TencentDB for Redis

Operation Guide

Product Documentation

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

Operation Overview

Last updated：2023-03-14 15:41:18

During the use of TencentDB for Redis, you may encounter problems related to accessing, maintaining, backing up, and restoring instances. This document describes common operations on TencentDB for Redis instances.

Instance

A database instance can contain multiple user-created databases and can be accessed using the same client tools and applications as those for a standalone database instance.
Common operations in a TencentDB for Redis instance are as follows:
Creating Instance
Connecting to Instance
Assigning Instance to Project
Changing Instance Specification
Enabling/Disabling Read/Write Separation
Clearing Instance
Terminating Instance
Upgrading Instance Version
Upgrading Instance Architecture
Configuring Multi-AZ Deployment
Upgrading to Multi-AZ Deployment
Accessing Multi-AZ Deployed Instance
Password-Free Access
Managing Account
Setting Instance Parameters
Applying Parameter Templates
Disabling Commands

Data

Common data operations in a TencentDB for Redis are as follows:

Backup and restoration

Backing up Data

Cloning Data

Data migration

Migration with redis-port

Migration with DTS

Security Group and Network

Security group

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.

Network

Configuring Network

Authorization Policy Syntax

Configuring Public Network Address

Monitoring and Alarms

Monitoring at Five-Second Granularity

Comparing Monitoring Data Among Instances

Configuring Alarm

Global Replication

Creating Global Replication Group

Managing Global Replication Group

Access Management

Overview

Last updated：2023-10-20 10:59:32

Issues

If you have multiple users managing different Tencent Cloud services such as CVM, VPC, and TencentDB, and they all share your Tencent Cloud account access key, you may face the following problems:

Your key will be easily compromised because it is shared by several users.

Your users might introduce security risks from misoperations due to the lack of user access control.

Solution

These problems can be eliminated by the use of sub-accounts which allow you to authorize different users to manage your different services. By default, a sub-account has no access to Tencent Cloud services or resources. To grant a sub-account such access, you need to create a CAM policy.

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions to your Tencent Cloud resources. Using CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management.

When using CAM, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, see Syntax Logic.

You can skip this section if you do not need to manage permissions to TDSQL-A resources for sub-accounts. This will not affect your understanding and use of the other sections of the document.

Getting started

A CAM policy must authorize or deny the use of one or more Redis operations. At the same time, it must specify the resources that can be used for the operations (which can be all resources or partial resources for certain operations). A policy can also include the conditions set for the manipulated resources.

Note

We recommend that you manage Redis resources and authorize Redis operations through CAM policies. Although the user experience does not change for existing users who are granted permissions by project, we do not recommend that you continue to manage resources and authorize operations in a project-based manner.

Effectiveness conditions cannot be set for Redis for the time being.

Relevant Information	Link
Basic policy structure	Authorization Policy Syntax
Operation definition in a policy	Redis Operations
Resource definition in a policy	Redis Resource Path
Resource-level permissions	Resource-level Permissions Supported by Redis

Authorization Policy Syntax

Last updated：2020-11-12 11:16:17

CAM Policy Syntax

```
{
         "version":"2.0",
          "statement":
          [
             {
                "effect":"effect",
                "action":["action"],
                "resource":["resource"],
                 "condition": {"key":{"value"}}
             }
         ]
  }
```

version: it is required. Currently, only "2.0" is allowed.

statement: describes the details of one or more privileges. This element contains a privilege or privilege set of other elements such as effect, action, resource, and condition. One policy has only one statement.

effect: describes whether the result produced by the statement is "allowed" (allow) or "denied" (deny). This element is required.

action: describes the allowed or denied operation. An operation can be an API or a feature set (a set of specific APIs prefixed with "permid"). This element is required.

resource: describes the details of authorization. A resource is described in a six-piece format. Detailed resource definitions vary by product. This element is required.

condition: describes the condition for the policy to take effect. A condition consists of operator, action key, and action value. A condition value may contain information such as time and IP address. Some services allow you to specify additional values in a condition. This element is optional.

Redis Operations

In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with "redis:" should be used for Redis, such as redis:CreateRedis or redis:DeleteInstance.

To specify multiple actions in a single statement, separate them with commas, as shown below:

```
"action":["redis:action1","redis:action2"]
```

You can also specify multiple actions using a wildcard. For example, you can specify all actions whose name begins with "Describe", as shown below:

```
"action":["redis:Describe*"]
```

If you want to specify all operations in Redis, use a wildcard "*" as shown below:

```
"action"："["redis:*"]
```

Redis Resource Path

Each CAM policy statement has its own resources.

The general form of resource path is as follows:

```
qcs:project_id:service_type:region:account:resource
```

project_id: describes the project information, which is only used to enable compatibility with legacy CAM logic and can be left empty.

service_type: describes the product abbreviation such as Redis.

-**region**: region information, for example, bj.

account: the root account of the resource owner, such as uin/653339763.

resource: describes detailed resource information of each product, such as instanceId/instance_id1 or instanceId/*.

For example, you can specify a resource for a specific instance (crs-psllioc8) in a statement as shown below:

```
"resource":[ "qcs::redis:bj:uin/12345678:instance/crs-psllioc8"]
```

You can also use the wildcard "*" to specify all instances that belong to a specific account as shown below:

```
"resource":[ "qcs::redis:bj:uin/12345678:instance/*"]
```

If you want to specify all resources or if a specific API operation does not support resource-level permission control, you can use the wildcard "*" in the "resource" element as shown below:

```
"resource": ["*"]
```

To specify multiple resources in a single command, separate them with commas. Below is an example where two resources are specified:

```
"resource":["resource1","resource2"]
```

The table below describes the resources that can be used by Redis and the corresponding resource description methods, where words prefixed with $ are placeholders, "region" refers to a region, and "account" refers to an account ID.

Resource	Resource Description Method in Authorization Policy
Instance	`qcs::redis:$region:$account:instance/$instanceId`

VPC	`qcs::vpc:$region:$account:vpc/$vpcId`
Security group	`qcs::cvm:$region:$account:sg/$sgId`

Authorizable Resource Types

Last updated：2023-05-23 10:31:25

Resource-level permission can be used to specify which resources a user can manipulate. Redis supports certain resource-level permission. This means that for Redis operations that support resource-level permission, you can control the time when a user is allowed to perform operations or to use specified resources. The following table describes the types of resources that can be authorized in CAM.

Resource Type	Resource Description in Authorization Policy
TencentDB for Redis instance	`qcs::redis:$region::instance/*` `qcs::redis:$region:$account:instance/$instance`

The table below lists the TencentDB for Redis API operations that currently support resource-level permission control, as well as the resources that each operation supports. You can use the `*` wildcard in a resource path when defining it.

List of APIs supporting resource-level authorization

API Name	Resource Path
AssignProject	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
AssociateSecurityGroups	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
AutoRenew	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
BackupInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CleanInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CleanUpInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ClearInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ClearRedis	`qcs::redis:$region:$account:instance/*`

	`qcs::redis:$region:$account:instance/$instance`
CreateInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CreateInstanceAccount	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CreateInstanceHour	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CreateInstances	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
CreateRedis	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DeleteInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DeleteInstanceAccount	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeAutoBackupConfig	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeBackupUrl	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeDBSecurityGroupsDetail	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceAccount	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceBackups	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceDealDetail	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceParamRecords	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceParams	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`

DescribeInstanceSecurityGroup	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceSecurityGroupsAssociated	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceShards	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstanceSlowlog	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeInstances	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeProjectSecurityGroup	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedis	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedisDealDetail	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedisProduct	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedisProductList	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedisRegions	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeRedisZones	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeSlowLog	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeTaskInfo	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeTaskList	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DescribeTasks	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`

DescribeVPCRedis	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DestroyPostpaidInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DestroyPrepaidInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
DisableReplicaReadonly	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
EnableReplicaReadonly	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ExportRedisBackup	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
GetBackupDownloadUrl	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
GetRedisBackupList	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
GetRedisPerformance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
GetRedisSlowLogList	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
GetRedisTaskList	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
InitRedisPassword	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
InquiryRedisPrice	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ManualBackupInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModfiyInstancePassword	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModfiyRedisPassword	`qcs::redis:$region:$account:instance/*`

	`qcs::redis:$region:$account:instance/$instance`
ModifyAutoBackupConfig	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyDBInstanceSecurityGroups	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyInstanceAccount	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyInstanceParams	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyInstanceSecurityGroup	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyNetworkConfig	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyRedisName	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyRedisParams	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ModifyRedisProject	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
RenewInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
RenewRedis	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ResetPassword	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
ResetRedisPassword	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
RestoreInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`

SetRedisAutoRenew	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
StartupInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
SwitchInstanceVip	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
UnAssociateSecurityGroups	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
UpgradeInstance	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
UpgradeRedis	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`
UpgradeRedisInquiryPrice	`qcs::redis:$region:$account:instance/*` `qcs::redis:$region:$account:instance/$instance`

List of APIs not supporting resource-level authorization

For a TencentDB API operation that does not support authorization at the resource level, you can still authorize a user to perform it, but you must specify `*` as the resource element in the policy statement.

API Name	Description
CreateInstances	Creates a TencentDB for Redis instance
CreateParamTemplate	Creates a parameter template
DeleteParamTemplate	Deletes a parameter template
DescribeInstanceDealDetail	Queries the order information
DescribeParamTemplateInfo	Gets the details of parameter template
DescribeParamTemplates	Queries the list of parameter templates
DescribeTaskInfo	DescribeTasks
DescribeTasks	Queries the list of tasks
ModifyParamTemplate	Modifies a parameter template
ListUsers	Queries sub-account names

ListCollaborators	Queries collaborator account names
ListWeChatWorkSubAccounts	Queries WeChat usernames

SDK Connection

PHP Connection Sample

Last updated：2022-07-05 10:55:17

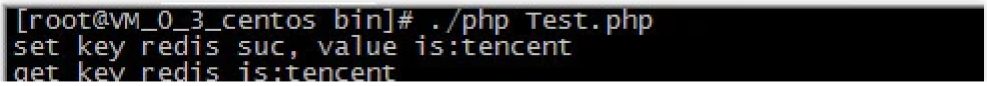
Preparations before running:

Download the phpredis client.

Sample code:

```
<?php
  /**For the following parameters, enter your Redis instance's private IP, port num
  $host = "192.xx.xx.2";
  $port = 6379;
  $instanceid = "c532952f-55dc-4c22-a941-63057e560788";
  $pwd = "123tj6na";

  $redis = new Redis();
  // Connect to Redis
  if ($redis->connect($host, $port) == false) {
    die($redis->getLastError());
  }
  // Authenticate
  if ($redis->auth($instanceid . ":" . $pwd) == false) {
    die($redis->getLastError());
  }

  /**You can start manipulating the Redis instance. For more information, see https

  // 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 result:



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

Java Connection Sample

Last updated：2023-03-14 15:53:35

This document provides client code samples for Java to help you access a database with or without SSL encryption enabled.

Preparations

Get the **private IPv4 address** and **port** information for database connection in the **Network Info** section on the **Instance Details** page in the TencentDB for Redis console. For detailed directions, see Viewing Instance Information.

Get the account and password for database access. For detailed directions, see Managing Account.

Download and install Jedis. The latest version is recommended.

If you want to connect to the database over SSL, enable SSL encryption to get the SSL certificate file.

Connection sample with SSL encryption not enabled

You need to modify the parameters based on the comments, including IP, port, account, and password for database access.

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

public class HelloRedis {

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

            /**You can start manipulating the Redis 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:

Connection sample with SSL encryption enabled

You need to modify the parameters based on the comments, including SSL certificate file, IP, port, account, and password for database access.

```
import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;

import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;


public class Main {

    public static void main(String[] args) throws Exception {
        KeyStore trustStore = KeyStore.getInstance("jks");
        // `ca.jks` is the certificate file name.
        try (InputStream inputStream = new FileInputStream("ca.jks") ){
            trustStore.load(inputStream, null);
        }
        TrustManagerFactory trustManagerFactory =      TrustManagerFactory.getInsta
        trustManagerFactory.init(trustStore);
        TrustManager[] trustManagers = trustManagerFactory.getTrustManagers();

        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, trustManagers, new SecureRandom());
        SSLSocketFactory sslSocketFactory =  sslContext.getSocketFactory();
        GenericObjectPoolConfig genericObjectPoolConfig = new GenericObjectPoolConf

        //with ssl config jedis pool
        // `vip` is the private IPv4 address for database connection, `6379` is the
        JedisPool pool = new JedisPool(genericObjectPoolConfig, "vip",
                6379, 2000, "pwd", 0, true, sslSocketFactory, null, null);
        Jedis jedis = pool.getResource();
        System.out.println(jedis.ping());
        jedis.close();
    }
}
```

Node.JS Connection Sample

Last updated：2022-07-05 10:55:45

Preparations before running:

Run the following command to install node-redis:

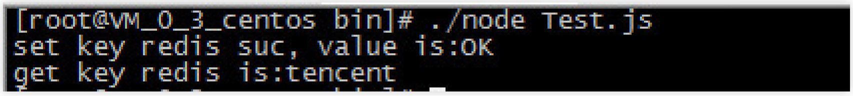
`npm install hiredis redis`

Sample code:

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

```
/**For the following parameters, enter your Redis instance's private IP, port numbe
var host = "192.xx.xx.2",
port = "6379",
instanceid = "c53xx52f-55dc-4c22-a941-630xxx88",
pwd = "12as6zb";
// Connect to Redis
var client  = redis.createClient(port, host, {detect_buffers: true});
// Redis connection error
client.on("error", function(error) {
    console.log(error);
});
// Authenticate
client.auth(instanceid + ":" + pwd);

/**You can start manipulating the Redis 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 result:



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

Python Connection Sample

Last updated：2022-07-06 15:44:23

This document provides client code samples for Python to help you access a database with or without SSL encryption enabled.

Prerequisites

Get the **private IPv4 address** and **port** information for database connection in the **Network Info** section on the **Instance Details** page in the TencentDB for Redis console. For detailed directions, see Viewing Instance Details.

Get the account and password for database access. For detailed directions, see Managing Account.

Download and install redis-py. The latest version is recommended.

If you want to connect to the database over SSL, enable SSL encryption to get the SSL certificate file.

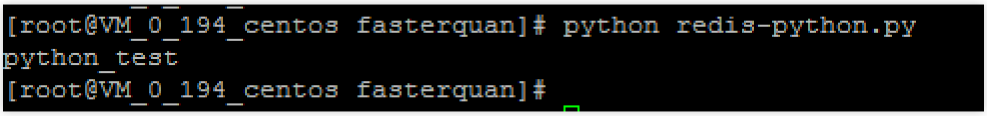
Connection Sample Without SSL Encryption Enabled

You need to modify the parameters based on the comments, including IP, port, account, and password for database access.

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-
import redis

#Replace with the connected instance host and port here
host = '192.xx.xx.195'
port = 6379

#Replace with the instance ID and password here
user='username'
pwd='password'
```

```
#When connecting, specify the AUTH information through the `password` parameter. If
r = redis.StrictRedis(host=host, port=port, password=user+'@'+pwd)

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

Execution result:



```
[root@VM_0_194_centos fasterquan]# python redis-python.py
python_test
[root@VM_0_194_centos fasterquan]#
```

Connection Sample With SSL Encryption Enabled

You need to modify the parameters based on the comments, including SSL certificate file, IP, port, account, and password for database access.

```
import redis3 as redis3

if __name__ == "__main__":
#`vip` is the private IPv4 address for database connection, `6379` is the default p
    client = redis3.Redis(host="vip", port=6379, password="pwd", ssl=True, ssl_cert
                          ssl_ca_certs="ca.pem")
    print(client.ping())
```

C Connection Sample

Last updated：2022-07-05 10:56:11

Preparations before running:

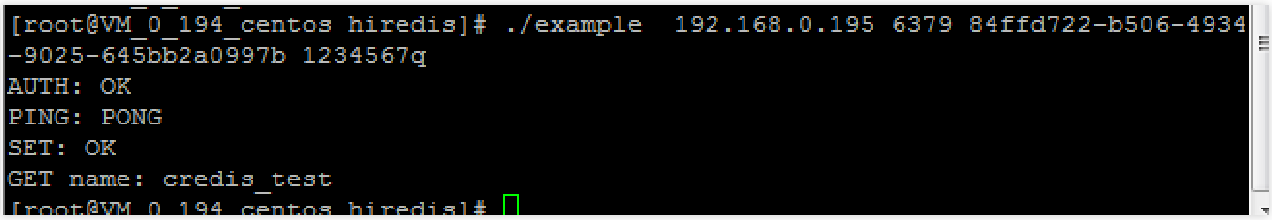
Doawnload and install hiredis.

