Monitoring Features
- Configuring Security Groups
- Disabling Commands

©2013-2022 Tencent Cloud. All rights reserved.

Operation Guide
Instance Connection Using Programming Languages
.Net Connection Sample

Last updated：2023-12-21 21:10:05

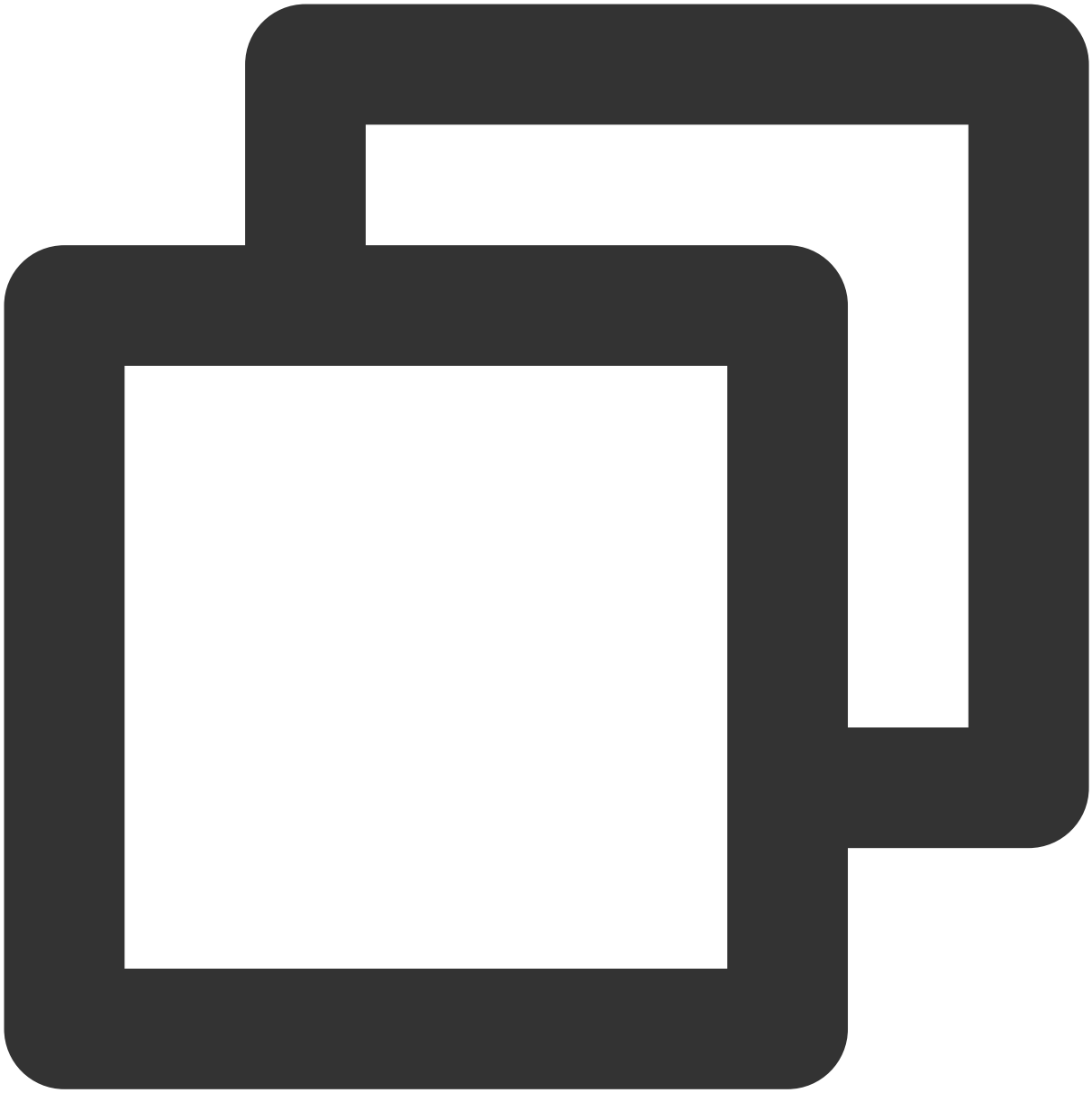
Preparations before running:

Download and install ServiceStack.Redis.

Sample code:

Do not use connection pool

©2013-2022 Tencent Cloud. All rights reserved.



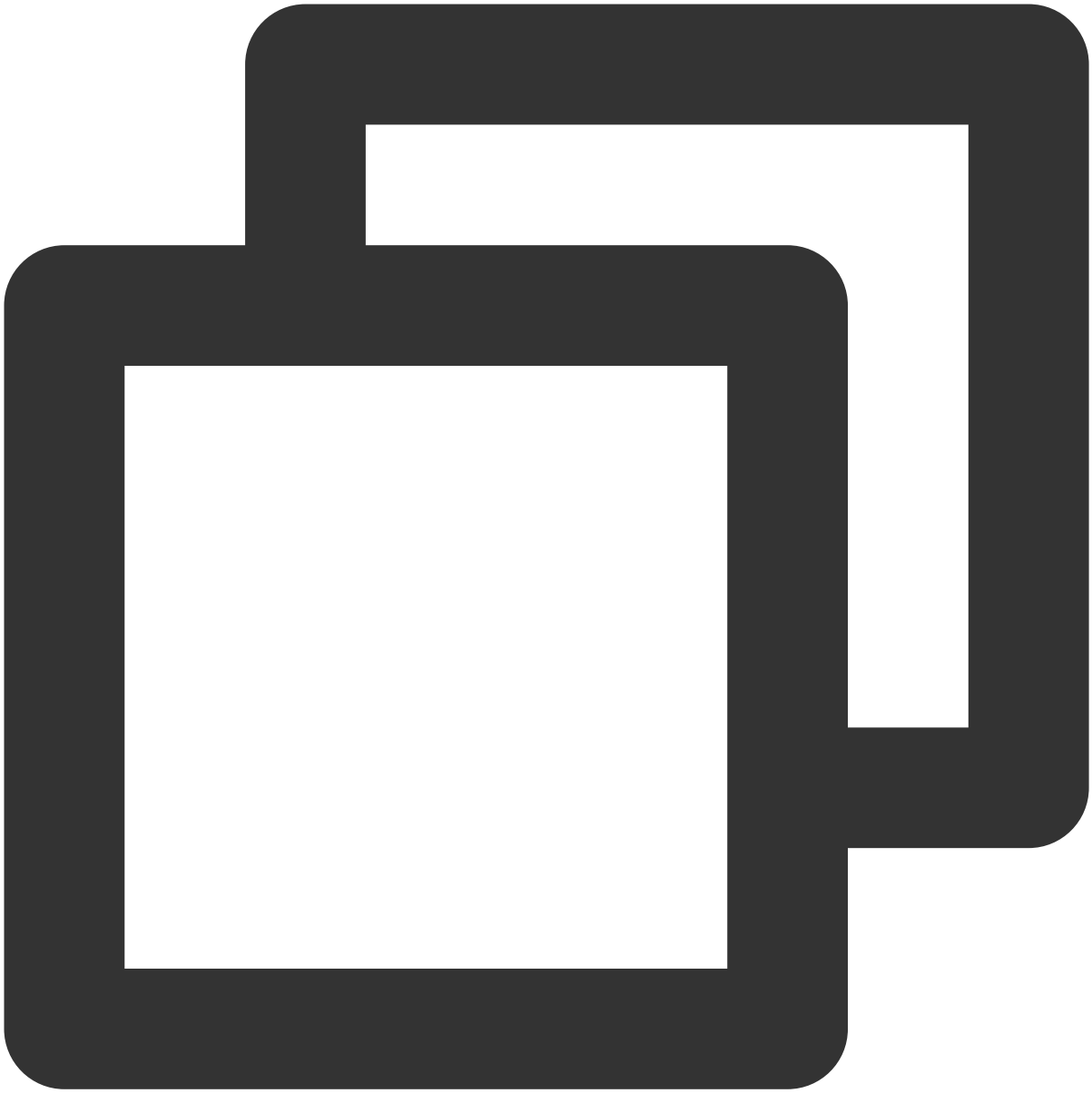
```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        {
            string host = "10.xx.xx.46";// The host address used to access the insta
            int port = 6379;// Port number
            string instanceId = "bd87dadc-8xx1-4xx1-86dd-021xxxcde96";// Instance ID
            string pass = "1234567q";// Password

            RedisClient redisClient = new RedisClient(host, port, instanceId + ":" +
            string key = "name";
            string value = "QcloudV5!";
            redisClient.Set(key, value); // Set the value
            System.Console.WriteLine("set key:[" + key + "]value:[" + value + "]");
            string getValue = System.Text.Encoding.Default.GetString(redisClient.Get
            System.Console.WriteLine("value:" + getValue);
            System.Console.Read();
           }
      }
 }
```

Use ServiceStack 4.0 connection pool

©2013-2022 Tencent Cloud. All rights reserved.