Sample Code:

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

```
#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  */
    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:



```
[root@VM_0_194_centos hiredis]# ./example  192.168.0.195 6379 84ffd722-b506-4934
-9025-645bb2a0997b 1234567q
AUTH: OK
PING: PONG
SET: OK
GET name: credis_test
[root@VM_0_194_centos hiredis]#
```

Go Connection Sample

Last updated：2022-07-05 10:56:24

Preparations before running:

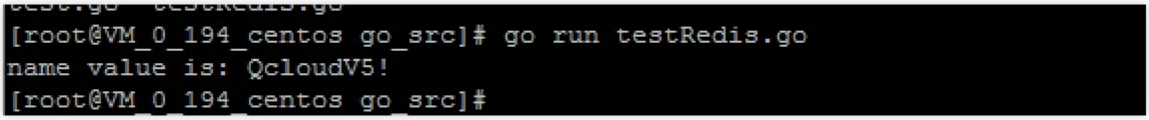
Download the Go-redis client.

Sample code:

```
package main
```

```
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 Redis server 192.xx.xx.195:6379 and authorize the instanceId p
    spec := redis.DefaultSpec().Host(host).Port(port).Password(instanceId+":"+pass);
    client, err := redis.NewSynchClientWithSpec(spec)

    if err != nil { // Whether the connection is incorrect

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

        return

    }

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

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

    if err != nil { // Incorrect value set

        log.Println(err)

        return

    }

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

    if err != nil {

        log.Println(err)

        return
    }

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

}
```

Execution result:



```
test.go  testRedis.go
[root@VM_0_194_centos go_src]# go run testRedis.go
name value is: QcloudV5!
[root@VM_0_194_centos go_src]#
```

.Net Connection Sample

Last updated：2022-07-05 10:56:40

Preparations before running:

Download and install ServiceStack.Redis.

Sample code:

Do not use connection pool

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
       {
           string host = "10.xx.xx.46";// Host address to access the instance
           int port = 6379;// Port information
           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 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

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

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

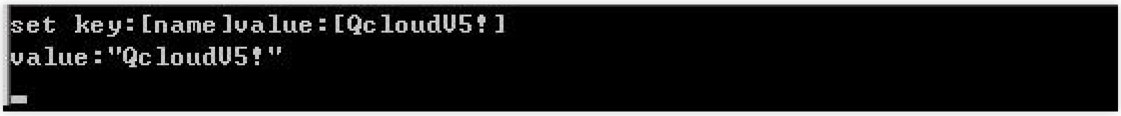
```
         {
              string[] testReadWriteHosts = new[] {
              "redis://:fb92bxxxabf11e5:1234xx8a1A@10.x.x.1:6379"/*redis://:instance
              };
              RedisConfig.VerifyMasterConnections = false;// Need to be set
              PooledRedisClientManager redisPoolManager = 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

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

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

```
    {
           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:



```
set key:[name]value:[QcloudV5!]
value:"QcloudV5!"
```

Daily Instance Operation

Viewing Instance Information

Last updated：2023-05-23 10:27:24

Overview

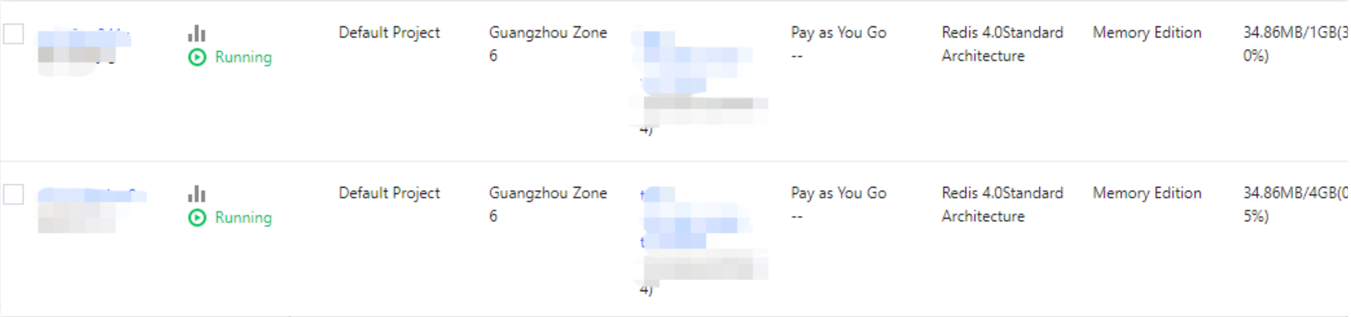
After purchasing a TencentDB for Redis instance, you can quickly view its details in the console, such as the status, capacity usage, master/replica nodes in the cluster, and network status. You can also perform Ops and management operations efficiently.

Prerequisites

You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.
The instance is isolated in the recycle bin and has not been terminated.. For more information, see Restoring Isolated Instance.

Viewing Instance List

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
In the search box in the top-right corner, you can search for the target instance by instance ID, instance name, private IP, or tag key. Only instance name can be fuzzily searched.
If you can't find the target instance in the instance list, select **Recycle Bin** on the left sidebar to check whether it is isolated there due to overdue payments. For more information, see Restoring Isolated Instance.
4. View the target instance information, such as the status, specification, and storage engine.



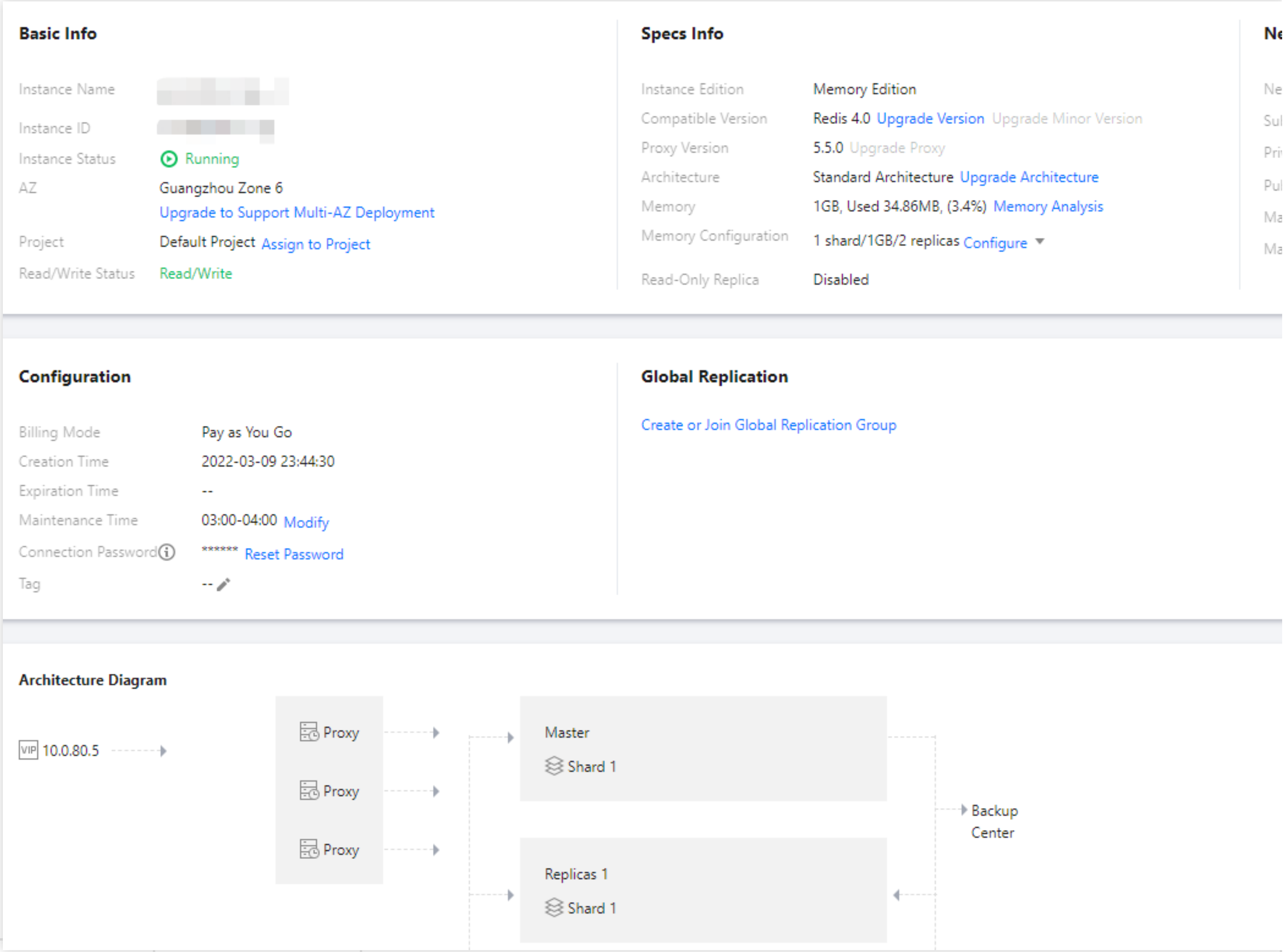
□	[redacted]	Running	Default Project	Guangzhou Zone 6	[redacted]	Pay as You Go --	Redis 4.0Standard Architecture	Memory Edition	34.86MB/1GB(3 0%)
□	[redacted]	Running	Default Project	Guangzhou Zone 6	[redacted]	Pay as You Go --	Redis 4.0Standard Architecture	Memory Edition	34.86MB/4GB(C 5%)

Instance Information List	Description
Instance ID/Name	Instance ID: The unique ID of the instance. Name: The name set when creating the instance. Mouse over the instance name, click ✎ to edit the instance name for easy identification and management.
Monitoring/Status/Task	Monitoring: Click 📊, on the monitoring panel, you can quickly view the monitoring metrics of the instance. For more information, see Monitoring at Five-Second Granularity. Status: The running status of the instance. The normal status is **Running**. Task: When there is a task to execute, the name of the task currently being executed by the instance will be displayed here, such as **Changing configuration**.
Project	The project to which the instance belongs. A Tencent Cloud account can create multiple projects. In the **Account Center** > **Project Management** in the upper right corner in the console, you can create projects, manage projects, and view the consumption details of the entire project. If you need to transfer to another project, see Assigning Instance to Project.
AZ	AZ information specified for the instance. If M is displayed on the right side of the AZ, the instance is deployed in multiple AZs. Mouse over this icon to view information about multiple AZs of the instance.
Deployment Mode	Specify whether the instance is deployed across AZs. Single-AZ: The current instance is deployed in the same AZ in the same region. Multi-AZ: The current instance is deployed in different AZs in the same region.
Network	The instance's VPC, subnet, and private IPv4 address. You can click the VPC name in blue font to view the network details. You can also configure the private

	IPv4 address for database access as instructed in Connecting to TencentDB for Redis Instance.
Billing Mode	It supports the pay-as-you-go billing method. For more information, see Billing Overview.
Auto-Renewal	Specify whether the auto-renewal feature is enabled for the current instance.
Architecture	For the version information and architecture of the database, see Product Series.
Instance Edition	Currently, only **Memory Edition** is supported.
Used/Total	The memory capacity used by the current instance and the total memory capacity.
Creation Time	The exact date and time when the instance was created.
Tag	Displays tag information for an instance. Instances can be managed by tags. For how to edit tags, see Editing Instance Tag.
Operation	Click **Log In** to access the database through DMC. For directions, see Connecting to TencentDB for Redis Instance. You can click **Configure** and then select an operation as needed. Specifically, you can select **Expand Node** or **Reduce Node** to expand or reduce the instance node memory, select **Add Replica** or **Delete Replica** to add or delete instance replicas, or select **Add Shard** or **Delete Shard** to add or delete shards in the cluster architecture respectively. For directions, see Changing Instance Specification. Choose **More** > **Performance**/**Security** to view the performance diagnosis report of the instance. For directions, see Performance Optimization Overview. Select **More** > **Security Group** to select the security group inbound rules again. For directions, see Configuring Security Group. For pay-as-you-go instances, select **More** > **Terminate** to return the instance and isolate the instance in the recycle bin. For directions, see Returning and Isolating Instance. Select **More** > **Edit Tag** to modify the tag key value of the instance. For directions, see Editing Instance Tag.

Viewing Instance Details

In the **Instance ID**/**Name** column of the target instance, click the instance ID in blue font to enter the **Instance Details** page.



Basic Info

Instance Name	[redacted]
Instance ID	[redacted]
Instance Status	Running
AZ	Guangzhou Zone 6 Upgrade to Support Multi-AZ Deployment
Project	Default Project Assign to Project
Read/Write Status	Read/Write

Specs Info

Instance Edition	Memory Edition
Compatible Version	Redis 4.0 Upgrade Version Upgrade Minor Version
Proxy Version	5.5.0 Upgrade Proxy
Architecture	Standard Architecture Upgrade Architecture
Memory	1GB, Used 34.86MB, (3.4%) Memory Analysis
Memory Configuration	1 shard/1GB/2 replicas Configure
Read-Only Replica	Disabled

Ne

Ne
Sul
Pri
Pul
Ma
Ma

Configuration

Billing Mode	Pay as You Go
Creation Time	2022-03-09 23:44:30
Expiration Time	--
Maintenance Time	03:00-04:00 Modify
Connection Password	****** Reset Password
Tag	--

Global Replication

Create or Join Global Replication Group

Architecture Diagram

Section	Field	Description
Basic Information	Instance Name	The name set when creating the instance. Mouse over the instance name, click ✎ to edit the instance name for easy identification and management.
	Instance ID	The unique ID of the instance.
	Instance Status	The running status of the instance. The normal status is **Running**.
	AZ	The region and AZ to which the instance belongs. Click **Upgrade to Support Multi-AZ Deployment** to upgrade the single-AZ deployed instance to multi-AZ deployed one. For directions, see Upgrading to Multi-AZ Deployment.
	Project	The name of the project joined by the instance. Click **Assign to Project** to reassign the instance to another project. For directions, see Assigning Instance to Project.

	Read/Write Status	The current read and write status of the database.
Spec Info	Instance Edition	Currently, only Memory Edition is supported.
	Compatible Version	The information of version compatible with the Redis protocol. If **Upgrade Minor Version** is grayed out, the current version is the latest one; if it is in blue, you can click **Upgrade Minor Version** to upgrade to a higher version and experience the new kernel features. For directions, see Upgrading Instance Version.
	Proxy Version	Version information for the Redis proxy. If **Upgrade Proxy** is grayed out, the current proxy version is the latest one; if it is in blue, you can click **Upgrade Proxy** to upgrade to a higher version. For directions, see Upgrading Proxy.
	Architecture	For the information of instance deployment architecture, see Product Series. If the instance is on a standard architecture, click **Upgrade Architecture** to upgrade the standard architecture to a cluster architecture. For directions, see Upgrading Instance Architecture.
	Memory	The total memory capacity of the current instance, the used capacity, and the proportion of the used capacity. Click **Memory Analysis** to jump to the memory analysis page, where you can view the memory overhead of big keys in the database. Then you can quickly identify big keys, analyze their information, and split or clean them. For directions, see Memory Analysis.
	Memory Configuration	The memory configuration specification of the purchased instance, including shard quantity/memory capacity of each shard node/replica quantity. Click **Configure** to adjust these specifications. For directions, see Changing Instance Specification.
	Read-Only Replica	The enablement status of read/write separation.
Network Info	Network	You can click **Switch Network** to switch the VPC and subnet. For directions, see Configuring Network. If you need to create a VPC, see Creating VPC.
	Subnet	AZ-specific subnet in the instance VPC. A VPC allows for subnets in different AZs, which communicate with each other over the private network by default.
	Private IPv4 Address	The private IP address assigned to the database instance, which needs to be configured when accessing the database. For directions, see Connecting to TencentDB for Redis Instance.

		Click ✎ to modify the IP address and port number. Click ⧉ to copy the IP address and port number directly.
	Public Network Address	The public network address for database access, which is not enabled by default. You can click **Enable** to enable it for daily testing and management. For directions, see Configuring Public Network Address.
	Max Connections	The maximum number of client connections currently limited by the database. Click **Adjust** to change the number of access connections. For directions, see Adjusting the Number of Connections. Click **Real-Time Session** to view statistics such as the source of the current real-time session and the number of active connections of the instance. For directions, see Real-Time Session.
	Max Network Throughput	Maximum network throughput for database access, which is the triggering condition for inbound and outbound traffic throttling. Click **Adjust Bandwidth** to increase additional bandwidth. For directions, see Bandwidth Adjustment.
Configuration info	Billing Mode	Instance billing mode: pay-as-you-go.
	Creation Time	The time when the instance was created.
	Maintenance Time	The maintenance time for the instance. In order to ensure the stability of the database, the background system will perform maintenance operations on the instance during the maintenance time from time to time. You can click **Modify** to adjust the maintenance time. We recommend that you set it during the off-peak hours. For directions, see Setting Maintenance Time.
	Connection Password	The password that needs to be configured when connecting to the database. Click **Reset Password** to reset the password or set password-free authentication. For directions, see Resetting Password.
	Tag	The tag associated with the instance, which can be modified. For directions, see Editing Instance Tag.
Data Sync	Sync Mode	The data sync mode used by the instance, such as DTS.
	Sync Task	Sync task ID.

	Sync Status	The status of the task execution.
	Sync Delay	The number of bytes of sync-delayed data.
	Instance Role	The role of an instance used for data sync, which indicates whether it is a source instance or a target instance.
	Sync Instance	The peer instance ID of the sync instance and its name.
Global Replication	Join or create a global replication group	The instance has not joined a global replication group. Click **Create or Join Global Replication Group** to apply to join the replication group. For directions, see Creating Global Replication Group. Before joining a global replication group, see its implementation mechanism and [use limits](https://www.tencentcloud.com/document/product/239/46563!904c30127631506ee8acc43c77151b8c).
	Replication Group ID	This parameter is displayed after the instance joins a global replication group, which is the ID of the replication group.
	Replication Group Name	This parameter is displayed after the instance joins a global replication group, which is the custom name of the replication group.
	Instance Role	The role assigned to the instance in the global replication group, which is either master or read-only instance. This parameter will be displayed after the instance joins a global replication group.
Architecture Diagram	Architecture diagram for database instance deployment.	

More Operations

Renaming an instance

1. In **Instance List**, mouse over the instance name to be modified, and click

✎ on the right side of it.

2. In the instance name input box, enter a new name, which must meet the following requirements:

It can contain 1–60 characters.

It can contain letters, digits, underscores, and hyphens.

A letter, digit, or special symbol is counted as one character.

Setting fields in the instance list

1. In the upper right corner of the instance list, click ⚙.
2. On the **Display Settings** page, select the fields to be displayed.
3. Click **OK**, and you can see the set fields in the instance list.

Exporting the instance list

In the upper right corner of the instance list, click ⤓, and you can export the entire instance list.

Related APIs

API Name	Description
describeInstances	Queries the information of the instance list

Assigning Instance to Project

Last updated：2023-10-20 10:56:23

Overview

A project is a set of applications or services that share resources. Each project is unique, with its own applications, services, and resources isolated from and unaffected by those in other projects.

You can specify an appropriate project for your database instances to facilitate collaboration. In this way, you can easily manage your resources globally and stay on top of the billing details of the entire project.

TencentDB for Redis 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.

Prerequisites

You have created a TencentDB for Redis instance and assigned it to a project. Unassigned instances belong to the **default project**. For more information, see Creating TencentDB for Redis Instance.

The database instance is in **Running** status, with no ongoing tasks.

You have created the target project in Project Management.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. In the **Instance ID**/**Name** column of the target instance, click the instance ID in blue font to enter the **Instance Details** page.
5. In the **Basic Info** section on the **Instance Details** page, click **Assign to Project** after **Project**.
6. In the pop-up window, click **View Details**, confirm the information of the instance to be moved, search for the target project by name, select it, and click **OK**.
7. Wait for the project transfer to complete. Then, you can see that the project name in **Project** has been changed.

Related APIs

API Name	Description
ModifyInstance	Modifies the information of an instance

Editing Instance Tag

Last updated：2023-05-23 10:48:56

Overview

A tag consists of a tag key and value. It can be used to tag TencentDB for Redis instances. If you have multiple types of resources under your Tencent Cloud account which are correlated in many ways, and your resources are growing and becoming increasingly difficult to manage, you can use tags to group and categorize resources that serve the same purpose or are associated with each other. In this way, when performing daily Ops or troubleshooting, you can quickly search for resources and perform batch operations for more efficient Ops.

Billing Details

Tag is a free service provided by Tencent Cloud for your Tencent Cloud account. There are no additional charges, simply go to the Console to use this service.

Note

A tag consists of 1 tag key and 1 tag value (tagKey:tagValue).

Each resource can have 50 tags at most.

For each resource, each tag key can have only 1 value.

Prerequisites

You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.

The current tags of the instance need to be edited.

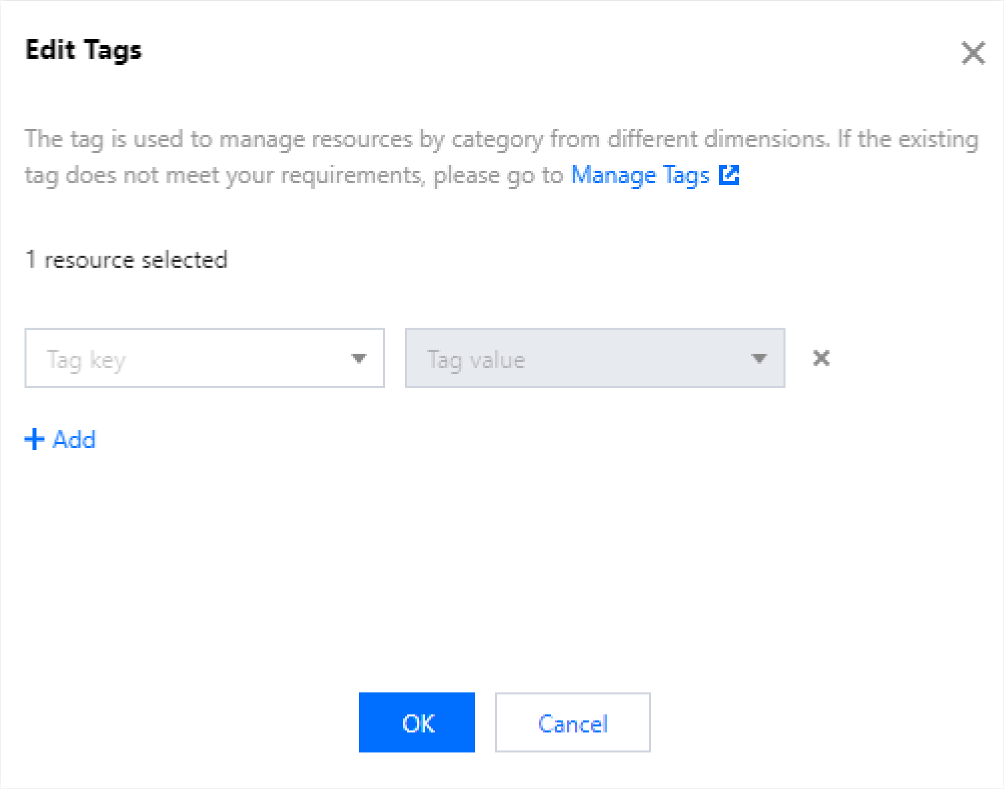
Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Enter the **Edit Tag** page in any of the following ways:

In the **Operation** column of the target instance, select **More** > **Edit Tag**.

Click the target instance ID, go to the **Configuration Info** on the **Instance Details** page, and click ✎ on the right side of **Tag**.

5. On the **Edit Tag** page, select an appropriate tag key from the **Tag Key** drop-down list and select the tag value from the **Tag Value** drop-down list.



Edit Tags

The tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to Manage Tags

1 resource selected

6. (Optional) If existing tags don't meet your business requirements, perform the following operations:

6.1 In the top-right corner of the current page, click **Manage Tags**.

6.2 On the **Manage Tags** page, click **Create Tag**.

6.3 On the **Create Tag** page, read the notes on tag configuration.

6.4 Set a new tag key in the **Tag Key** input box and enter the tag value in the **Tag Value** input box. The requirements for the tag key are as follows:

It can contain 1–127 characters.

It can contain letters and digits.

It can contain the following special symbols: plus sign, equal sign, underscore, hyphen, dot, colon, slash, @, parentheses, and brackets.

6.5 Click **OK**.

6.6 Go back to the **Edit Tag** page of the database instance. Click **Reload** in the **Tag Key** drop-down list, select the created tag key, and select the tag value.

7. Click **OK**.

References

For more information on tag management, see Tag.

Setting Maintenance Time

Last updated：2023-06-15 16:52:01

Overview

Maintenance time is a very important concept for TencentDB for Redis. To ensure the stability of your TencentDB for Redis instance, the backend system performs maintenance operations and troubleshooting on the instance during the maintenance time. To minimize the potential impact on your business, we recommend that you set an acceptable maintenance time for your business instance, usually during off-peak hours.

In addition, we also recommend you perform operations involving data migration during the maintenance time, such as instance version or architecture upgrade. Taking the database instance version upgrade as an example, as syncing the full and incremental data involves data migration, after the upgrade is completed, a momentary disconnection from the database may occur. When the upgrade is initiated, the **Switch Time** can be selected as **During maintenance time**, so that the instance version switch will be started during the next **maintenance time** after the data sync is completed. Note that when you select **During maintenance time** for **Switch Time**, the switch will not occur immediately after the database version upgrade is completed; instead, the sync will continue till the instance goes into the next **maintenance time**, during which the switch will be performed. Therefore, the overall time it takes to upgrade the instance may be longer.

Note

Before maintenance is carried out for TencentDB for Redis, notifications will be sent to the contacts configured in your Tencent Cloud account via SMS and email.

Version Description

The maintenance time is 03:00–04:00 AM by default, which can be adjusted as needed on all Redis versions.

Prerequisites

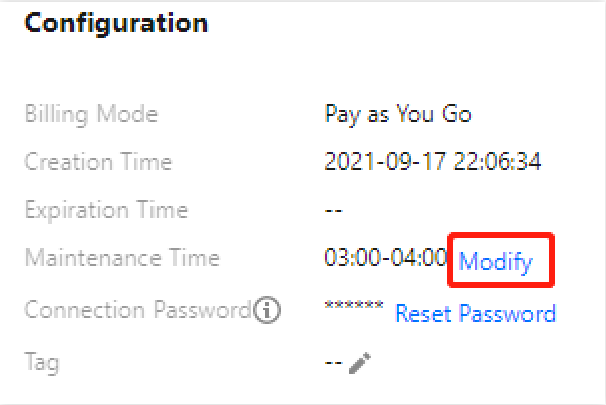
You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.

The instance is in **Running** status.

The instance maintenance time needs to be adjusted.

Directions

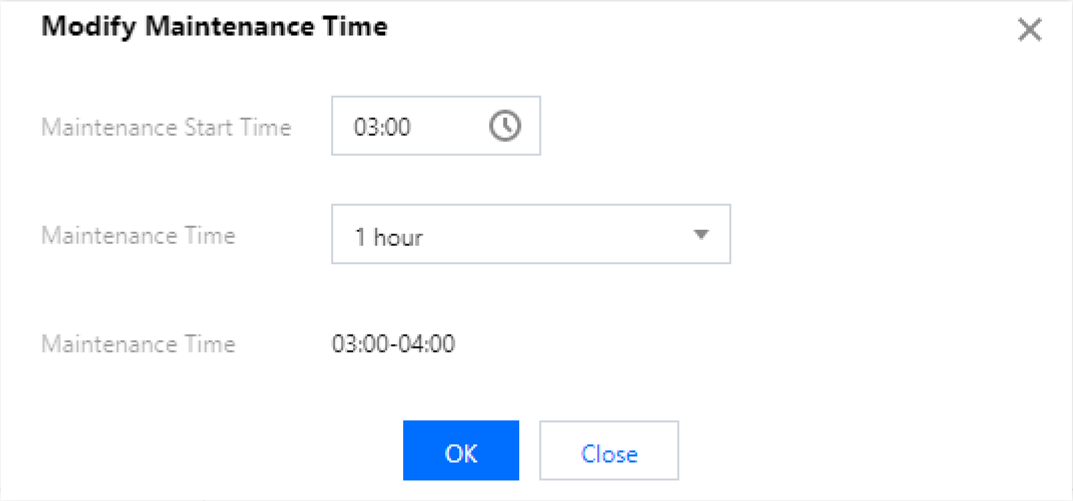
1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Configuration** section on the **Instance Details** page, click **Modify** on the right of **Maintenance Time**.



Configuration

Billing Mode	Pay as You Go
Creation Time	2021-09-17 22:06:34
Expiration Time	--
Maintenance Time	03:00-04:00 Modify
Connection Password	****** Reset Password
Tag	--

6. In the **Modify Maintenance Time** window, refer to the table below to set the maintenance time.



Modify Maintenance Time

Maintenance Start Time: 03:00

Maintenance Time: 1 hour

Maintenance Time: 03:00-04:00

Parameter	Description
Maintenance Start Time	Select the start time of the maintenance time in the drop-down list.
Maintenance Duration	Select the desired duration of the maintenance time in the drop-down list, which can be 30 minutes, 1 hour, 1.5 hours, 2 hours, or 3 hours.
Maintenance Time	The new maintenance time is displayed, which is 03:00–04:00 AM by default.

7. Click **OK**.

Related APIs

API Name	Description
DescribeMaintenanceWindow	Queries instance maintenance time
ModifyMaintenanceWindow	Modifies instance maintenance time

Changing Instance Specification

Last updated：2024-03-13 11:28:40

Overview

This document describes how to elastically scale an instance in the TencentDB for Redis console to better optimize resource utilization and costs in real time.

For instance specifications adjustment, an instance can be scaled quickly in the console without having to stop the services. No operations are required at your side.

Expand or reduce nodes: This refers to adjusting the memory capacity of instance nodes to meet ever-changing memory needs and avoid lags caused by insufficient memory.

Add or delete replicas: This refers to adjusting the number of instance replicas. Replicas are nodes other than the master node. All replicas of the Standard Edition play a role in supporting the system's high availability, so the more the replicas, the higher the availability. If the number of replicas is greater than or equal to 1, read/write separation can be enabled to extend the read performance through replica nodes.

Add or delete shards: This refers to adjusting the number of shard nodes by assigning them different keys in the sharding mode of instances in cluster architecture, so that the system performance can be horizontally scaled.

Version Description

Currently, Redis 4.0, 5.0, and 6.2 Standard Edition instances support expanding and reducing nodes as well as adding and deleting replicas.

Currently, Redis 4.0, 5.0, and 6.2 Cluster Edition instances support expanding and reducing nodes as well as adding and deleting replicas and shards.

Currently, Redis 2.8 Standard Edition instances only support expanding and reducing nodes.

Billing Overview

Monthly subscription

Configuration upgrade: The system will calculate the price difference between instance specifications and charge it to your account. You need to top up first if your account balance is insufficient. Subsequently, usage fees will be calculated based on the new instance specification.

Configuration downgrade: The price difference is calculated as follows: **Refund = residual value of the original configuration - purchase price of the new configuration**.

Residual value of original configuration: It equals to the effective order amount of the original configuration minus the used value of the original configuration.

Effective order amount of original configuration: It equals to the amount paid for the effective order, excluding discounts and vouchers.

Used value of the original configuration: As of the day when downgrade is initiated, if the usage has lasted for one month or longer, fees will be calculated at the corresponding monthly subscription price and discount listed on the official website; otherwise, fees will be calculated at the pay-as-you-go price. The usage is accurate down to the second.

Purchase value of new configuration: It equals to the current official price of the new configuration multiplied by the remaining validity period.

For monthly bill details, see Pricing.

Pay-as-you-go

The instance will be billed hourly based on the new specification on the next hour under tier 1, and fees will be settled on each hour (Beijing time). The pay-as-you-go billing mode adopts tiered pricing in three tiers as detailed in Billing Overview. For tiered prices, see Pricing.

Prerequisites

You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.

The instance and its associated instances are in **Running** status and are not executing any tasks.

You have calculated the required specifications and understood the fees. Make sure that your Tencent Cloud account balance is sufficient.

Memory Edition (Standard Architecture)

Note:

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

To expand the capacity of a Memory Edition instance in standard architecture, if the remaining available capacity of the physical machine is insufficient, migration will occur, which will not affect your access to the instance. However, a momentary disconnection will occur after the migration is completed, so we recommend that your business have a reconnection mechanism.