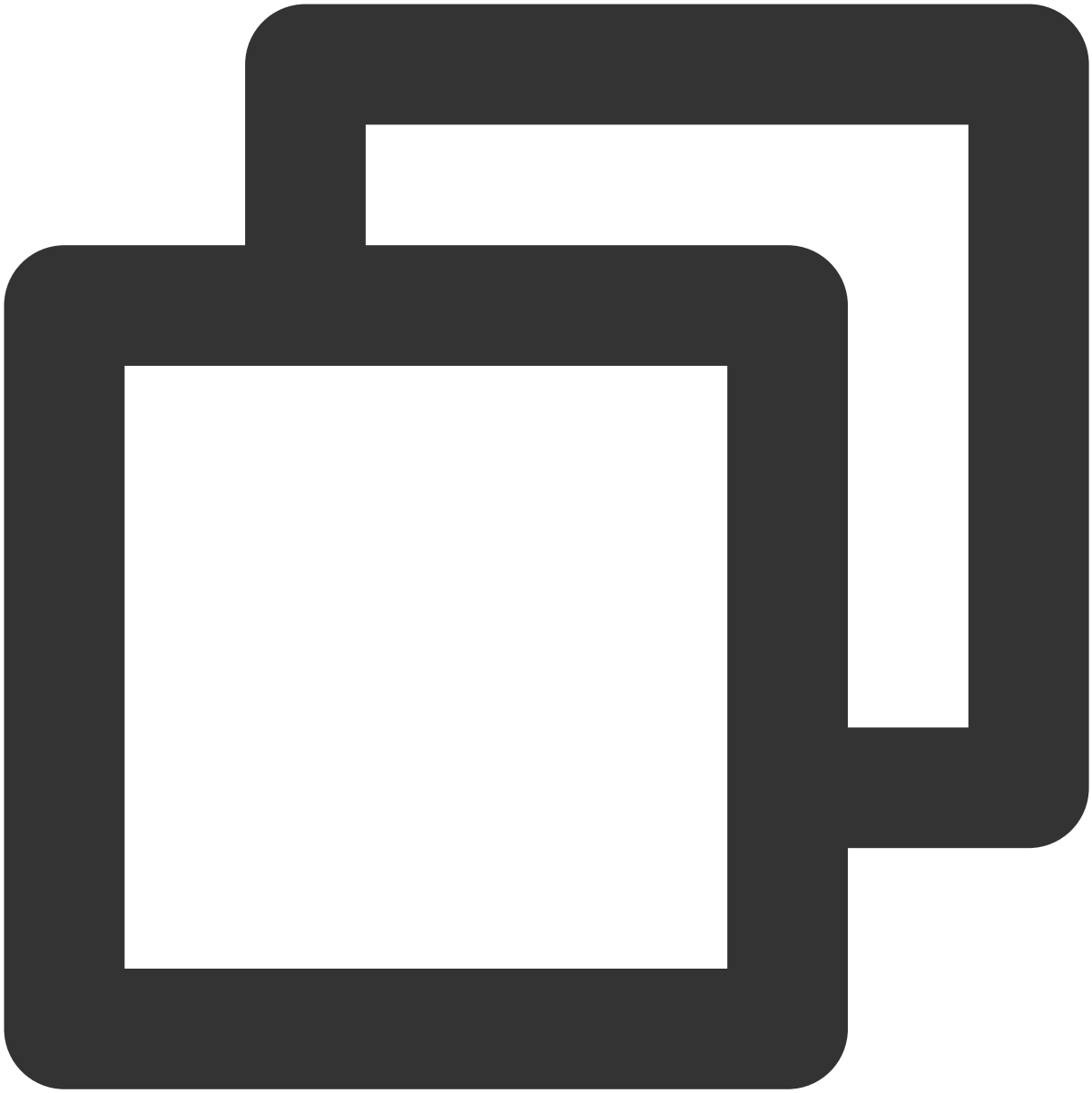
```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        {
             string[] testReadWriteHosts = new[] {
             "redis://:fb92bxxxabf11e5:1234xx8a1A@10.x.x.1:6379" /*redis://:instanc
             };
             RedisConfig.VerifyMasterConnections = false;// Need to be set
             PooledRedisClientManager redisPoolManager = new PooledRedisClientManag
             10/*connection pool timeout period*/, testReadWriteHosts);
             for (int i = 0; i < 100; i++)
            {
                 IRedisClient redisClient = redisPoolManager.GetClient();// Get the
                 RedisNativeClient redisNativeClient = (RedisNativeClient)redisClie
                 redisNativeClient.Client = null;// Need to be set
                 try
                {
                   string key = "test1111";
                   string value = "test1111";
                   redisClient.Set(key, value);
                   redisClient.Dispose();//
                }
                catch (Exception e)
                {
                    System.Console.WriteLine(e.Message);
                }
            }
            System.Console.Read();
         }
     }
}
```

Use ServiceStack 3.0 connection pool

©2013-2022 Tencent Cloud. All rights reserved.



```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ServiceStack.Redis;
using System;

namespace ConsoleApplication3
{
  class Program
  {
     static void Main(string[] args)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    {
            string[] testReadWriteHosts = new[] {
                "fb92bfxxbf11e5:123456xx1A@10.x.x.1:6379" /*instance ID:password@acc
                };
                PooledRedisClientManager redisPoolManager = new PooledRedisClientMan
                quantity*/, 10/*connection pool timeout period*/, testReadWriteHosts
                for (int i = 0; i < 100; i++)
               {
                IRedisClient redisClient = redisPoolManager.GetClient();// Get the c
                try
               {
                    string key = "test1111";
                    string value = "test1111";
                    redisClient.Set(key, value);
                    redisClient.Dispose();//
                }
                catch (Exception e)
              {
                     System.Console.WriteLine(e.Message);
              }
           }
           System.Console.Read();
       }
   }
}
```

Execution results:

©2013-2022 Tencent Cloud. All rights reserved.

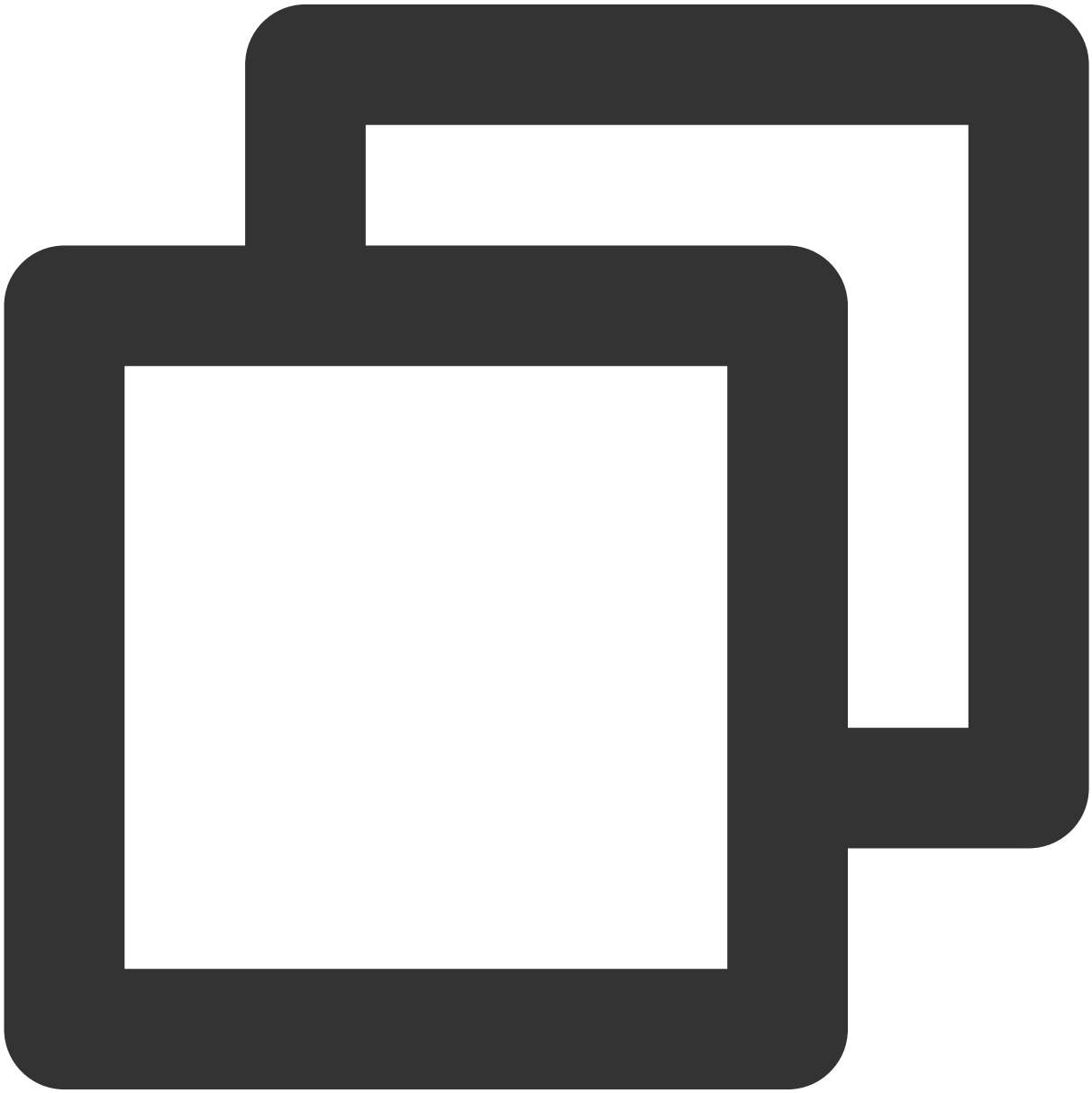
C Connection Sample

Last updated：2023-12-21 21:10:31

Preparations before running:

Download and install hiredis.

Sample code:



```
#include <stdio.h>
#include <stdlib.h>
```

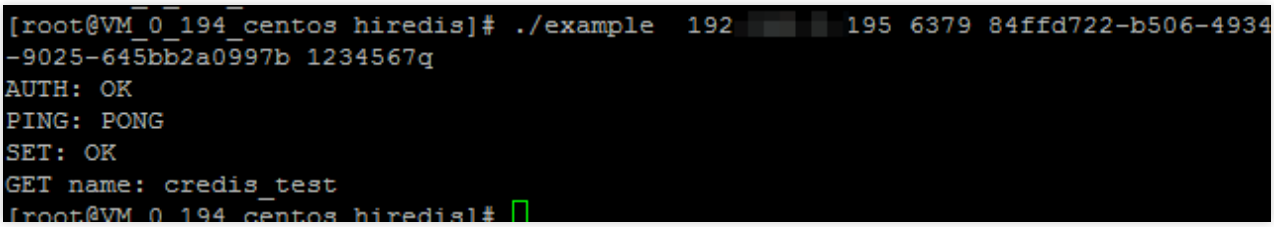
©2013-2022 Tencent Cloud. All rights reserved.

```
#include <string.h>

#include <hiredis.h>

int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;

if (argc < 4) {
      printf("Usage: 192.xx.xx.195 6379 instance_id password\\n");
      exit(0);
    }
    const char *hostname = argv[1];
    const int port = atoi(argv[2]);
    const char *instance_id = argv[3];
    const char *password    = argv[4];


    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    c = redisConnectWithTimeout(hostname, port, timeout);
    if (c == NULL || c->err) {
        if (c) {
            printf("Connection error: %s\\n", c->errstr);
            redisFree(c);
        } else {
            printf("Connection error: can't allocate redis context\\n");
        }
        exit(1);
    }

    /* AUTH */
    reply = redisCommand(c, "AUTH %s", password);
    printf("AUTH: %s\\n", reply->str);
    freeReplyObject(reply);

    /* PING server */
    reply = redisCommand(c,"PING");
    printf("PING: %s\\n", reply->str);
    freeReplyObject(reply);

    /* Set a key */
    reply = redisCommand(c,"SET %s %s", "name", "credis_test");
    printf("SET: %s\\n", reply->str);
    freeReplyObject(reply);

    /* Try a GET  */
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    reply = redisCommand(c,"GET name");
    printf("GET name: %s\\n", reply->str);
    freeReplyObject(reply);

    /* Disconnects and frees the context */
    redisFree(c);

    return 0;
}
```

Execution results:



©2013-2022 Tencent Cloud. All rights reserved.