As the maximum capacity of a Memory Edition instance in standard architecture is 64 GB, you cannot expand its capacity beyond that limit.

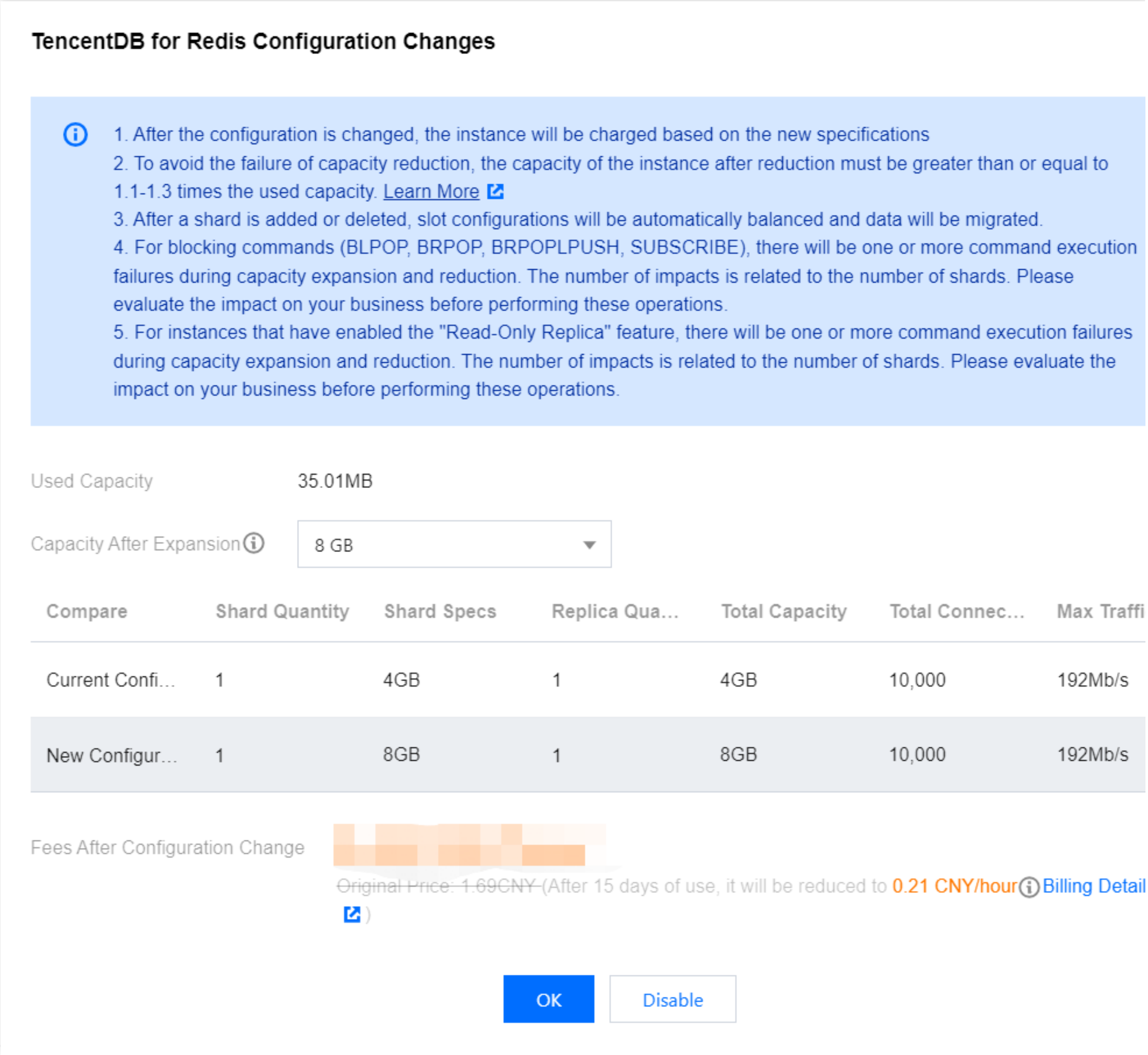
To avoid failure in capacity reduction, the capacity after reduction must be at least 1.3 times the amount of existing data. After the capacity reduction, you will receive an automatic refund.

As a trial version, the 256 MB specification of TencentDB for Redis 4.0 or 5.0 is only suitable for product verification in testing environments. It is available only in the following AZs:
Guangzhou (Zones 6 and 7), Shanghai (Zones 2, 3, 4 and 5), Beijing (Zones 1, 2, 3, 4, 5, 6, and 7), and Shenzhen Finance (Zones 1, 2, and 3). Other 1 GB and above specifications can be smoothly downgraded to the 256 MB specification.

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. In the **Operation** column, expand or reduce nodes or add or delete replicas.

Select **Configure** > **Expand Node** to enter the **TencentDB for Redis Configuration Changes** page and select the desired node capacity.

Select **Configure** > **Reduce Node** to enter the **TencentDB for Redis Configuration Changes** page and select the desired node capacity. The parameters for node reduction are similar to those for node expansion. **Capacity After Expansion** refers to the capacity specification of each shard after reduction. The instance capacity after reduction must be at least 1.3 times the used capacity. You should compare the capacity specification before and after reduction to check whether this requirement is met.



TencentDB for Redis Configuration Changes

1. After the configuration is changed, the instance will be charged based on the new specifications
2. To avoid the failure of capacity reduction, the capacity of the instance after reduction must be greater than or equal to 1.1-1.3 times the used capacity. Learn More
3. After a shard is added or deleted, slot configurations will be automatically balanced and data will be migrated.
4. For blocking commands (BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE), there will be one or more command execution failures during capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the impact on your business before performing these operations.
5. For instances that have enabled the "Read-Only Replica" feature, there will be one or more command execution failures during capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the impact on your business before performing these operations.

Used Capacity 35.01MB

Capacity After Expansion 8 GB

Compare	Shard Quantity	Shard Specs	Replica Qua...	Total Capacity	Total Connec...	Max Traffi
Current Confi...	1	4GB	1	4GB	10,000	192Mb/s
New Configur...	1	8GB	1	8GB	10,000	192Mb/s

Fees After Configuration Change [redacted]

~~Original Price: 1.69CNY~~ (After 15 days of use, it will be reduced to 0.21 CNY/hour Billing Detail)

Parameter	Description
Used Capacity	The used capacity of the current instance.
Min Memory Specification	The minimum memory specification of the current instance required to prevent the disk space from being used up.
Capacity After Expansion	The capacity specification per shard after scaling.
VS	Compare the current configuration with the new configuration, including **Shard Quantity**, **Shard Specs**, **Replica Quantity**, **Total Capacity**, **Total Connections**, and **Max Traffic**.
Configuration	This parameter will be displayed if the billing mode is monthly subscription. You need to pay the

Change Fee	price difference in case of capacity expansion, and subsequent fees will be charged based on the new specification. In case of capacity reduction, the amount to be refunded will be displayed.
Fees after Configuration Change	This parameter will be displayed if the billing mode is pay-as-you-go, indicating the hourly unit price after instance configuration adjustment. You can click **Billing Details** to view the billable items and billing formula and confirm the fees.

Select **Configure** > **Add Replica** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of replicas in the drop-down list next to **Replica Quantity**. Other parameters are similar to those for node expansion. For more information on how to add replicas to a multi-AZ deployed instance, see [Adding Replicas to Multi-AZ Deployed Instance]
[https://www.tencentcloud.com/document/product/239/46556!a7722353282e5e71dad7776e601e41d9).
Select **Configure** > **Delete Replica** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of replicas in the drop-down list next to **Replica Quantity**. Other parameters are similar to those for node expansion.

5. Confirm the configuration adjustment and click **OK**.
6. Return to the instance list. After the instance becomes **Running** in the status , you can use it normally.

Memory Edition (Cluster Architecture)

Note:

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

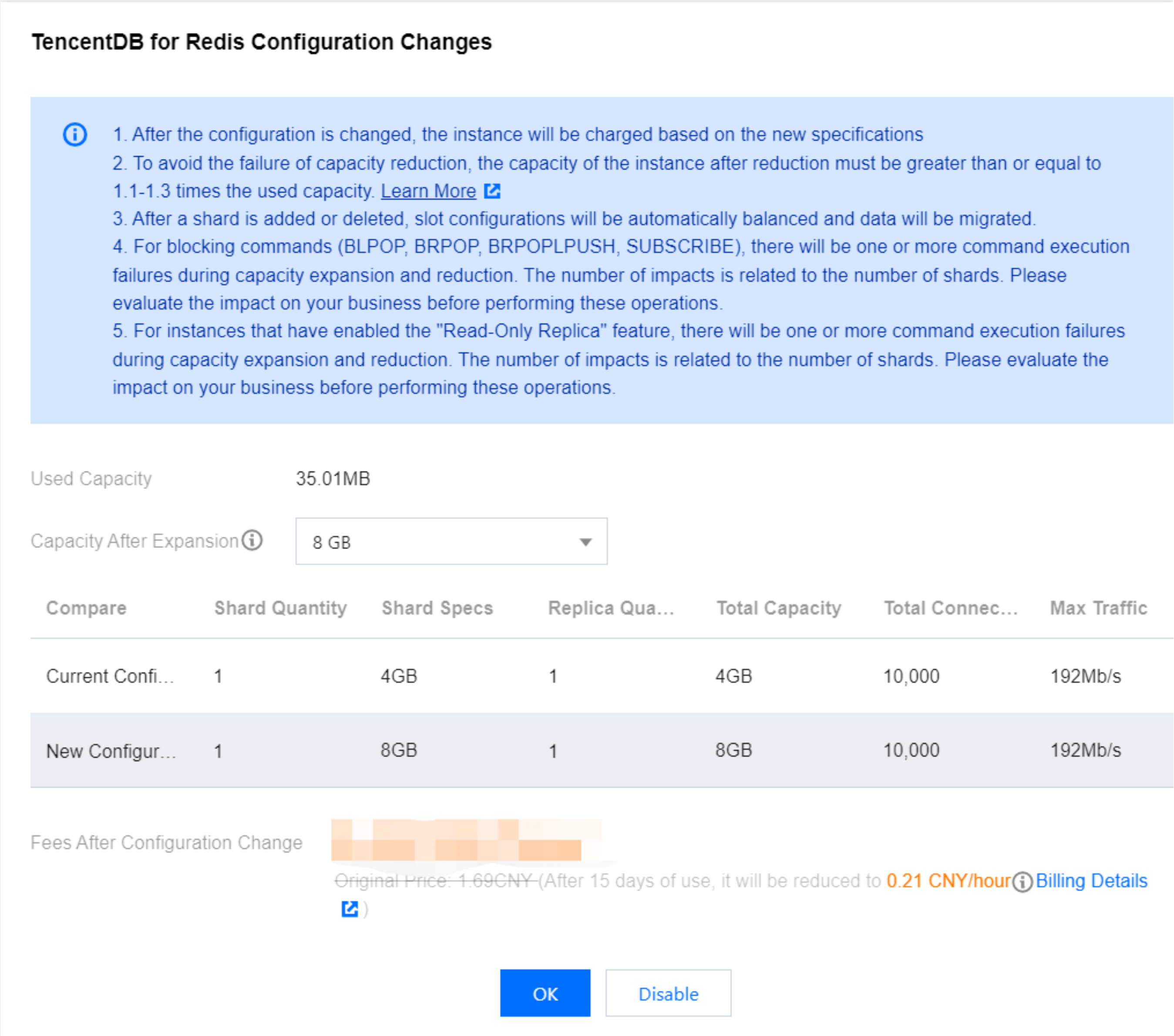
To avoid failure in capacity reduction, the capacity after reduction must be at least 1.3 times the amount of existing data. After the capacity reduction, you will receive an automatic refund.

When shards are added or deleted, the system will automatically balance the slot configuration and migrate data, which may fail in rare cases. We recommend you perform such operations during off-peak hours to avoid the impact of migration on business access.

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. In the **Operation** column, expand or reduce nodes or add or delete replicas or shards.

Select **Configure** > **Expand Node** to enter the **TencentDB for Redis Configuration Changes** page and select the desired node capacity per shard.

Select **Configure** > **Reduce Node** to enter the **TencentDB for Redis Configuration Changes** page and select the desired node capacity per shard. The parameters for node reduction are similar to those for node expansion. **Shard Size** refers to the capacity specification of each shard after reduction. The instance capacity after reduction must be at least 1.3 times the used capacity. You should compare the capacity specification before and after reduction to check whether this requirement is met.



TencentDB for Redis Configuration Changes

1. After the configuration is changed, the instance will be charged based on the new specifications
2. To avoid the failure of capacity reduction, the capacity of the instance after reduction must be greater than or equal to 1.1-1.3 times the used capacity. Learn More
3. After a shard is added or deleted, slot configurations will be automatically balanced and data will be migrated.
4. For blocking commands (BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE), there will be one or more command execution failures during capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the impact on your business before performing these operations.
5. For instances that have enabled the "Read-Only Replica" feature, there will be one or more command execution failures during capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the impact on your business before performing these operations.

Used Capacity 35.01MB

Capacity After Expansion 8 GB

Compare	Shard Quantity	Shard Specs	Replica Qua...	Total Capacity	Total Connec...	Max Traffic
Current Confi...	1	4GB	1	4GB	10,000	192Mb/s
New Configur...	1	8GB	1	8GB	10,000	192Mb/s

Fees After Configuration Change [redacted]

~~Original Price: 1.69CNY~~ (After 15 days of use, it will be reduced to 0.21 CNY/hour Billing Details)

Parameter	Description
Used Capacity	The used capacity of the current cluster instance.
Min Memory Specification	The minimum memory specification per shard of the current cluster instance required to prevent the disk space from being used up.
Shard Size	The capacity specification per shard after scaling.
VS	Compare the current configuration with the new configuration, including **Shard Quantity**, **Shard Specs**, **Replica Quantity**, **Total Capacity**, **Total Connections**, and **Max Traffic**.

Configuration Change Fee	This parameter will be displayed if the billing mode is monthly subscription. You need to pay the price difference in case of capacity expansion, and subsequent fees will be charged based on the new specification. In case of capacity reduction, the amount to be refunded will be displayed.
Fees after Configuration Change	This parameter will be displayed if the billing mode is pay-as-you-go, indicating the hourly unit price after instance configuration adjustment. You can click Billing Details to view the billable items and billing formula and confirm the fees.

Select **Configure** > **Add Replica** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of replicas in the drop-down list next to **Replica Quantity**. Other parameters are similar to those for node expansion. For more information on how to add replicas to a multi-AZ deployed instance, see Adding Replicas to Multi-AZ Deployed Instance.

Select **Configure** > **Delete Replica** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of replicas in the drop-down list next to **Replica Quantity**. Other parameters are similar to those for node expansion.

Select **Configure** > **Add Shard** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of shards in the drop-down list next to **Shard Quantity**. Other parameters are similar to those for node expansion.

Select **Configure** > **Delete Shard** to enter the **TencentDB for Redis Configuration Changes** page. Select the desired number of shards in the drop-down list next to **Shard Quantity**. Other parameters are similar to those for node expansion.

5. Confirm the configuration adjustment and click **OK**.
6. Return to the instance list. After the instance becomes **Running** in the status , you can use it normally.

Related APIs

API Name	Description
UpgradeInstance	Upgrades the instance configuration

Adjusting the Number of Connections

Last updated：2024-03-13 11:29:36

Overview

If the current database specification can‘t sustain massive concurrent application requests due to insufficient connections, **Connection Utilization** may get too high. To handle such access spikes, you can directly increase the number of maximum connections in the console.

Notes

A single shard can sustain up to 10,000 connections by default, and the maximum number of connections to the entire instance is the maximum number of connections per shard multiplied by the shard quantity. A standard architecture instance has only one shard.

When you adjust the number of connections, the value range per shard is as detailed below:

If the read-only replica feature is disabled

The maximum number of connections to each shard can be 10000–40000.

If the read-only replica feature is enabled

The maximum number of connections to each shard can be 10000–10000 * (replica quantity + 3).

Notes

Increasing the maximum number of connections has no impact on the business.

If the maximum number of connections is decreased, new connections may fail to be established when the number of connections reaches the upper limit.

If the problem persists after you increase the maximum number of connections, contact the aftersales service or submit a ticket for assistance.

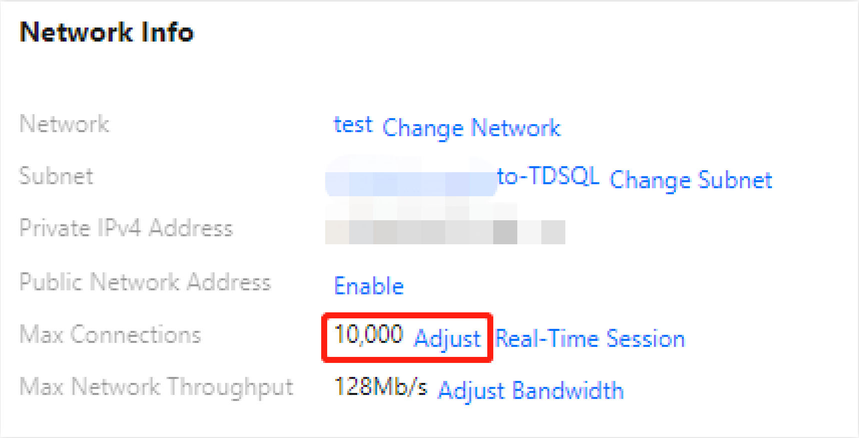
Prerequisites

You have created a TencentDB for Redis instance.

The database instance is in **Running** status, with no ongoing tasks.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Network Info** section on the **Instance Details** page, you can view the instance's current maximum number of connections after **Max Connections**. Click **Adjust**.

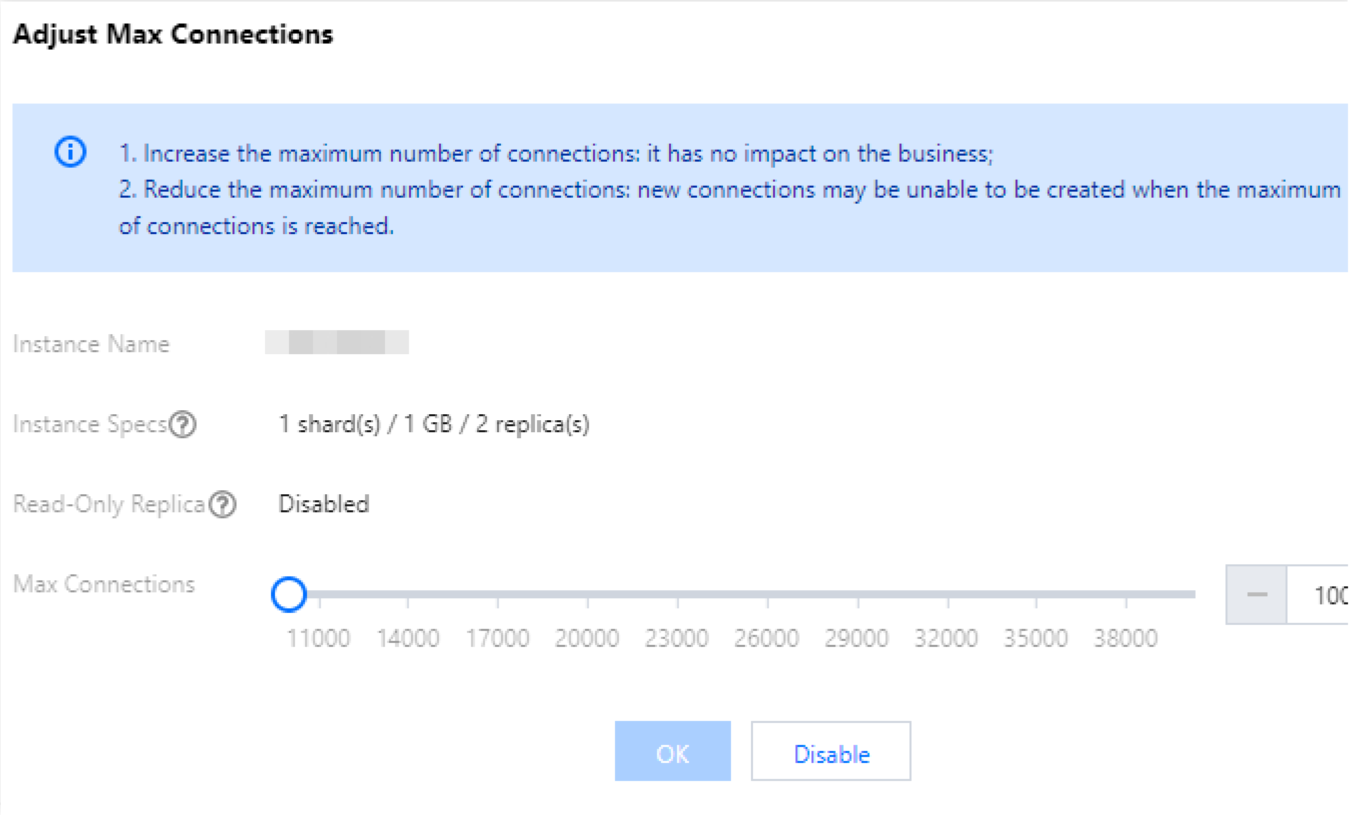


Network Info

Network	test Change Network
Subnet	[redacted] to-TDSQL Change Subnet
Private IPv4 Address	[redacted]
Public Network Address	Enable
Max Connections	10,000 Adjust Real-Time Session
Max Network Throughput	128Mb/s Adjust Bandwidth

6. In the **Adjust Max Connections** window, confirm the instance information and specification and increase the value.

Standard architecture



Adjust Max Connections

1. Increase the maximum number of connections: it has no impact on the business;