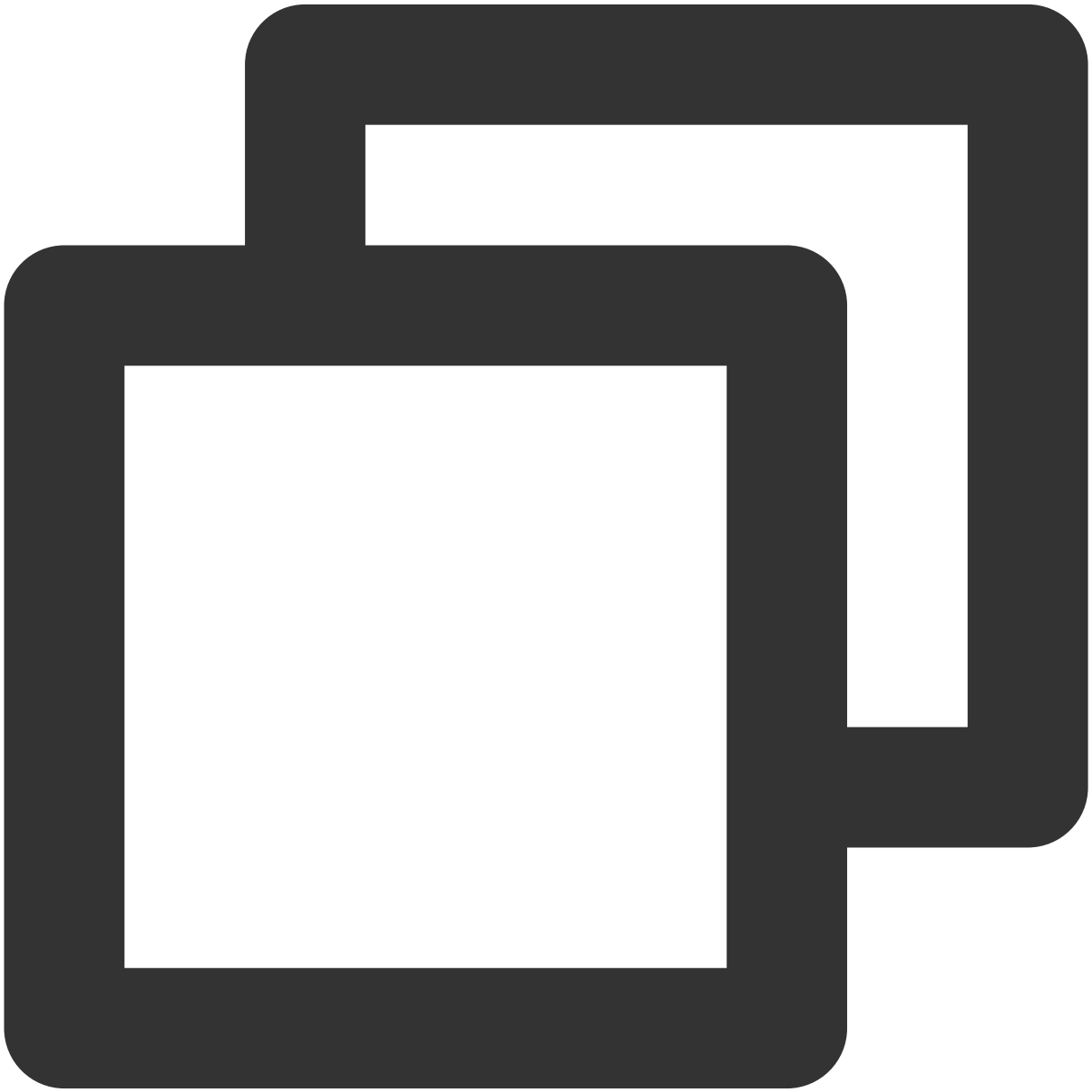
Go Connection Sample

Last updated：2023-12-21 21:10:45

Preparations before running:

Download the Go-redis client.

Sample code:



```
package main
```

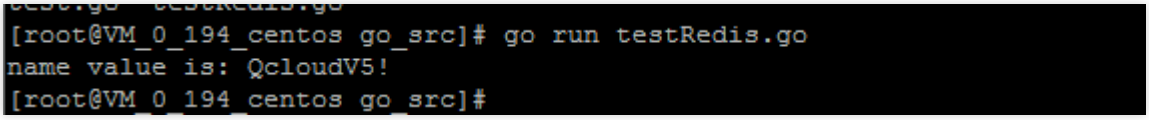
©2013-2022 Tencent Cloud. All rights reserved.

```
import(

   "fmt"

   "redis"

   "log"

)

func main() {

   const host=192.xx.xx.195
   const port=6379
   const instanceId="84ffd722-b506-4934-9025-64xxx997b"
   const pass="123d7sq"
   // Connect to the Tendis server via the IP address "192.xx.xx.195:6379" and auth
   spec := redis.DefaultSpec().Host(host).Port(port).Password(instanceId+":"+pass);
   client, err := redis.NewSynchClientWithSpec(spec)

   if err != nil { // Whether an error occurs with the connection

      log.Println("error on connect redis server")

      return

   }

   newvalue :=[]byte("QcloudV5!");

   err=client.Set("name",newvalue);

   if err != nil { // Incorrect value

      log.Println(err)

      return

   }

   value, err := client.Get("name") // Value

   if err != nil {

      log.Println(err)

      return
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    }

    fmt.Println("name value is:",fmt.Sprintf("%s", value)) // Output

}
```

Execution results:



©2013-2022 Tencent Cloud. All rights reserved.

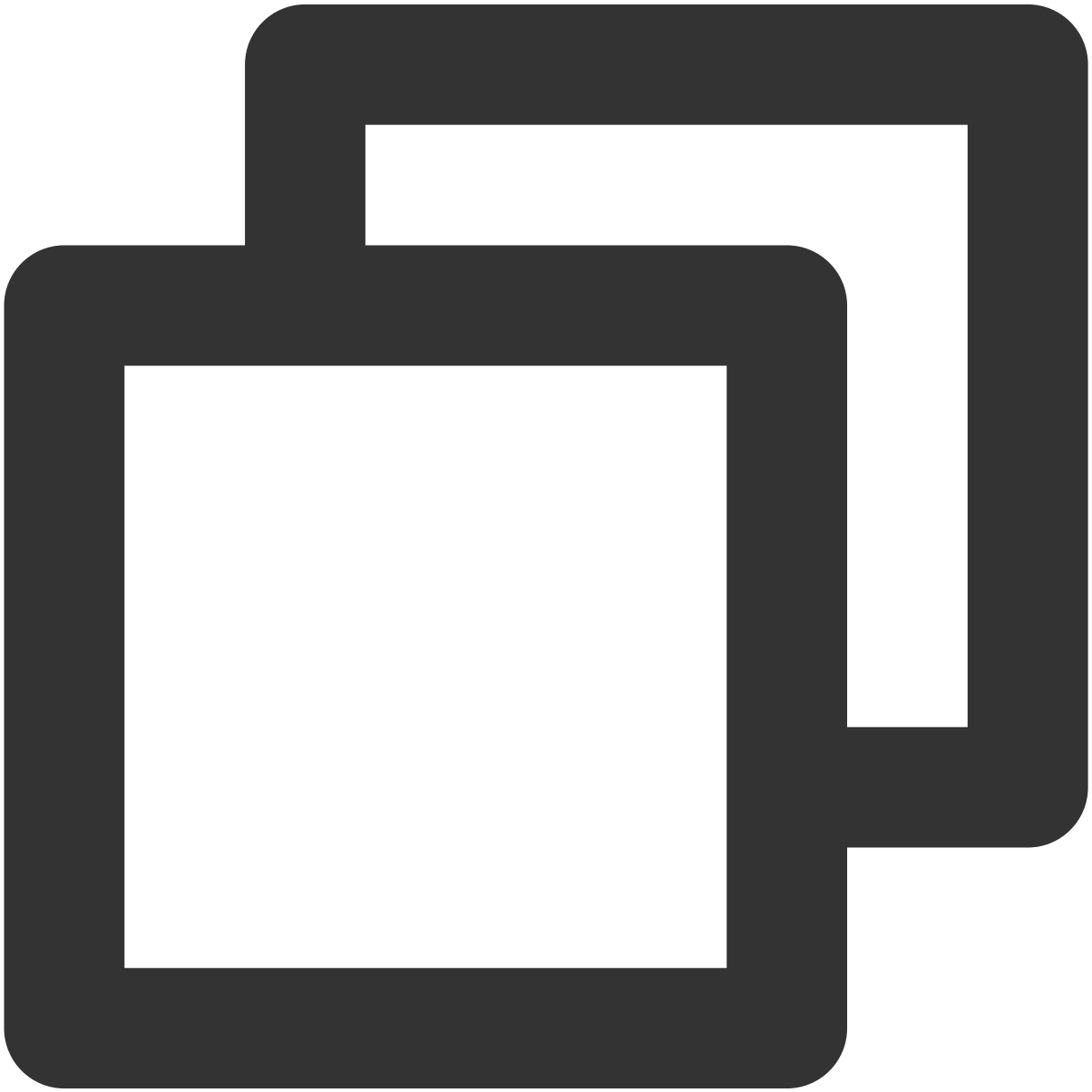
Java Connection Sample

Last updated：2023-12-21 21:10:55

Preparations before running:

Download the Jedis client.

Sample code:

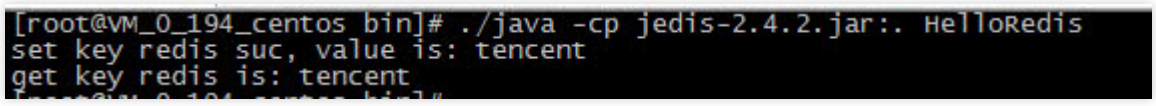


```
import redis.clients.jedis.Jedis;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
public class HelloRedis {

  public static void main(String[] args) {
        try {
            /**Enter your Tendis instance private IP, port number, instance ID, and
            String host = "192.xx.xx.195";
            int port = 6379;
            String instanceid = "crs-09xxxqv";
            String password = "123ad6aq";
            // Connect to the Tendis instance
            Jedis jedis = new Jedis(host, port);
            // Authenticate
            jedis.auth(instanceid + ":" + password);

            /**You can start manipulating the Tendis instance. For more information
            // Set the key
            jedis.set("redis", "tencent");
            System.out.println("set key redis suc, value is: tencent");
            // Get the key
            String value = jedis.get("redis");
            System.out.println("get key redis is: " + value);

            // Close and exit
            jedis.quit();
            jedis.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Execution results:



©2013-2022 Tencent Cloud. All rights reserved.

Node.js Connection Sample

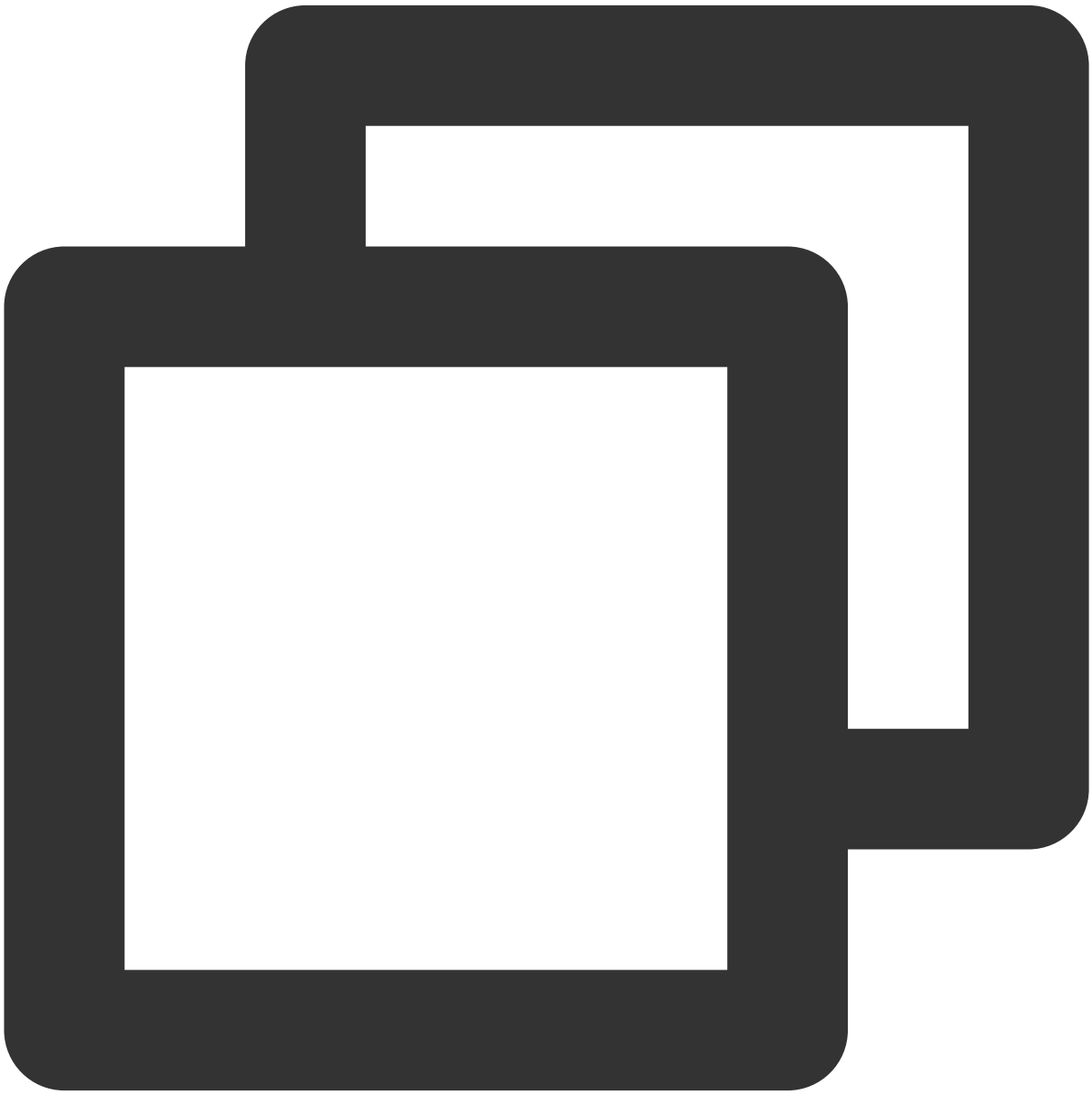
Last updated：2023-12-21 21:11:10

Preparations before running:

Run the following command to install node-redis:

```
npm install hiredis redis
```

Sample code:



```
var redis = require("redis");
```

©2013-2022 Tencent Cloud. All rights reserved.

```
/**Enter your Tendis instance private IP, port number, instance ID, and password in
var host = "192.xx.xx.2",
port = "6379",
instanceid = "c53xx52f-55dc-4c22-a941-630xxx88",
pwd = "12as6zb";
// Connect to the Tendis instance
var client  = redis.createClient(port, host, {detect_buffers: true});
// Connection error
client.on("error", function(error) {
    console.log(error);
});
// Authenticate
client.auth(instanceid + ":" + pwd);

/**You can start manipulating the Tendis instance. */
// Set the key
client.set("redis", "tencent", function(err, reply){
    if (err) {
        console.log(err);
            return;
    }
    console.log("set key redis " + reply.toString() + ", value is tencent");
});

// Get the key
client.get("redis", function (err, reply) {
    if (err) {
        console.log(err);
        return;
    }
    console.log("get key redis is:" + reply.toString());
// End the program and close the client
    client.end();
});
```

Execution results:

```
[root@VM_0_3_centos bin]# ./node Test.js
set key redis suc, value is:OK
get key redis is:tencent
```

©2013-2022 Tencent Cloud. All rights reserved.

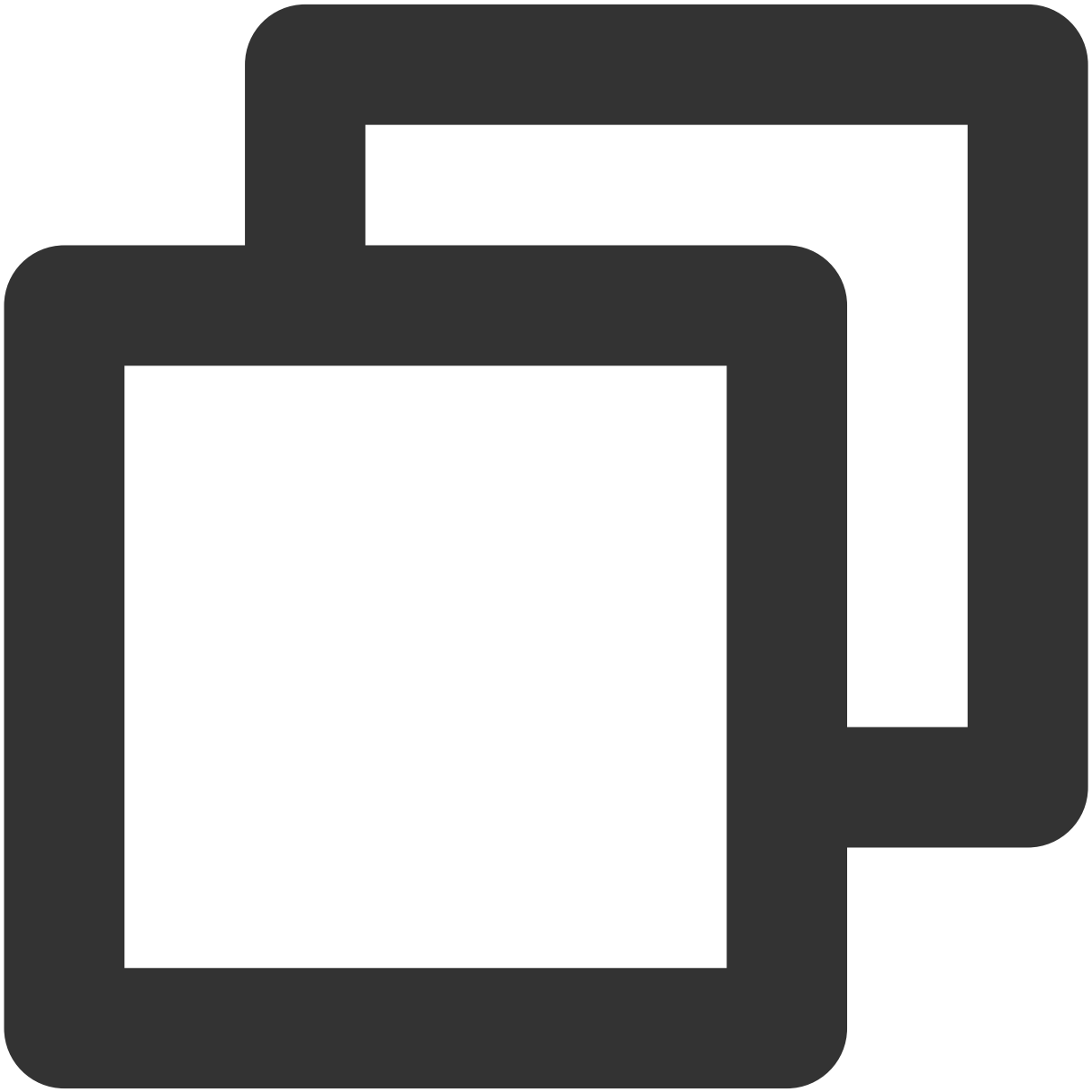
PHP Connection Sample

Last updated：2023-12-21 21:11:22

Preparations before running:

Download the phpredis client.

Sample code:



```
<?php
  /**Enter your Tendis instance private IP, port number, instance ID, and password
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    $host = "192.xx.xx.2";
    $port = 6379;
    $pwd = "123tj6na";

    $redis = new Redis();
    // Connect to the Tendis instance
    if ($redis->connect($host, $port) == false) {
      die($redis->getLastError());
    }
    // Authenticate
    if ($redis->auth($pwd) == false) {
      die($redis->getLastError());
    }

    /**You can start manipulating the Tendis instance. For more information, please v

    // Set the key
    if ($redis->set("redis", "tencent") == false) {
      die($redis->getLastError());
    }
    echo "set key redis suc, value is:tencent\\n";

    // Get the key
    $value = $redis->get("redis");
    echo "get key redis is:".$value."\\n";
  ?>
```

Execution results:

```
[root@VM_0_3_centos bin]# ./php Test.php
set key redis suc, value is:tencent
get key redis is:tencent
```

©2013-2022 Tencent Cloud. All rights reserved.

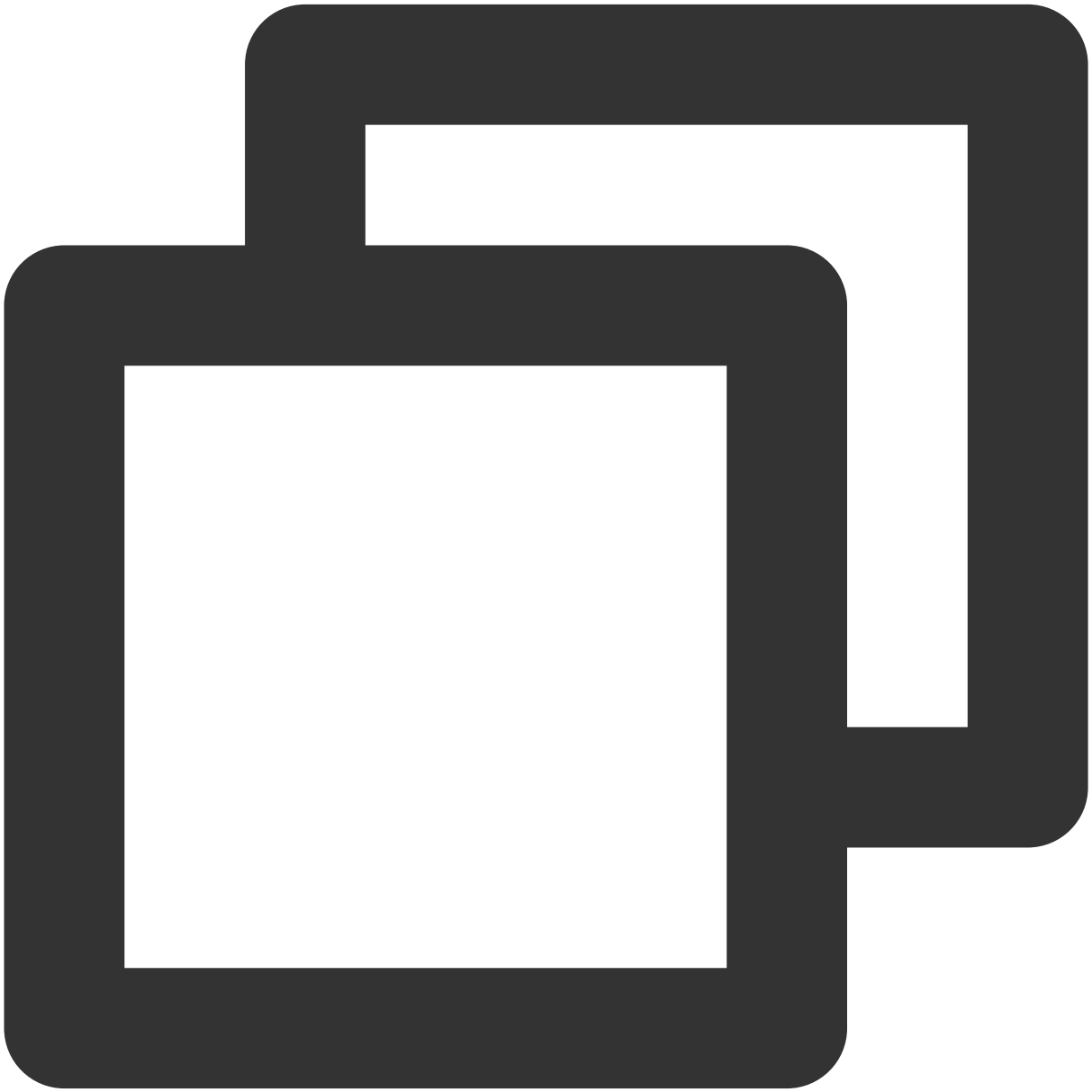
Python Connection Sample

Last updated：2023-12-21 21:11:33

Preparations before running:

Download and install redis-py.

Sample code:

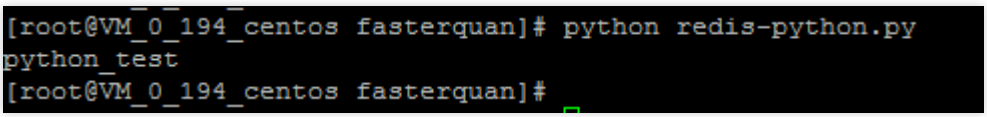


```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
```

©2013-2022 Tencent Cloud. All rights reserved.

```
import redis

#Replace with the host address and port number of the instance to be connected
host = '192.xx.xx.195'
port = 6379

#Replace with the password of the instance to be connected

pwd='password'

#When connecting, specify the AUTH information through the "password" parameter.
r= redis.StrictRedis(host=host, port=port, password=pwd)