2. Reduce the maximum number of connections: new connections may be unable to be created when the maximum of connections is reached.

Instance Name [redacted]

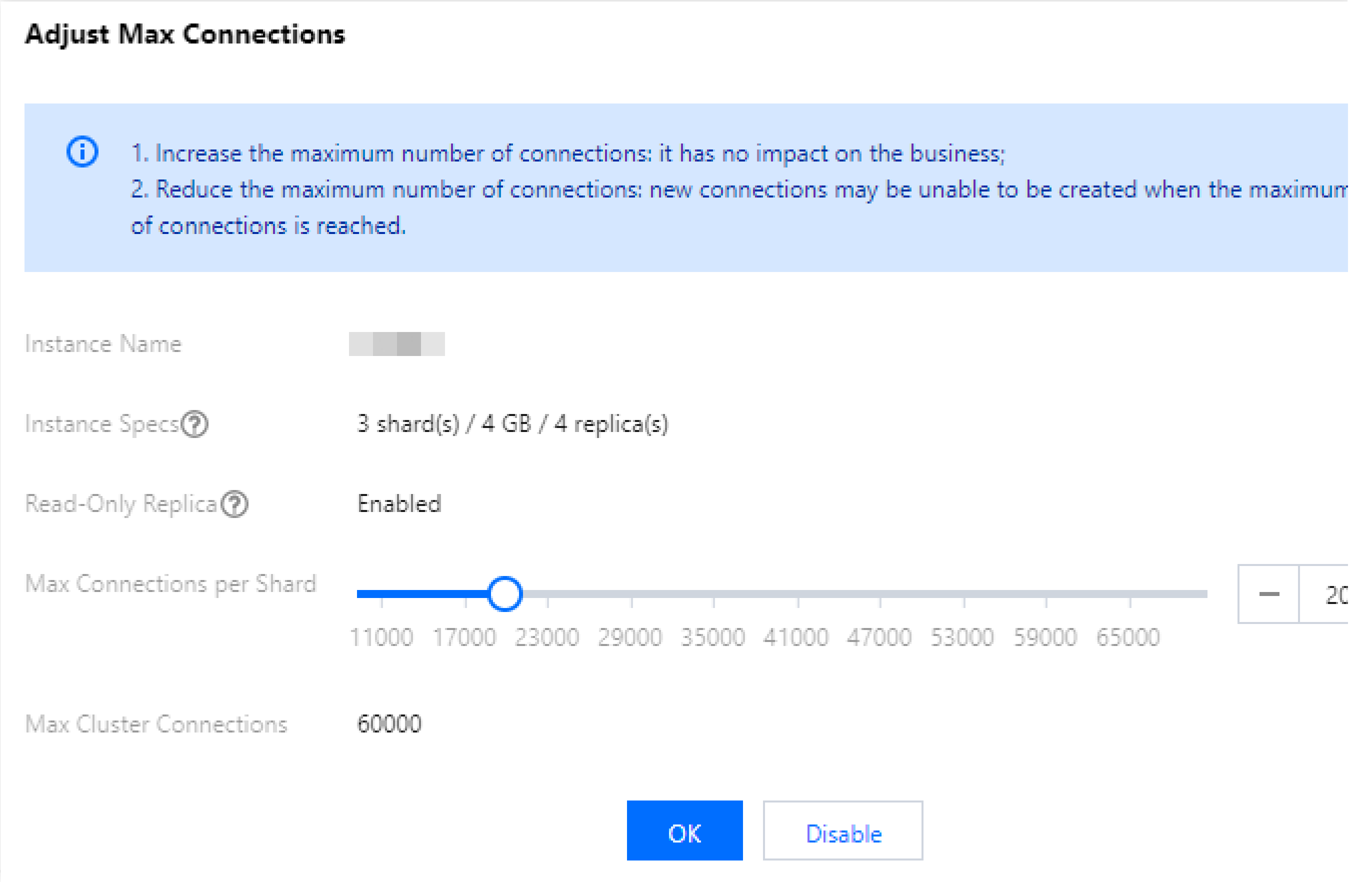
Instance Specs 1 shard(s) / 1 GB / 2 replica(s)

Read-Only Replica Disabled

Max Connections 10C

Parameter	Description
Instance Name	Instance name information.
Instance Specs	Instance specification information, including shard quantity, total memory, and replica quantity. A standard architecture instance has only one shard.
Read-Only Replica	Set whether the read-only replica feature (read/write separation) is enabled.
Max Connections	Adjust the maximum number of connections on the slider.

Cluster architecture



Adjust Max Connections

1. Increase the maximum number of connections: it has no impact on the business;
2. Reduce the maximum number of connections: new connections may be unable to be created when the maximum of connections is reached.

Instance Name [redacted]

Instance Specs 3 shard(s) / 4 GB / 4 replica(s)

Read-Only Replica Enabled

Max Connections per Shard 20

Max Cluster Connections 60000

Parameter	Description
Instance Name	Instance name information.
Instance Specs	Instance specification information, including shard quantity, shard capacity, and replica quantity.
Read-Only Replica	Set whether the read-only replica feature (read/write separation) is enabled.
Max Connections per Shard	Adjust the maximum number of connections to each shard on the slider.
Max Cluster Connections	The maximum number of connections to the entire instance is calculated automatically, which is the maximum number of connections per shard multiplied by the shard quantity.

7. Click **OK**. On the left sidebar, you can click **Task Management** to view the task progress. After the task is executed, you can view the new maximum number of connections to the entire instance after **Max Connections** in the **Network Info** section on the **Instance Details** page.

Related APIs

API	Description
describeInstances	Queries the list of instances.

FAQs

If you find that **Connection Utilization** in **System Monitoring** is too high, adjust the maximum number of connections as instructed in High Connection Utilization.

Enabling/Disabling Read/Write Separation

Last updated：2023-10-20 10:57:44

Overview

TencentDB for Redis supports read/write separation for business scenarios with more reads but less writes, which can well cope with read requests concentrating on frequently read data.

Billing

The read-only replica feature is currently in free trial.

Note

Enabling read/write separation may cause data read inconsistency (the data on replica nodes are older than that on the master node). Therefore, you need to check whether your business can tolerate data inconsistency first.

Disabling read/write separation may interrupt existing connections momentarily, so you are recommended to do so during off-peak hours.

Prerequisites

The database instance is on v4.0 or later.

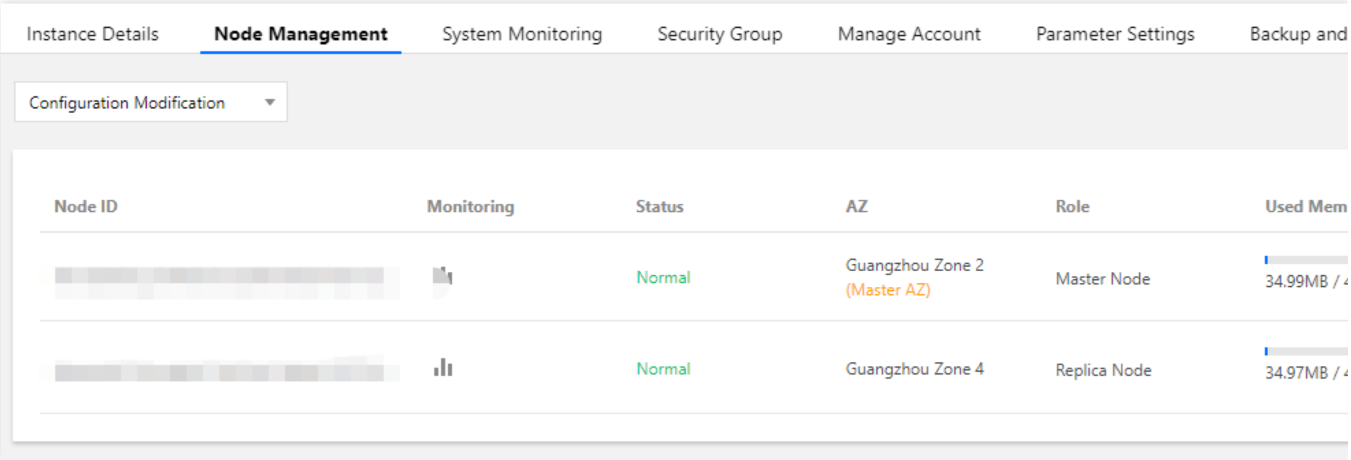
The database instance is in **Running** status.

Directions

Enabling Read/Write Separation

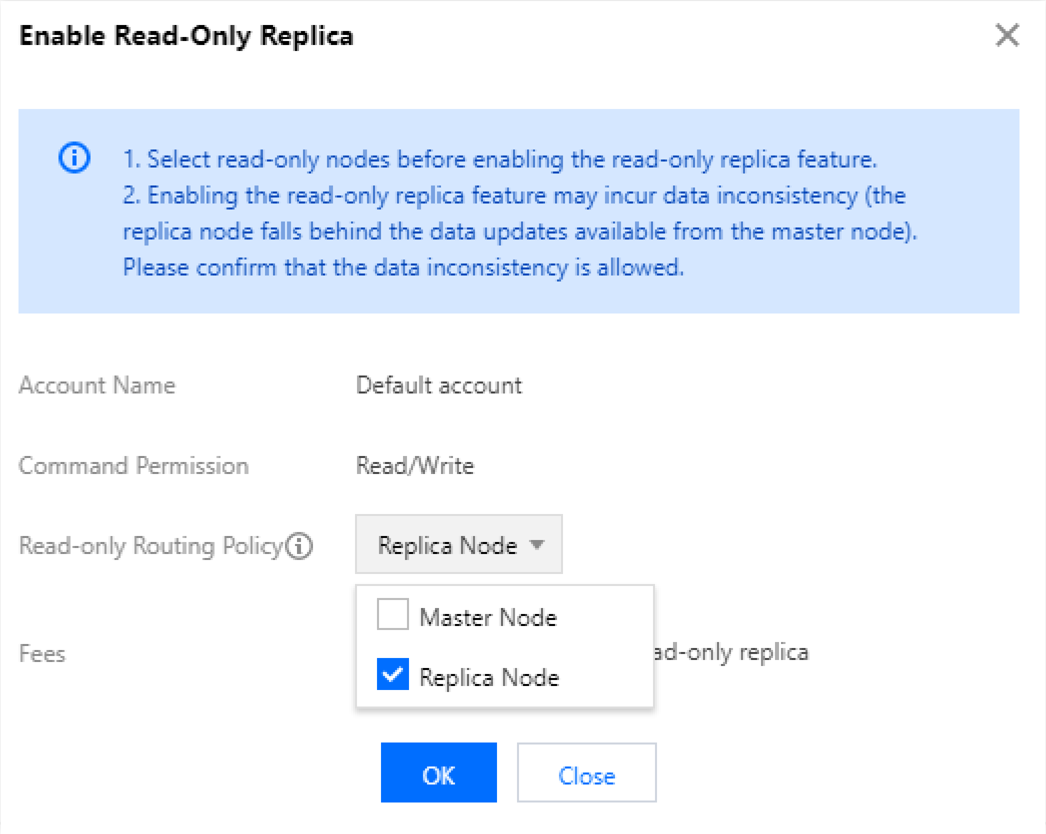
1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the **Instance ID** to enter the **Instance Details** page and click the **Node Management** tab.
5. In the upper right corner of the **Node Management** page, click

next to **Read-Only Replica**.



Node ID	Monitoring	Status	AZ	Role	Used Mem
[redacted]		Normal	Guangzhou Zone 2 (Master AZ)	Master Node	34.99MB /
[redacted]		Normal	Guangzhou Zone 4	Replica Node	34.97MB /

6. In the pop-up window, configure read-only replica nodes. The parameters are as detailed below:



Enable Read-Only Replica

1. Select read-only nodes before enabling the read-only replica feature.
2. Enabling the read-only replica feature may incur data inconsistency (the replica node falls behind the data updates available from the master node). Please confirm that the data inconsistency is allowed.

Account Name: Default account

Command Permission: Read/Write

Read-only Routing Policy: Replica Node

□ Master Node
☑ Replica Node

Fees: ad-only replica

Parameter	Description
Account ID	It is fixed to the **Default account**; that is, the system can enable read-only replicas only for the default account.
Command Permission	It is fixed to **Read/Write**; that is, the default account has the read/write permissions.

Read-Only Routing Policy	It is **Replica Node** by default. You can also select **Master Node** or select both **Replica Node** and **Master Node**. Read requests are automatically routed to the configured read-only nodes by load balancing.
Read Local Nodes Only	This parameter will be displayed if the instance is deployed in multiple AZs. It is fixed to **Disabled**. You can configure the `read-local-node-only` parameter on the**Parameter Settings** page in the console to enable/disable this feature.
Fees	This feature is current in free trial.

7. After confirming that the parameter settings are correct, click **OK**.
8. The **Instance Status** will change to **Processing**. Wait for it to change to **Running**. Then, in the **Specs Info** section on the **Instance Details** page, you can see that **Read-Only Replica** is **Enabled** and try out read/write separation.

Disabling Read/Write Separation

Note

If the read-only routing policy of the instance's custom account specifies that read requests are routed to the replica node, the read-only replica feature cannot be directly disabled. You need to change it to route to the master node, and then disable the read-only replica feature.

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the **Instance ID** to enter the **Instance Details** page and click the **Node Management** tab.
5. In the upper right corner of the **Node Management** page, click next to **Read-Only Replica**.
6. In the **Disable Read-Only Replica** window, read the note on the impact of disabling read-only replica and click **OK**.
7. The **Instance Status** will change to **Processing**. Wait for it to change to **Running**. Then, in the **Specs Info** section on the **Instance Details** page, you can see that **Read-Only Replica** is **Disabled**.

Related APIs

API Name	Description
EnableReplicaReadonly	Enables read/write separation
DisableReplicaReadonly	Disables read/write separation

Clearing Instances

Last updated：2023-10-20 10:57:18

Overview

TencentDB for Redis supports quickly clearing all instance data in the console. Clearing an instance will perform the **FLUSHALL** operation on the instance, and all instance data will be cleared and cannot be recovered. Proceed with caution.

Note

Once cleared, the data cannot be recovered. Make sure you have backed up all data before submitting a clearing request.
The database access will be blocked during data clearup. If a large amount of data is requested, the database will be disconnected and cannot provide services.
When a master instance in a global replication group is cleared up, all of the other instances in the group are cleared up as well.
In a global replication group, you can't clear up a read-only instance as clearup is a write operation. You can clear up the group after removing all read-only instances from it.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the top-right corner of the **Instance Details** page, click **Clear Instance**.
6. In the **Clear Instance** dialog box, learn about the impact of instance clearing, enter the instance access password in the input box next to **Password**, and click **OK**.
7. On the left sidebar, select **Task Management** and wait for the task to complete.

Related APIs

API Name	Description

ClearInstance	Clears a Redis instance

Returning and Isolating Instance

Last updated：2023-04-03 16:08:46

Overview

If you no longer need pay-as-you-go instances and your Tencent Cloud account has no overdue payments, you can directly terminate them in the console to avoid further fee deduction. Terminated instances are retained in the recycle bin for two hours, and you can start them up to restore them. After two hours, the system will directly eliminate them, and all their data will be permanently deleted.

Notes

After an instance is returned, once its status changes to **Isolated** or **To be deleted**, it will no longer incur fees.

Notes

After the instance is terminated, all its data will be cleared and cannot be recovered. Be sure to back up your data first before submitting a termination task.

When the instance is terminated, its IP resources will be released simultaneously.

Refund policy

After an instance is terminated, the refund procedures are as detailed below:

For instances that met the 5-day no-questions-asked refund policy, the payment will be returned to your Tencent Cloud account.

For normal instances, the payment will be returned to your Tencent Cloud account by the proportion of the cash and gift cards paid for the purchase.

For orders from promotional reward channel, the refund will be charged 25% of their actual cash payment amount. Currently, self-service refund is unavailable for such kind of orders, you can contact us to apply for the refund.

For more information on self-service refund and examples.

Terminating a pay-as-you-go instance

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.

3. In the **Instance List**, select the target pay-as-you-go instance and click **More** > **Terminate** in its **Operation** column.

4. In the **Terminate Instance** pop-up window, confirm the information of the target instance, understand the impact of instance termination, and click **Terminate**.
5. On the left sidebar, select **Recycle Bin**, and you can see that the terminated pay-as-you-go instance is isolated there. The instance is in **Isolated** status and will no longer incur fees.

6. (Optional) Click **Start Up** to restore the instance, or click **Eliminate Now** to directly eliminate it. For detailed directions, see Restoring Isolated Instance.

Related APIs

API	Description
DestroyPostpaidInstance	Terminates a pay-as-you-go instance.
DestroyPrepaidInstance	Returns a monthly subscribed instance.

Restoring Isolated Instance

Last updated：2022-07-25 17:24:54

Overview

Tencent Cloud recycle bin is a cloud service repossession mechanism. When you return an instance or your account balance is insufficient to pay the fees of a pay-as-you-go instance, the instance will be moved to the recycle bin. You can restore the pay-as-you-go instance from the recycle bin in the retention period.

Pay-as-you-go instance repossession mechanism

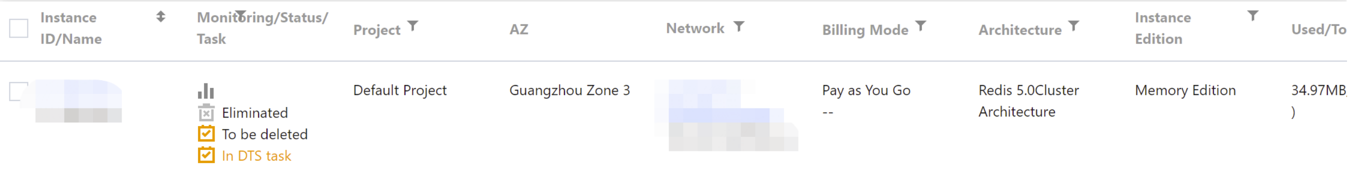
If you no longer need pay-as-you-go instances and your Tencent Cloud account has no overdue payments, you can directly return them into the recycle bin as instructed in Returning and Isolating Instance. Terminated instances are retained in the recycle bin for two hours, and you can start them up to restore them. After two hours, the system will directly eliminate them, and all their data will be deleted and cannot be recovered.
For pay-as-you-go instances, billing will continue within 24 hours after your account balance drops below 0. After 24 hours, the instances will be automatically moved to the recycle bin for isolation, billing will stop, and you won't be able to use instance resources. In case of overdue payments, the instances will be retained in the recycle bin for 24 hours. If you top up your account within 24 hours, you can restore instance resources; otherwise, the system will automatically terminate them after 24 hours, and all data will be cleared and cannot be recovered.

Prerequisites

The instance is isolated in the recycle bin.
Your Tencent Cloud account balance is sufficient.

Restoring one isolated instance

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Redis** > **Recycle Bin**.
3. Above the instance list on the right, select the region.
4. On the **Recycle Bin** page on the right, you can see the list of instances in the recycle bin, all of which are in the **Isolated** status.



□ Instance ID/Name	Monitoring/Status/Task	Project	AZ	Network	Billing Mode	Architecture	Instance Edition	Used/To
□ [redacted]	Eliminated To be deleted In DTS task	Default Project	Guangzhou Zone 3	[redacted]	Pay as You Go --	Redis 5.0Cluster Architecture	Memory Edition	34.97MB)

5. In the instance list in the recycle bin, find the target instance and click **Start Up** in the **Operation** column.

6. Confirm the instance information and restore it.

If the instance is pay-as-you-go, confirm the instance information and click **OK** in the **Start Instance** window to restore it.

Note:

As TencentDB for Redis is an in-memory database, to use the batch instance restoration feature, submit a ticket for application.

Related APIs

API	Description
StartupInstance	Restores an isolated instance.

Eliminating Instance

Last updated：2023-05-23 10:48:36

Overview

When you return an instance or your account balance is insufficient to pay the fees of a pay-as-you-go instance, the instance will be moved to the recycle bin. If you have backed up your data and you are sure that you don't need the instance any more, you can release all its resources during the retention period to avoid resource waste.

Prerequisites

The instance is isolated in the recycle bin, and the data has been backed up.

The instance is no longer needed.

Directions

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Redis** > **Recycle Bin**.
3. Above the instance list on the right, select the region.
4. On the **Recycle Bin** page on the right, you can see the list of instances in the recycle bin, all of which are in the **Isolated** status.
5. In the instance list in the recycle bin, find the target instance and click **Eliminate Now** in the **Operation** column.
6. In the **Eliminate Now** window, confirm the instance information and click **OK**.

Note

The instance will be completely eliminated, and its data will not be recoverable. Therefore, you need to back up the data in advance.

Related APIs

API Name	Description
CleanUpInstance	Deactivates an instance in the recycle bin immediately

Instance Upgrade

Upgrading Instance Version

Last updated：2023-03-14 15:40:37

Overview

TencentDB for Redis is compatible with Redis 2.8, 4.0, 5.0, and 6.2. Upgrade to a compatible version and minor version upgrade are supported, so that you can upgrade your instance to a newer version for more features.

Version differences

Compatible Version	Minor Version	Optimizations and Fixes
Redis 4.0	4.3.0	When a failed replica node is discovered in the cluster, messages can be sent to the cluster, making it quicker to locate faulty nodes. Performance optimization: `zmalloc_get_rss()` is executed in the BIO thread to avoid blocking the main thread and increasing the request latency. Fixed the issue where the `rdbLoadRio()` function might trigger a crash in some cases.
Redis 5.0	5.2.0	Performance optimization: `zmalloc_get_rss()` is executed in the BIO thread to avoid blocking the main thread and increasing the request latency. Fixed the issue where the `rdbLoadRio()` function might trigger a crash in some cases.
Redis 6.2	6.2.5	Supported Redis 6.2.

Upgrade description

Currently, only standard architecture instances can be upgraded to a compatible version, while cluster architecture instances cannot.

Instances can be upgraded from an earlier version to a later one; for example, you can upgrade from Redis 4.0 to 5.0.

Cross-version upgrade is supported.

If an instance is upgraded to a compatible version, no billing changes will be caused.

Downgrade to a compatible version is not supported.

During minor version upgrade, the system automatically detects the minor version, and you cannot select a target version.

As the version release time varies by region, the minor version release status is as displayed in the console.

How upgrade works

1. Apply for resources: Apply for the resources of the new instance version, including proxy, Redis master node, and Redis replica node resources.
2. Sync the data: Sync the full and incremental data from the instance on the old version to the instance on the new version.
3. Wait for the switch: Wait until data sync is completed or wait for a switch window.
4. Switch the instances: When the switch conditions are met (data sync is almost completed, and the requirements for the switch window are met), stop writing data into the old instance, unbind the virtual IP (VIP) address from it, and bind the VIP to the new instance.
5. Complete the upgrade: Update the instance status.

Upgrade impact

The version upgrade process mainly consists of data sync and instance switch:

During data sync, the service will not be affected.

During switch, the instances will become read-only for less than 1 minute (to wait for the completion of data sync), and a momentary disconnection (within seconds) will occur; therefore, your business should have an automatic reconnection mechanism.

Preparations for upgrade

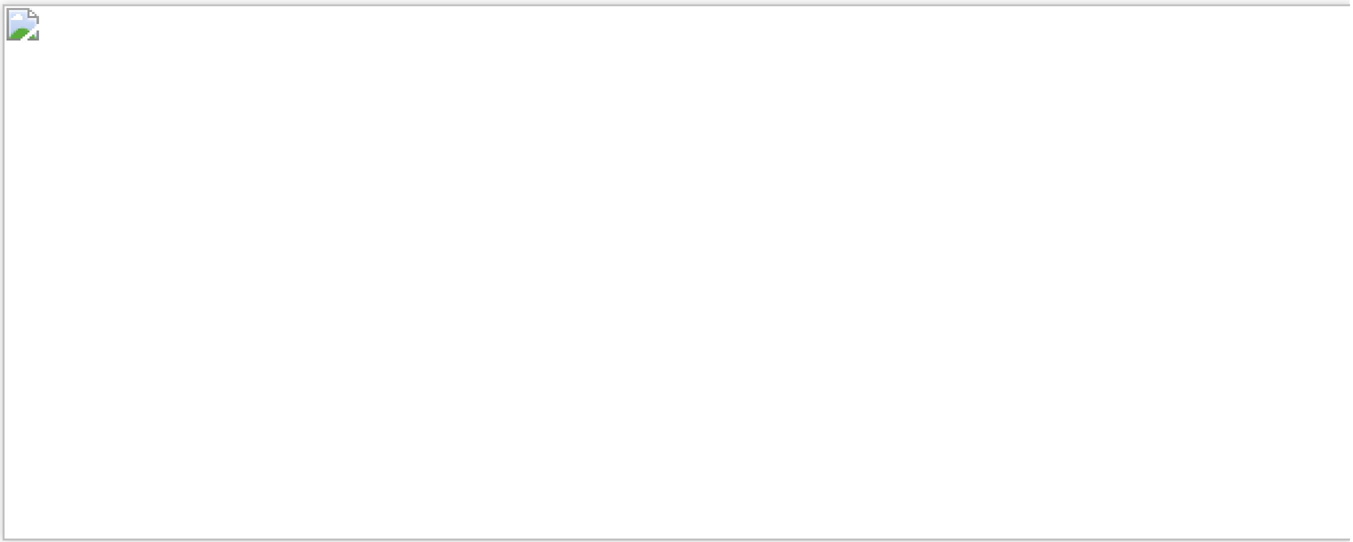
The instance to be upgraded is in **Running** status and is not executing any tasks.

The target version is confirmed.

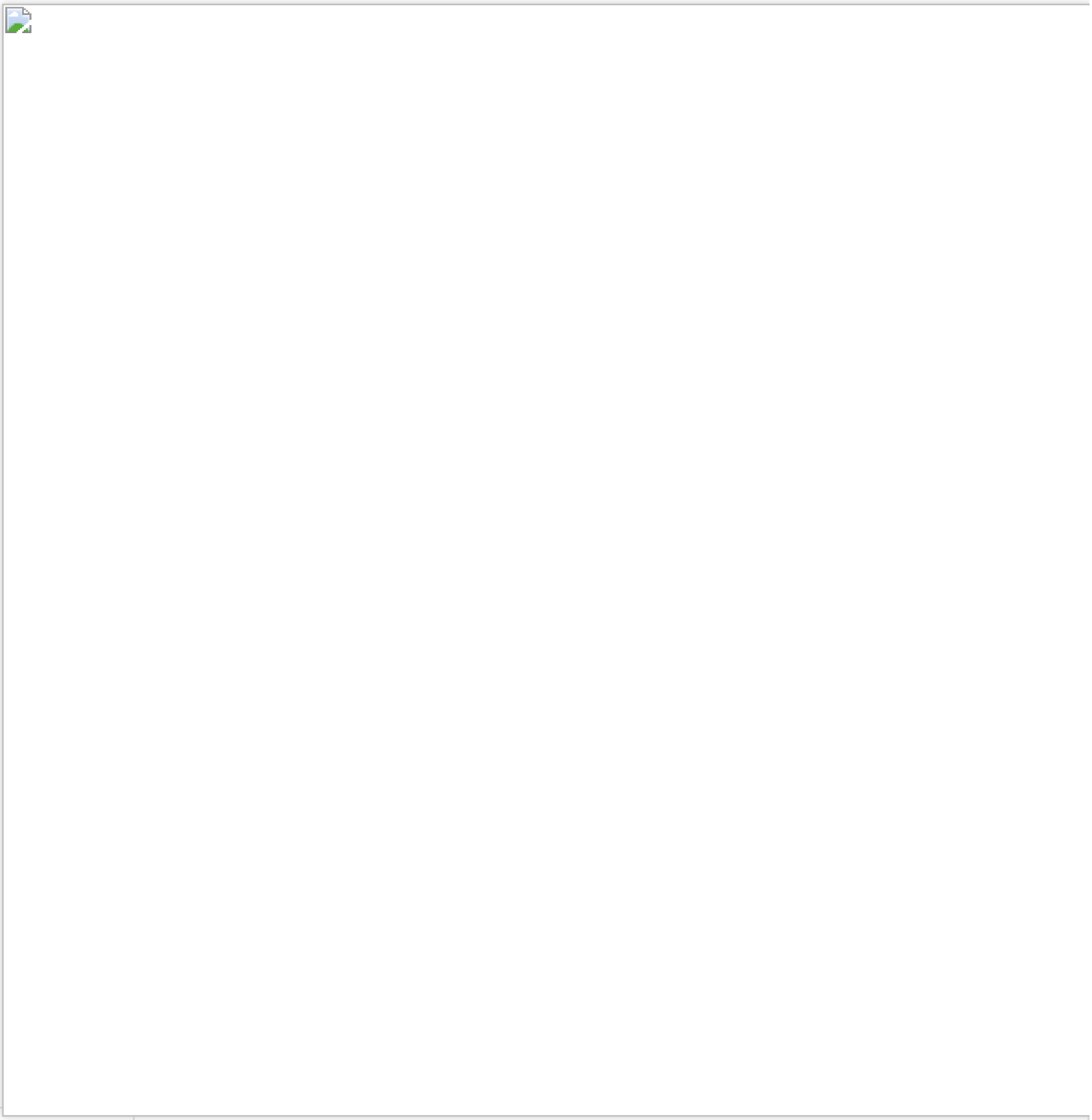
Upgrading the version

1. Log in to the TencentDB for Redis console.

2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Specs Info** section on the **Instance Details** page, click **Upgrade Version** after **Compatible Version**.



6. In the pop-up window, confirm the information of the target instance based on the following table, configure the target version, and click **OK**.



Parameter	Description
Instance ID	ID of the instance to be upgraded.
Instance Name	Name of the instance to be upgraded.
Compatible Version	The current compatible Redis version of the instance to be upgraded.
Architecture	Architecture information of the instance to be upgraded. Currently, version upgrade is supported only for standard architecture instances.

Memory	Memory size of the instance to be upgraded.
Version Upgrade	Select the target version in the drop-down list. You can upgrade from an earlier version to a later version or across versions.
Preview New Specs	Preview information of the instance specifications after upgrade.
Switch Time	**Switch Now**: The switch will be performed when the data sync is almost completed (the data left to be synced is less than 10 MB). **Switch in Maintenance Time**: The switch will be performed during the instance maintenance time. If the switch conditions cannot be met in the current maintenance time, the switch will be attempted in the next maintenance time. You can modify the **Maintenance Window** on the instance details page.
Total Fees	Fees after instance upgrade. No billing changes will be caused.

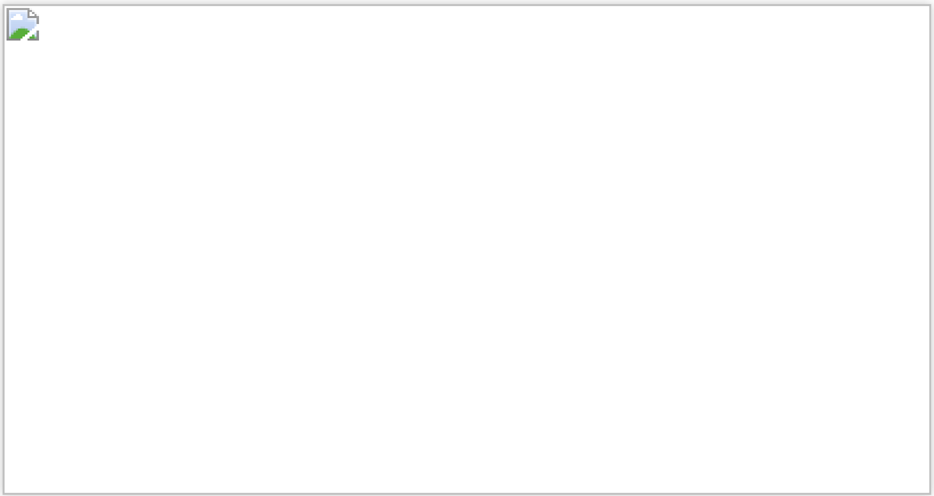
7. On the left sidebar, select **Task Management**, wait for the task to complete, and you can see that the version of the instance has been upgraded in the instance list.

Upgrading the minor version

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Specs Info** section on the **Instance Details** page, click **Upgrade Minor Version** after **Compatible Version**.

Notes

The system automatically detects the minor version. If the **Upgrade Minor Version** button is grayed out, the instance is already on the latest minor version.



6. In the **Upgrade Minor Version** window, confirm the instance information and the target version and select the upgrade time in **Switch Time**.

Switch Now: The switch will be performed when the data sync is almost completed (the data left to be synced is less than 10 MB).

Switch in Maintenance Time: The switch will be performed during the instance maintenance time. If the switch conditions cannot be met in the current maintenance time, the switch will be attempted in the next maintenance time. You can modify the **Maintenance Window** on the instance details page.

7. On the left sidebar, select **Task Management**, wait for the task to complete, and you can see that the minor version of the instance has been upgraded in the instance list.

Related APIs

API	Description
UpgradeInstanceVersion	Upgrades the version of an instance
UpgradeSmallVersion	Upgrades the minor version of an instance

Upgrading Instance Architecture

Last updated：2023-04-19 14:32:16

Overview

TencentDB for Redis supports standard architecture and cluster architecture. To help you process ever-growing business data, it allows you to upgrade from standard architecture to cluster architecture if the performance and capacity of standard architecture are insufficient.

Upgrade description

For Redis 4.0 or later, a standard architecture instance can be upgraded to a cluster architecture instance on the same version; for example, you can upgrade from Redis 4.0 Standard Architecture to Redis 4.0 Cluster Architecture. Cross-version architecture upgrade is not supported; for example, you cannot upgrade from Redis 4.0 Standard Architecture to Redis 5.0 Cluster Architecture.

The architecture of Redis 2.8 cannot be upgraded.

Cluster architecture cannot be downgraded to standard architecture.

Cross-AZ architecture upgrade is not supported.

Cluster architecture upgrade is not supported for pay-as-you-go instances.

After standard architecture is upgraded to cluster architecture, fees will be charged based on cluster architecture and thus get increased. For more information, see Pricing.

How upgrade works

Redis standard architecture can be directly upgraded to cluster architecture (single-shard) in three minutes with no data migration required.

In Redis 4.0 or later, if standard architecture is upgraded to cluster architecture, only the runtime mode of the instance will change from having no slot limit to having one, but no data migration will occur.

Upgrade preparations (compatibility check)

To avoid business failures caused by compatibility problems during migration to cluster architecture, check the compatibility before the upgrade:

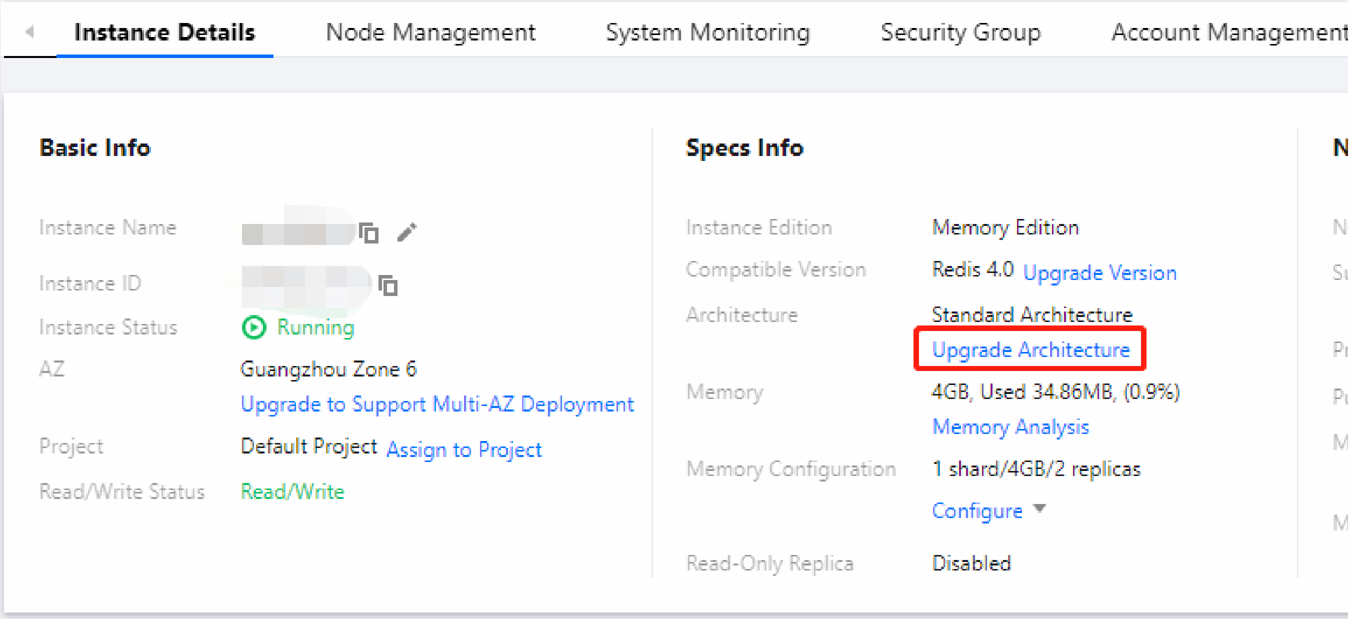
Cluster architecture stores data in a distributed manner, and its biggest difference from standard architecture lies in whether a single command supports multikey access. For the cluster architecture, commands can be categorized into supported, partially supported, and unsupported. For the complete list of compatible commands, see Overview.
For more information about compatibility check, see Check on Migration from Standard Architecture to Cluster Architecture.

Upgrade impact

Generally, upgrade can be completed in three minutes.
During the upgrade, existing connections will be closed momentarily; therefore, your business should have a reconnection mechanism.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. Click the instance ID to enter the **Instance Details** page.
5. In the **Specs Info** section on the **Instance Details** page, click **Upgrade Architecture** after **Architecture**.



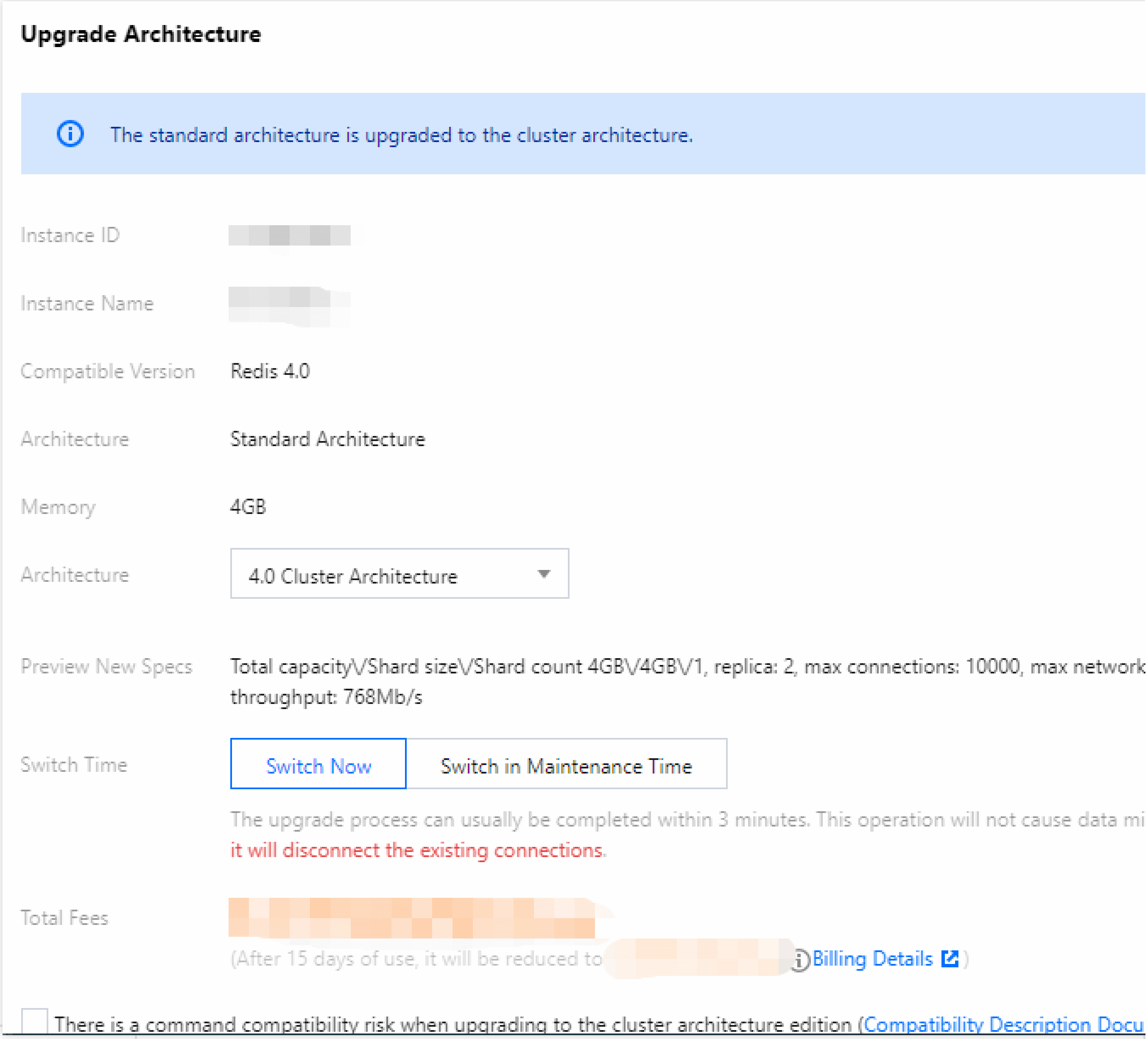
Basic Info

Instance Name [redacted]
Instance ID [redacted]
Instance Status Running
AZ Guangzhou Zone 6
Project Default Project
Read/Write Status Read/Write

Specs Info

Instance Edition Memory Edition
Compatible Version Redis 4.0
Architecture Standard Architecture
Memory 4GB, Used 34.86MB, (0.9%)
Memory Configuration 1 shard/4GB/2 replicas
Read-Only Replica Disabled

6. In the pop-up window, configure the upgrade parameters according to the table below.



Upgrade Architecture

ⓘ The standard architecture is upgraded to the cluster architecture.

Instance ID	[redacted]
Instance Name	[redacted]
Compatible Version	Redis 4.0
Architecture	Standard Architecture
Memory	4GB
Architecture	4.0 Cluster Architecture ▼
Preview New Specs	Total capacity\/Shard size\/Shard count 4GB\/4GB\/1, replica: 2, max connections: 10000, max network throughput: 768Mb/s
Switch Time	Switch Now / Switch in Maintenance Time The upgrade process can usually be completed within 3 minutes. This operation will not cause data mi it will disconnect the existing connections.
Total Fees	[redacted] (After 15 days of use, it will be reduced to [redacted] ⓘBilling Details ☐)

□ There is a command compatibility risk when upgrading to the cluster architecture edition (Compatibility Description Docu

Parameter	Description
Instance ID	ID of the instance to be upgraded.
Instance Name	Name of the instance to be upgraded.
Compatible Version	The current compatible Redis version of the instance to be upgraded.
Architecture	Architecture information of the instance to be upgraded.
Memory	Memory size of the instance to be upgraded.