#Database operations can be performed after the connection is established. For more
r.set('name', 'python_test');
print r.get('name')
```

Execution results:



©2013-2022 Tencent Cloud. All rights reserved.

Instance Maintenance and Management
Specifying Projects for Instances

Last updated：2023-12-21 21:11:48

This document describes how to assign instances to different projects in the console for easier management.

Overview

TencentDB for Tendis supports assigning instances to different projects for easier management. Assigned instances can be reassigned to other projects.

Note:

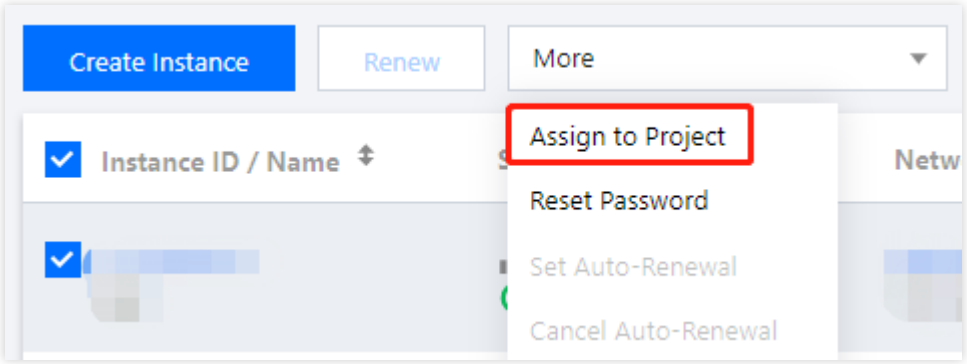
Assigning and reassigning TencentDB instances will not affect the services provided by the instances.

Directions

1. Log in to the TencentDB for Tendis console, select a region, select the desired instance in the instance list, and click **More** > **Assign to Project** at the top.

Note:

Alternatively, you can click the instance name/ID, and on the displayed instance details page, click **Assign to Project** in the **Project** section.



2. In the pop-up dialog box, select a project and click **OK**.

©2013-2022 Tencent Cloud. All rights reserved.

Upgrading Instance Specification

Last updated：2023-12-21 21:11:57

This document describes how to upgrade the node specification and expand shard disk capacity for a TencentDB for Tendis instance in the console.

Overview

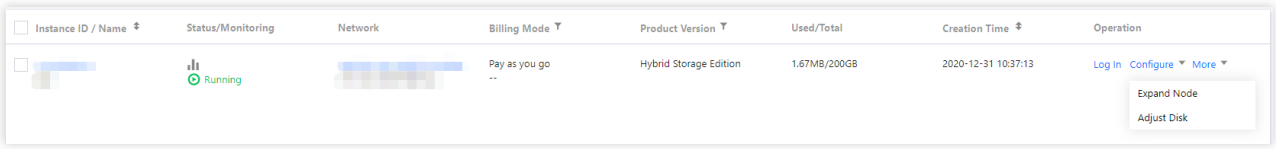
You can elastically scale your TencentDB for Tendis instances as needed to ensure sufficient resources and better resource allocation.

Note:

Currently, TencentDB for Tendis does not support capacity reduction, and the Storage Edition supports neither capacity expansion nor reduction.

Directions

1. Log in to the TencentDB for Tendis console. In the instance list, select a region at the top, locate the instance to be scaled, and click **Configure** > **Expand Cache** or **Expand Disk** in the **Operation** column.



2. In the pop-up dialog box, adjust the configuration and click **OK**.

Note:

After the configuration is adjusted, the instance will be charged at the price of the new configuration.

To avoid failure in capacity reduction, the capacity after reduction should be at least 1.3 times the amount of existing data.

When shards are added or deleted, the system will automatically balance the slot configuration and migrate data.

During capacity expansion and reduction, such blocking commands as `BLPOP` , `BRPOP` , `BRPOPLPUSH` , and `SUBSCRIBE` may fail once or more (which is related to the number of shards). Please assess the impact on your business before starting capacity expansion and reduction.

During capacity expansion and reduction, commands executed on an instance with read-only replicas enabled may fail once or more (which is related to the number of shards). Please assess the impact on your business before starting capacity expansion and reduction.

©2013-2022 Tencent Cloud. All rights reserved.

3. Return to the instance list. After the status of the instance changes to **Running**, the instance can be used normally.

©2013-2022 Tencent Cloud. All rights reserved.

Terminating Instances

Last updated：2023-12-21 21:12:09

This document describes how to terminate a TencentDB for Tendis instance in the console.

Overview

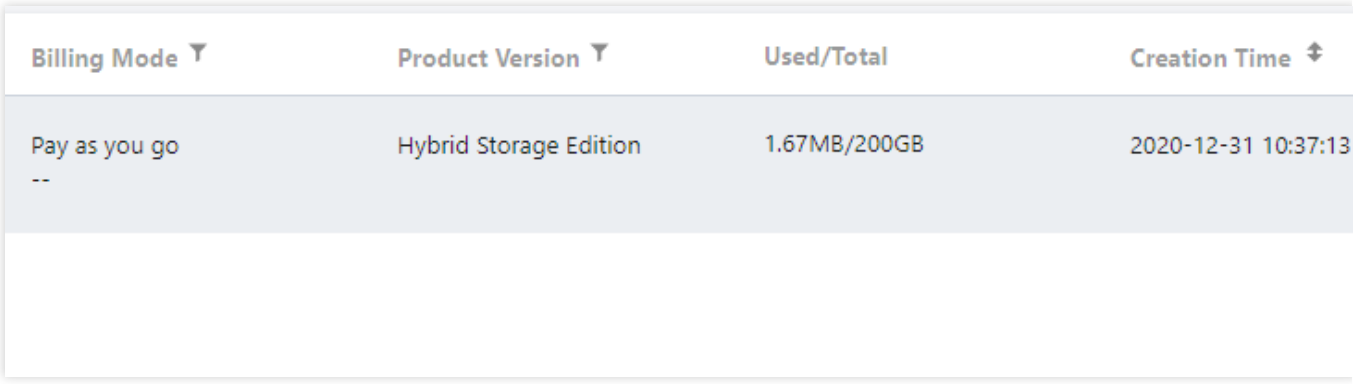
Based on your business needs, you can return pay-as-you-go instances in the console in a self-service manner.
After a pay-as-you-go instance is returned, it will be moved to the TencentDB recycle bin and retained there for 24 hours. During the retention period, the instance cannot be accessed but can be restored after startup.
When an instance is returned and its status has changed to **Isolated**, it will no longer generate fees.

Note:

After the instance is terminated, its data cannot be recovered, and its backup files will also be terminated, so the data cannot be restored in the cloud. Please store your backup files safely elsewhere in advance.
When an instance is terminated, its IP address will be released.

Directions

1. Log in to the TencentDB for Tendis console. In the instance list, select a region at the top, locate the desired instance, and click **More** > **Terminate** in the **Operation** column.



2. In the pop-up dialog box, confirm that everything is correct and click **Terminate**.

Note:

Note that all data will be cleared and cannot be recovered after termination.

©2013-2022 Tencent Cloud. All rights reserved.

Monitoring Features

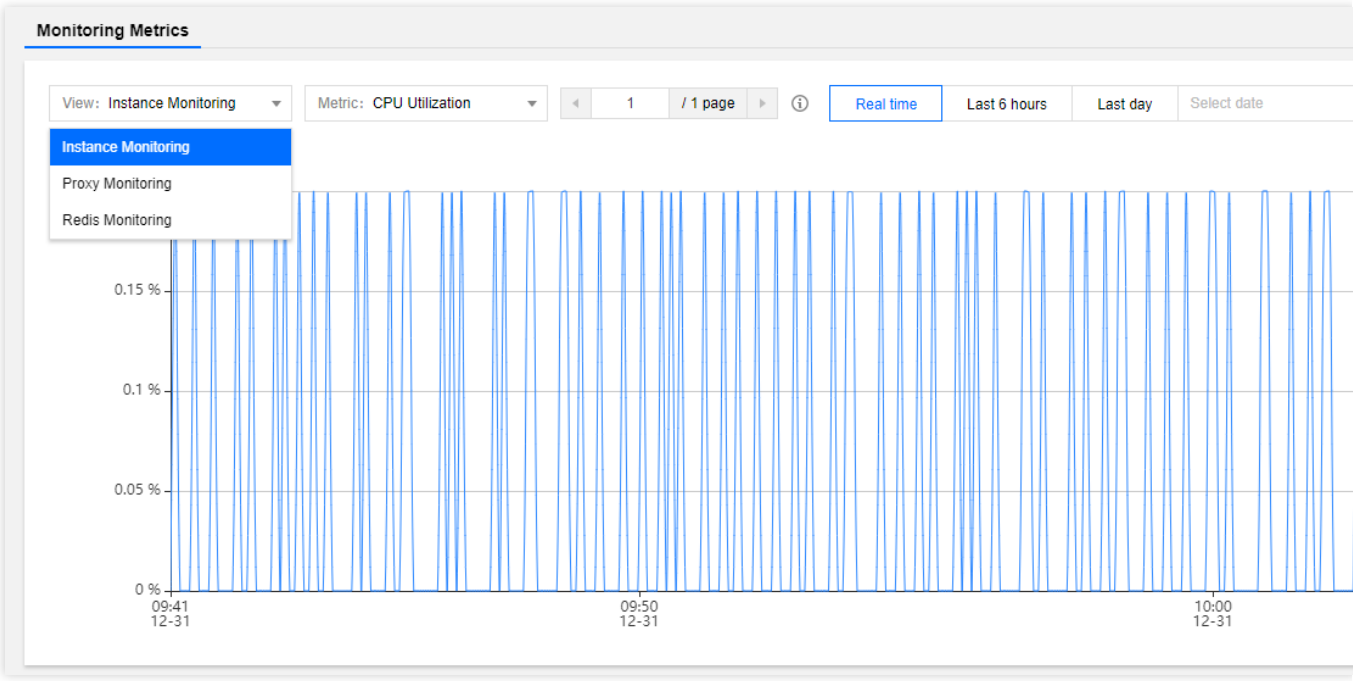
Last updated：2023-12-21 21:12:28

TencentDB for Tendis provides a complete and easy-to-use monitoring service where you don’t have to worry about, for example, collecting monitoring data or OPS of the monitoring system. The monitoring service includes Proxy monitoring, Redis monitoring, and Tendis monitoring which summarizes the monitoring data of an entire instance. Details are as follows:

Proxy monitoring: provides monitoring information of all Proxy nodes in an instance. TencentDB for Tendis instances in standard or cluster architecture have Proxy nodes.

Redis monitoring: provides monitoring information of TencentDB for Tendis primary and secondary nodes.

Tendis monitoring: summarizes the monitoring data of an entire instance (including Proxy nodes and Tendis nodes) and aggregates data according to the SUM, AVG, MAX, and LAST aggregation algorithms.



Monitoring Granularity and Monitoring Data Retention Period

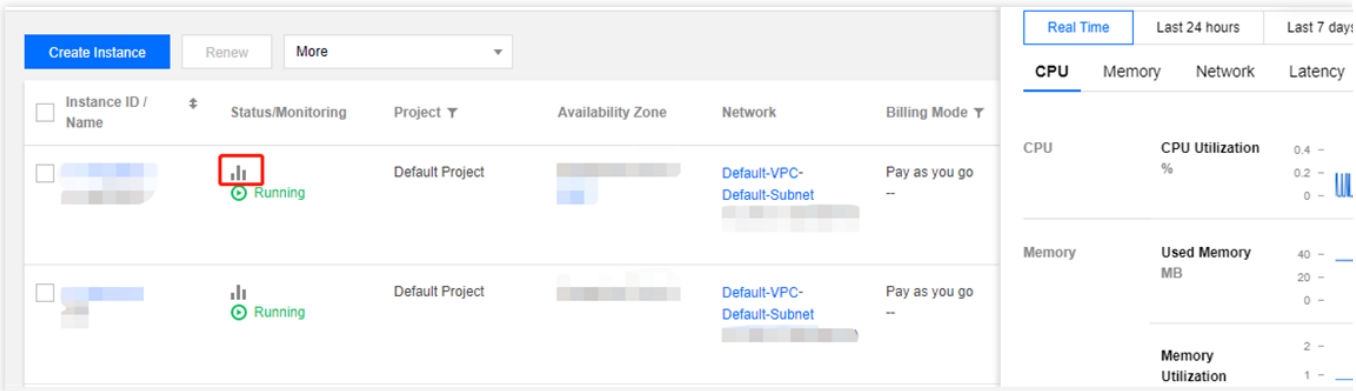
Tendis currently supports monitoring metrics at the 1-minute, 5-minutes, 1-hour, or 1-day granularity. For the retention period of monitoring data at each granularity, please see Use Limits.

Viewing Monitoring Information

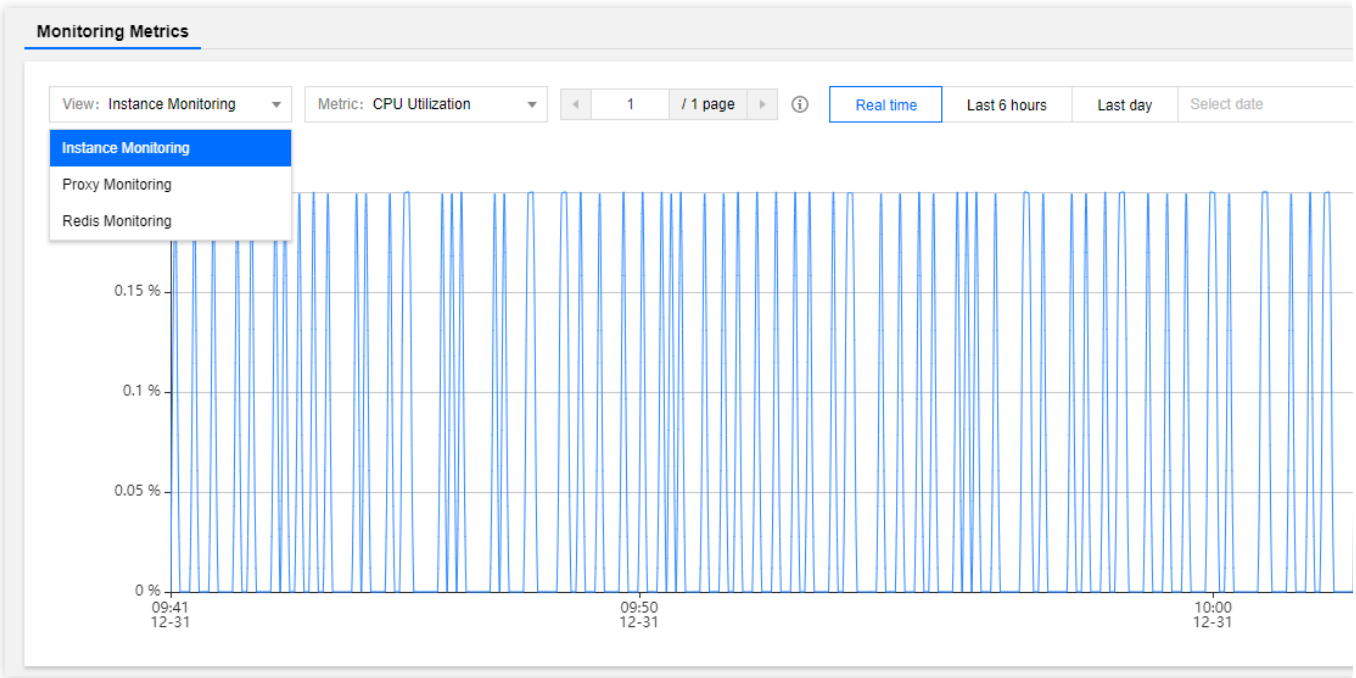
You can view TencentDB for Tendis monitoring information in the instance list and on the instance monitoring page in the TencentDB for Tendis console, or in the Cloud Monitor console.

©2013-2022 Tencent Cloud. All rights reserved.

Instance list: log in to the TencentDB for Tendis console, click the **View Monitoring** icon in the instance list as shown below, and view monitoring metrics in the pop-up window on the right.



Instance monitoring page: log in to the TencentDB for Tendis console, click an instance ID in the instance list and enter the instance management page, select **System Monitoring**, and view monitoring data on the **Monitoring Metrics** tab.



Monitoring Metric Description

Proxy monitoring

Each Tendis instance contains at least 3 Proxy nodes. Generally, the number of Proxy nodes is 1.5 times that of Tendis nodes. The Proxy node supports the following monitoring metrics:

Category	Metric	Parameter	Unit	Description
CPU	CPU utilization	cpu_util	%	Proxy CPU utilization

©2013-2022 Tencent Cloud. All rights reserved.

Request	Total requests	proxy_commands	requests/second	The number of Proxy command executions per second
	Key requests	cmd_key_count	keys/second	The number of keys accessed by a command per second
	Mget requests	cmd_mget	requests/second	The number of Mget command executions per second
	Execution errors	cmd_err	errors/second	The number of Proxy command execution errors per second, for example, when a command does not exist, parameters are incorrect, etc.
	Big value requests	cmd_big_value	requests/second	The number of executions of commands larger than 32 KB per second
Network	Connections	connections	-	The number of TCP connections to an instance
	Connection usage	connections_util	%	The ratio of the number of TCP connections to the maximum number of connections
	Inbound traffic	in_flow	MB/s	Private network inbound traffic
	Inbound traffic utilization	in_bandwidth_util	%	The ratio of the actually used private inbound traffic to the maximum traffic
	Inbound traffic limit count	in_flow_limit	-	The number of times inbound traffic triggers a traffic limit
	Outbound traffic	out_flow	MB/s	Private network outbound traffic
	Outbound traffic utilization	out_bandwidth_util	%	The ratio of the actually used private outbound traffic to the maximum traffic
	Outbound traffic limit count	out_flow_limit	-	The number of times outbound traffic triggers a traffic limit

©2013-2022 Tencent Cloud. All rights reserved.

Latency	Average execution latency	latency_avg	ms	The average execution latency from Proxy to Redis server
	Max execution latency	latency_max	ms	The maximum execution latency from Proxy to Redis server
	Average read latency	latency_read	ms	The average execution latency of read commands from Proxy to Redis server. For more information about read command types, please see Command types.
	Average write latency	latency_write	ms	The average execution latency of write commands from Proxy to Redis server. For more information about write command types, please see Command types.
	Average latency of other commands	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server

Redis monitoring

The Redis node monitoring includes monitoring information of all primary nodes and secondary nodes in an instance or a cluster. The following monitoring metrics are supported.

Category	Metric	Parameter	Unit	Description
CPU	CPU utilization	cpu_util	%	Average CPU utilization
Network	Connections	connections	-	The number of connections from Proxy to a node
	Connection usage	connections_util	%	The connection usage of a node
Memory	Used memory	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	mem_util	%	The ratio of the memory actually used to the total memory requested

©2013-2022 Tencent Cloud. All rights reserved.

	Keys	keys	-	Total number of keys stored in an instance (first-level keys)
	Expired keys	expired	-	The number of keys expired in a time window, which is equal to the value of `expired_keys` output by the `info` command
	Evicted keys	evicted	-	The number of keys evicted in a time window, which is equal to the value of `evicted_keys` output by the `info` command
	Replication delay	repl_delay	Byte	The command delay between the secondary node and the primary node
Request	Total requests	commands	queries/second	QPS, that is, the number of command executions per second
	Read requests	cmd_read	requests/second	The number of read command executions per second. For more information about read command types, please see Command types.
	Write requests	cmd_write	requests/second	The number of write command executions per second. For more information about write command types, please see Command types.
	Other requests	cmd_other	requests/second	The number of command (excluding write and read commands) executions per second
Response	Slow queries	cmd_slow	-	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read request hits	cmd_hits	-	The number of keys successfully requested by read commands, which is equal to the value of the `keyspace_hits` metric output by the `info` command
	Read request misses	cmd_miss	-	The number of keys unsuccessfully requested by read commands, which is equal to the value of the

©2013-2022 Tencent Cloud. All rights reserved.

				`keyspace_misses` metric output by the `info` command
	Read request hit rate	cmd_hits_ratio	%	Key hits/(Key hits + Key misses). This metric reflects the cache miss situation.

Tendis monitoring

The Tendis monitoring includes all monitoring data of an instance, including the monitoring data of Proxy nodes and Redis nodes, which is aggregated by the SUM, AVG, MAX, and LAST algorithms.

Category	Metric	Associated Node View	Parameter	Unit	Description
CPU	CPU utilization	Tendis node	cpu_util	%	Average CPU utilization
	Max CPU utilization of a node	Tendis node	cpu_max_util	%	The maximum CPU utilization of a node (shard or replica) in an instance
Memory	Used memory	Tendis node	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	Tendis node	mem_util	%	The ratio of the memory actually used to the total memory requested
	Max memory utilization of a node	Tendis node	mem_max_util	%	The maximum memory utilization of a node (shard or replica) in an instance
	Keys	Tendis node	keys	-	Total number of keys stored in an instance (first-level keys)
	Expired keys	Tendis node	expired	-	The number of keys expired in a time window, which is equal to the value of `expired_keys` output

©2013-2022 Tencent Cloud. All rights reserved.

					by the `info` command
	Evicted keys	Tendis node	evicted	-	The number of keys evicted in a time window, which is equal to the value of `evicted_keys` output by the `info` command
Network	Connections	Proxy node	connections	-	The number of TCP connections to an instance
	Connection usage	Proxy node	connections_util	%	The ratio of the number of TCP connections to the maximum number of connections
	Inbound traffic	Proxy node	in_flow	MB/s	Private network inbound traffic
	Inbound traffic utilization	Proxy node	in_bandwidth_util	%	The ratio of the actually used private inbound traffic to the maximum traffic
	Inbound traffic limit count	Proxy node	in_flow_limit	-	The number of times inbound traffic triggers a traffic limit
	Outbound traffic	Proxy node	out_flow	MB/s	Private network outbound traffic
	Outbound traffic utilization	Proxy node	out_bandwidth_util	%	The ratio of the actually used private outbound traffic to the maximum traffic
	Outbound traffic limit count	Proxy node	out_flow_limit	-	The number of times outbound traffic triggers a traffic limit
	Average execution	Proxy node	latency_avg	ms	Average execution latency from Proxy to

©2013-2022 Tencent Cloud. All rights reserved.

	latency				Redis server
	Max execution latency	Proxy node	latency_max	ms	Maximum execution latency from Proxy to Redis server
	Average read latency	Proxy node	latency_read	ms	The average execution latency of read commands from Proxy to Redis server. For more information about read command types, please see Command types.
	Average write latency	Proxy node	latency_write	ms	The average execution latency of write commands from Proxy to Redis server. For more information about wirte command types, please see Command types.
	Average latency of other commands	Proxy node	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server
Request	Total requests	Tendis node	commands	requests/second	QPS, that is, the number of command executions per second
	Read requests	Tendis node	cmd_read	requests/second	The number of read command executions per second. For more information about read command types, please see Command types.
	Write	Tendis	cmd_write	requests/second	The number of write

©2013-2022 Tencent Cloud. All rights reserved.

requests	node			command executions per second. For more information about write command types, please see Command types.
Other requests	Tendis node	cmd_other	requests/second	The number of command (excluding write and read commands) executions per second
Big value requests	Proxy node	cmd_big_value	requests/second	The number of executions of commands larger than 32 KB per second
Key requests	Proxy node	cmd_key_count	keys/second	The number of keys accessed by a command per second
Mget requests	Proxy node	cmd_mget	requests/second	The number of Mget command executions per second
Slow queries	Tendis node	cmd_slow	-	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
Read request hits	Tendis node	cmd_hits	-	The number of keys successfully requested by read commands, which is equal to the value of the `keyspace_hits` metric output by the `info` command
Read request misses	Tendis node	cmd_miss	-	The number of keys unsuccessfully requested by read

©2013-2022 Tencent Cloud. All rights reserved.

					commands, which is equal to the value of the `keyspace_misses` metric output by the `info` command
	Execution errors	Proxy node	cmd_err	-	The number of command execution errors, for example, when a command does not exist, parameters are incorrect, etc.
	Read request hit rate	Tendis node	cmd_hits_ratio	%	Key hits/(Key hits + Key misses). This metric reflects the cache miss situation.

Command types

Type	Commands
Read command	get, strlen, exists, getbit, getrange, substr, mget, llen, lindex, lrange, sismember, scard, srandmember, sinter, sunion, sdiff, smembers, sscan, zrange, zrangebyscore, zrevrangebyscore, zrangebylex, zrevrangebylex, zcount, zlexcount, zrevrange, zcard, zscore, zrank, zrevrank, zscan, hget, hmget, hlen, hstrlen, hkeys, hvals, hgetall, hexists, hscan, randomkey, keys, scan, dbsize, type, ttl, touch, pttl, dump, object, memory, bitcount, bitpos, georadius_ro, georadiusbymember_ro, geohash, geopos, geodist, pfcount
Write command	set, setnx, setex, psetex, append, del, unlink, setbit, bitfield, setrange, incr, decr, rpush, lpush, rpushx, lpushx, linsert, rpop, lpop, brpop, brpoplpush, blpop, lset, ltrim, lrem, rpoplpush, sadd, srem, smove, spop, sinterstore, sunionstore, sdiffstore, zadd, zincrby, zrem, zremrangebyscore, zremrangebyrank, zremrangebylex, zunionstore, zinterstore, hset, hsetnx, hmset, hincrby, hincrbyfloat, hdel, incrby, decrby, incrbyfloat, getset, mset, msetnx, swapdb, move, rename, renamenx, expire, expireat, pexpire, pexpireat, flushdb, flushall, sort, persist, restore, restore-asking, migrate, bitop, geoadd, georadius, georadiusbymember, pfadd, pfmerge, pfdebug

©2013-2022 Tencent Cloud. All rights reserved.

©2013-2022 Tencent Cloud. All rights reserved.

Configuring Security Groups

Last updated：2023-12-21 21:12:45

Overview

A security group is a stateful virtual firewall capable of filtering. As an important means for network security isolation provided by Tencent Cloud, it can be used to set network access controls for one or more TencentDB instances. Instances with the same network security isolation demands in one region can be put into the same security group, which is a logical group. TencentDB and CVM share the security group list and are matched with each other within the security group based on rules. For specific rules and limitations, please see Security Group Overview.

Note:

TencentDB security group currently only supports network access control for VPCs and public network but not the classic network.

Security groups that currently support public network access are available only in the Guangzhou, Shanghai, Beijing, and Chengdu regions.

As TencentDB does not have active outbound traffic, outbound rules are not applicable to TencentDB.

Security Group Configuration for TencentDB

Step 1. Create a security group

1. Log in to the CVM console.
2. Select **Security Group** on the left sidebar, select a region, and click **New**.
3. In the pop-up dialog box, configure the following items and click **OK**.

Template: select a template based on the service to be deployed on the TencentDB instance in the security group, which simplifies the security group rule configuration, as shown below:

Template	Description	Remarks
Open all ports	All ports are open. May present security issues.	-
Open ports 22, 80, 443, and 3389 and the ICMP protocol	Ports 22, 80, 443, and 3389 and the ICMP protocol are opened to the internet. All ports are opened to the private network.	This template does not take effect for TencentDB.
Custom	You can create a security group and then add custom rules. For detailed directions, please see "Step 2. Add a security group rule" below.	-

©2013-2022 Tencent Cloud. All rights reserved.

Name: name of the security group.

Project: by default, **DEFAULT PROJECT** is selected. Select a project for easier management.

Notes: a short description of the security group for easier management.

Step 2. Add a security group rule

1. On the Security Group page, click **Modify Rules** in the **Operation** column on the row of the security group for which to configure a rule.
2. On the security group rule page, click **Inbound rule** > **Add a Rule**.
3. In the pop-up dialog box, set the rule.

Type: **Custom** is selected by default. You can also choose another system rule template.

Source or **Target**: traffic source (inbound rules) or target (outbound rules). You need to specify one of the following options:

Source or Target	Description
A single IPv4 address or an IPv4 range	In CIDR notation, such as 203.0.113.0, 203.0.113.0/24 or 0.0.0.0/0, where 0.0.0.0/0 indicates all IPv4 addresses will be matched.
A single IPv6 address or an IPv6 range	In CIDR notation, such as FF05::B5, FF05:B5::/60, ::/0 or 0::0/0, where ::/0 or 0::0/0 indicates all IPv6 addresses will be matched.
ID of referenced security group. You can reference the ID of: Current security group Other security group	To reference the current security group, please enter the ID of security group associated with the CVM. You can also reference another security group in the same region and belongs to the same project by entering the security group ID.
Reference an IP address object or IP address group object in a parameter template.	-

Protocol port: enter the protocol type and port range or reference a protocol/port or protocol/port group in a parameter template.

Note:

To connect to TencentDB for Tendis, port 6379 must be opened.

Policy: **Allow** or **Refuse**. **Allow** is selected by default.

Allow: traffic to this port is allowed.

Refuse: data packets will be discarded without any response.

Notes: a short description of the rule for easier management.

4. Click **Complete**.

©2013-2022 Tencent Cloud. All rights reserved.

Step 3. Configure a security group

A security group is an instance-level firewall provided by Tencent Cloud for controlling inbound traffic of TencentDB. You can associate a security group with an instance when purchasing it or later in the console.

Note:

Currently, security groups can be configured only for TencentDB for Tendis instances in VPC.

1. Log in to the TencentDB for Tendis console. In the instance list, click an instance ID or **Manage** in the **Operation** column to access the instance management page.
2. On the **Security Group** page, click **Configure Security Group**.
3. In the pop-up dialog box, select the security group to be bound and click **OK**.

Security Group Rule Import

1. On the Security Group page, click the ID of the target security group.
2. On the inbound rules or outbound rules tab, click **Import Rule**.
3. In the pop-up dialog box, select an edited inbound/outbound rule template file and click **Import**.

Note:

If there are existing rules in the security group, export them before importing new rules. Existing rules are overwritten after importing.

Security Group Clone

1. On the Security Group page, locate the desired security group and click **More** > **Clone** in the **Operation** column.
2. In the pop-up dialog box, select the target region and target project, enter the new security group name, and click **OK**. If the new security group needs to be associated with a CVM instance, do so by managing the CVM instances in the security group.

Security Group Deletion

1. On the Security Group page, locate the security group to be deleted and click **More** > **Delete** in the **Operation** column.
2. Click **OK** in the pop-up dialog box. If the current security group is associated with a CVM instance, it must be disassociated before it can be deleted.

©2013-2022 Tencent Cloud. All rights reserved.

Disabling Commands

Last updated：2023-12-21 21:12:59

This document describes how to disable commands in the TencentDB for Tendis console.

Overview

TencentDB for Tendis supports disabling some commands that may cause service instability or accidentally delete data, by configuring the `disable-command-list` parameter.

Directions

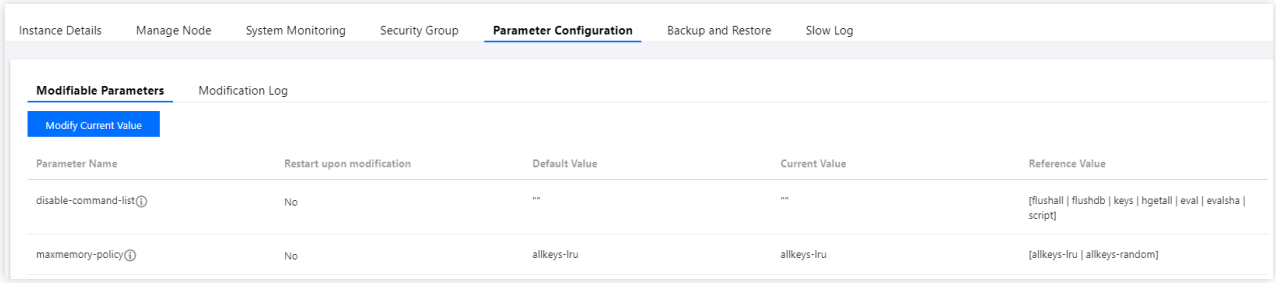
Disabling a command

1. Log in to the TencentDB for Tendis console, select a region, click an instance ID in the instance list, and enter the instance management page.
2. Select **Parameter Configuration** > **Modifiable Parameters** and configure the list of commands to be disabled in the `disable-command-list` parameter line.

Note:

Commands that can be disabled include `flushall`, `flushdb`, `keys`, `hgetall`, `eval`, `evalsha`, and `script`.

Command disablement will take effect within two minutes for existing connections without restarting the Tendis service.



Enabling a disabled command

1. Log in to the TencentDB for Tendis console, select a region, click an instance ID in the instance list, and enter the instance management page.

©2013-2022 Tencent Cloud. All rights reserved.

2. Select **Parameter Configuration** > **Modifiable Parameters** and remove a command from the list of disabled commands in **Current Value** to enable it.

Parameter modification history

1. Log in to the TencentDB for Tendis console, select a region, click an instance ID in the instance list, and enter the instance management page.
2. View the parameter modification history in **Parameter Configuration** > **Modification Log**.

©2013-2022 Tencent Cloud. All rights reserved.