Architecture	Select the target architecture in the drop-down list. Currently, the standard architecture can be upgraded to the cluster architecture, but the cluster architecture cannot be downgraded to the

	standard architecture.
Preview New Specs	Preview information of the instance specifications after upgrade.
Switch Time	**Switch Now**: The switch will be performed immediately. **Switch in Maintenance Time**: The switch will be performed during the instance maintenance time. You can modify the **Maintenance Time** on the instance details page. We recommend that you switch during off-peak hours.
Total Fees	Fees after the architecture upgrade. Pay-as-you-go: The hourly unit price after instance architecture upgrade. You can click **Billing Details** to view the billable items and billing formula and confirm the fees. Monthly subscription: The total fees of the instance before it expires after the architecture is upgraded.

7. There is a command compatibility risk when upgrading to the cluster architecture edition. Click **Compatibility Description Document**, confirm the compatibility risk, click **There is a command compatibility risk when upgrading to the cluster architecture edition (**Compatibility Description Document**), I've confirmed the compatibility risk and agreed to upgrade.**, and click **OK** to proceed with the upgrade.
8. On the order page, confirm the order information and the fees to be paid, and hover over ⓘ to view the detailed calculation of the fees. After confirming that everything is correct, click **Submit Order**. Make the payment and return to the instance list. After the **Instance Status** changes to **Running**, you can see that the instance architecture has been upgraded to cluster architecture in the instance list or instance details.

Related APIs

API	Description
UpgradeInstanceVersion	Upgrades the instance architecture

Upgrading Proxy

Last updated：2023-10-20 10:56:00

Overview

Minor versions of the TencentDB for Redis proxy are released from time to time to add more database features or fix known bugs.

Proxy Version	Proxy Minor Version	New Feature, Optimization, or Fix
Proxy 5.0	5.6.3	Supported `getex` , `auth name pwd` , and `xautoclaim` commands.
	5.6.0	Supported `wait` command in Cluster Architecture instances. Supported SSL encryption to implement encrypted data transfer.
	5.5.0	Supported `wait` commands in Cluster Architecture instances. Supported the “Read Local Nodes Only” feature. Supported the `dbsize` command in Cluster Edition instances to return the number of keys in all shards. Supported displaying the client port information in slow logs. Supported `flushall` and `flushdb` commands, which can be distributed to the master node of all shards in a Cluster Architecture instance while retaining data in nodes with the specified `nodeid` . Supported monitoring the number of big value requests. Supported the `Scan` command in Cluster Edition instances to traverse all shards. Fixed the issue where "ERR unknown command 'select' command" might be returned when the `select` command was executed after a transaction. Fixed the issue where the command was sent to an incorrect node and the `Move` error was reported when the locked connection wasn't released in time as the `watch+` transaction was used in the pipeline scenario.
	5.4.0	Optimized the statistics collection policies of P99 monitoring metrics, including metrics for all Redis commands.
	5.2.0	Supported the five-second granularity for monitoring data.
	5.1.0	Supported `keys` command in Cluster Architecture instances. Supported displaying the client address in slow logs.Client address Fixed the " ERR MULTI calls cannot be nested" error.

	5.0.0	Supported `unlink` and `exists` commands in Cluster Architecture instances.
Proxy 4.0	3.5.0	Supported the command analysis feature. You can view information such as QPS, P99 execution latency, average execution latency, and max execution latency of individual commands.
	3.3.0	Supported the five-second granularity for system monitoring data collection.
	3.2.0	Supported displaying the client address in slow logs. Fixed the "ERR MULTI calls can‘t be nested" error.

Note on Upgrade

The system automatically detects the minor version of the proxy. If the **Upgrade Proxy** button is grayed out, the instance proxy is already on the latest minor version.
As the version release time varies by region, the minor version release status is as displayed in the console.

Upgrade Impact

The version upgrade process mainly consists of data sync and instance switch:
During data sync, the service will not be affected.
During switch, the instances will become read-only for less than 1 minute (to wait for the completion of data sync), and a momentary disconnection (within seconds) will occur; therefore, your business should have an automatic reconnection mechanism.

Preparations for Upgrade

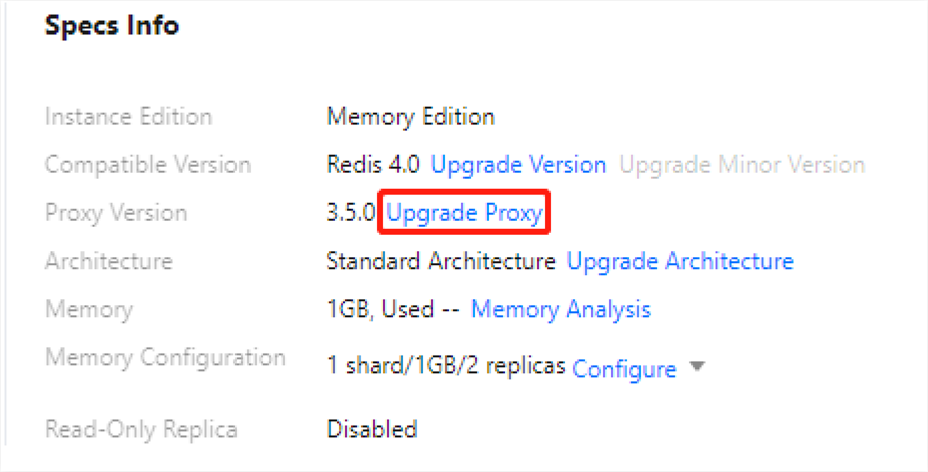
The instance to be upgraded is in **Running** status and is not executing any tasks.
We recommend that you perform upgrade in the maintenance time during off-peak hours.

Upgrade Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.

4. Click the instance ID to enter the **Instance Details** page.

5. In the **Specs Info** section on the **Instance Details** page, click **Upgrade Proxy** next to **Proxy Version**.



Specs Info

Instance Edition: Memory Edition

Compatible Version: Redis 4.0 Upgrade Version Upgrade Minor Version

Proxy Version: 3.5.0 Upgrade Proxy

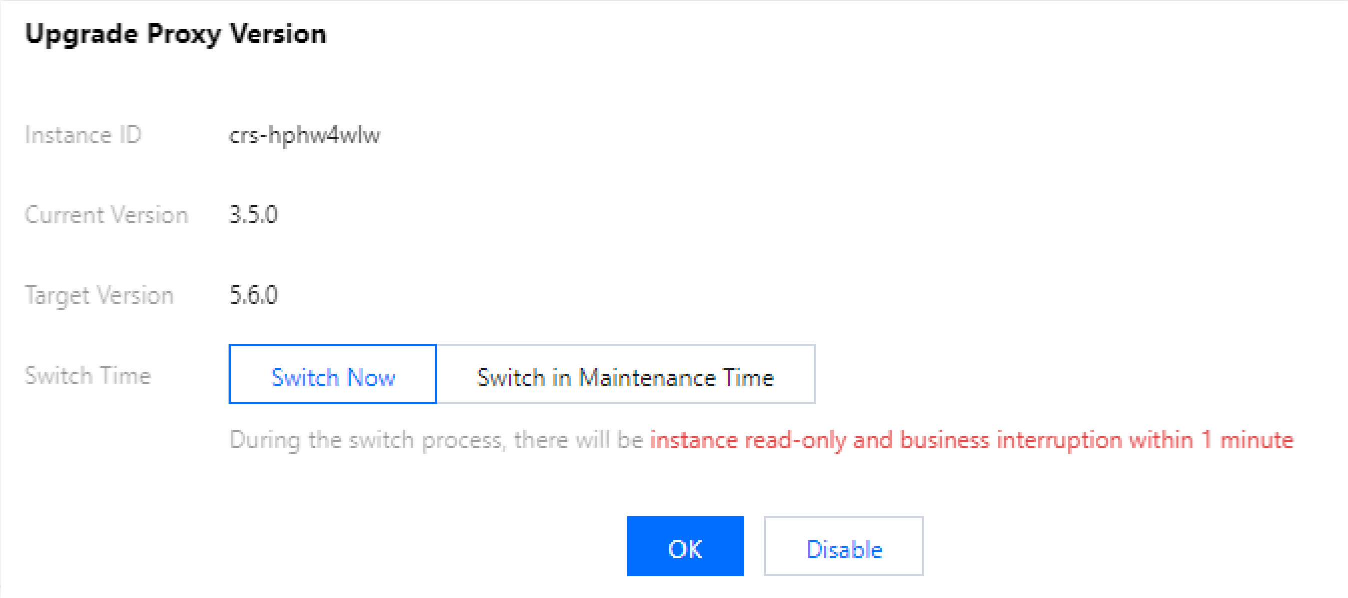
Architecture: Standard Architecture Upgrade Architecture

Memory: 1GB, Used -- Memory Analysis

Memory Configuration: 1 shard/1GB/2 replicas Configure

Read-Only Replica: Disabled

6. In the pop-up window, confirm the information of the target instance based on the following table, configure the target version, and click **OK**.



Upgrade Proxy Version

Instance ID: crs-hphw4wlw

Current Version: 3.5.0

Target Version: 5.6.0

Switch Time: Switch Now | Switch in Maintenance Time

During the switch process, there will be instance read-only and business interruption within 1 minute

Parameter	Description
Instance ID	ID of the instance to be upgraded
Current Version	Current minor version of the proxy
Target Version	Target version after proxy upgrade. The target version cannot be selected.
Switch Time	**Switch Now**: The switch will be performed when the data sync is almost completed (the data left to be synced is less than 10 MB).

	Switch in Maintenance Time: The switch will be performed during the instance maintenance time. If the switch conditions cannot be met in the current maintenance time, the switch will be attempted in the next maintenance time. You can modify the **Maintenance Window** on the instance details page.

7. Return to the instance list. After the **Instance Status** changes to **Running**, you can see that the instance version has been upgraded in the instance list or instance details.

Related APIs

API Name	Description
UpgradeProxyVersion	Upgrades proxy version

Node Management

Inspect Node Information

Last updated：2024-03-13 09:51:36

Overview

TencentDB for Redis allows you to view the instance node information, including: node ID, role, running status, and used capacity. It also supports node management, which includes: adjusting node specifications, promoting replica nodes to primary nodes, enabling read-only for replicas, and switching due to primary/secondary failures. Ops personnel can manage instance nodes efficiently and locate related anomalies during node operation through node management.

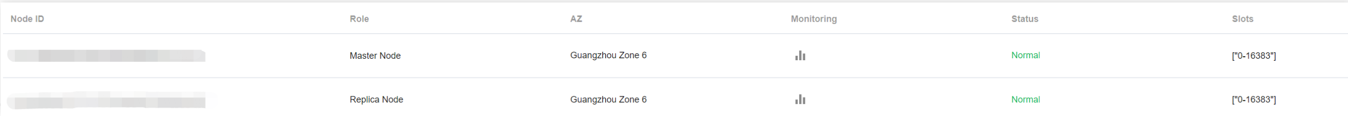
Version Description

TencentDB for Redis **4.0**, **5.0**, **6.2** versions standard architecture and cluster architecture instances, whether it's a single availability zone or multiple availability zones deployment, all **support** node management.
TencentDB for Redis **2.8** version **does not support** node management.

Viewing Node Information

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. On the **Instance List**, find the target instance whose node you need to view.
4. Click on its **Instance ID** to enter the **Instance Details** page, then click the **Node Management** tab.

Standard Architecture



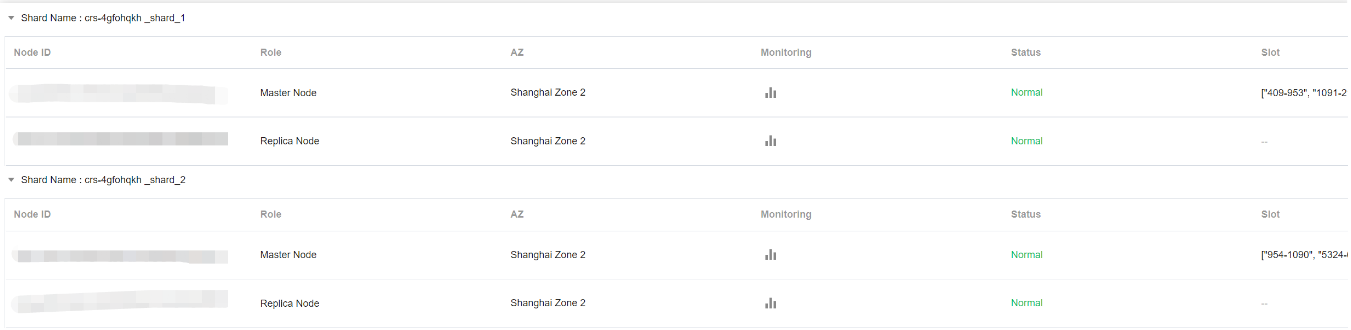
Node ID	Role	AZ	Monitoring	Status	Slots
[redacted]	Master Node	Guangzhou Zone 6	📊	Normal	["0-16383"]
[redacted]	Replica Node	Guangzhou Zone 6	📊	Normal	["0-16383"]

Parameter Name	Parameter Description
Node ID	The node ID number of the database instance.
Roles	The role of the current node, primary node or replica node.

Availability Zone	The availability zone to which the current node belongs.
Monitoring	click 📊 , view the monitoring views of various metrics of the node on the right monitoring panel. For specific information, see Monitoring at Five-Second Granularity.
Status	Running status of the current node.
Slots	Range of Slots on node.
Memory usage	Situation of memory capacity use on node.

Cluster Structure

Single Availability Zone



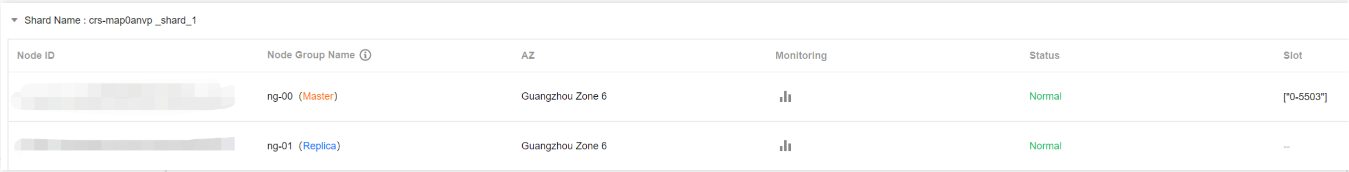
▾ Shard Name : crs-4gfohqkh _shard_1

Node ID	Role	AZ	Monitoring	Status	Slot
[redacted]	Master Node	Shanghai Zone 2	📊	Normal	["409-953", "1091-2
[redacted]	Replica Node	Shanghai Zone 2	📊	Normal	--

▾ Shard Name : crs-4gfohqkh _shard_2

Node ID	Role	AZ	Monitoring	Status	Slot
[redacted]	Master Node	Shanghai Zone 2	📊	Normal	["954-1090", "5324-
[redacted]	Replica Node	Shanghai Zone 2	📊	Normal	--

Multiple Availability Zones



▾ Shard Name : crs-map0anvp _shard_1

Node ID	Node Group Name ⓘ	AZ	Monitoring	Status	Slot
[redacted]	ng-00 (Master)	Guangzhou Zone 6	📊	Normal	["0-5503"]
[redacted]	ng-01 (Replica)	Guangzhou Zone 6	📊	Normal	--

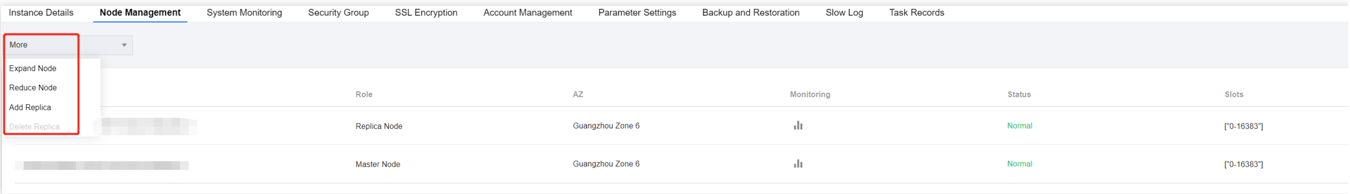
Parameter Name	Parameter Description
Shard Name	The shard name of a database sharded cluster. The shard name is concatenated from `InstanceID_shard_Shard Number` , and the shard number starts from 1 and is orchestrated in natural number order. For example: crs-mufy****_shard_1.
Node ID	The Node ID of the database instance, automatically generated by the system, serves as the unique identifier for the node.
Node Group	This parameter is displayed in cluster architectures spanning multiple availability zones.

Name	Multiple primary nodes from different shards logically form one primary node group, while replica nodes form a replica node group (replica set). Each node group name is concatenated from 'ng-number', where the number starts from 00 and is orchestrated in natural number order. Each node group name represents the primary-secondary role of the node.
Roles	This parameter represents the primary-secondary role of the node. It is displayed in cluster architecture within a single availability zone.
Availability Zone	The availability zone where the node is located. Multi-AZ instances will mark the primary availability zone or backup availability zone.
Monitoring	click 📊 , view the monitoring views of various metrics of the node on the right monitoring panel. For specific information, see Monitoring at Five-Second Granularity.
Status	Running status of the current node.
Slots	Range of Slots on node.
Memory Usage	Situation of memory usage on node.

More Actions

Configuration Adjustment

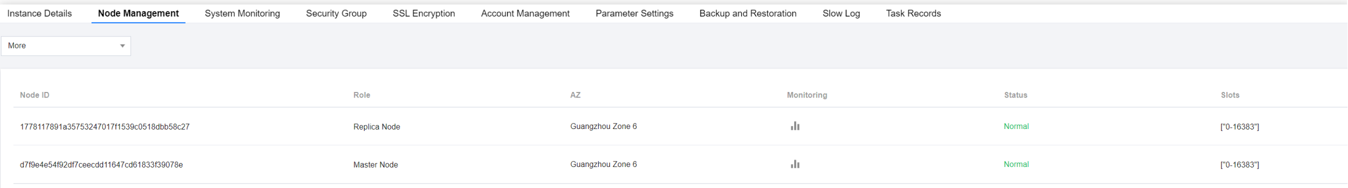
On the **Node Management** page, you can adjust the specifications of the instance node, including: expanding/reducing node capacity, adding/removing replicas, adding/removing shards (cluster configuration) and other operations. For how to configure specific parameters, see Changing Instance Specifications.



	Role	AZ	Monitoring	Status	Slots
[redacted]	Replica Node	Guangzhou Zone 6	📊	Normal	["0-16383"]
[redacted]	Master Node	Guangzhou Zone 6	📊	Normal	["0-16383"]

Promoting A Replica Node to Primary Node

For a single availability zone deployment instance, TencentDB for Redis supports manually promoting a replica node to primary node; for multi-availability zone instances, TencentDB for Redis supports promoting a replica node to primary node; the cluster configuration supports promoting a replica node group to primary node group, and promotes the original replica node or replica node group's availability zone to primary availability zone. For specific operations, see Manually Promoting to Primary Node.

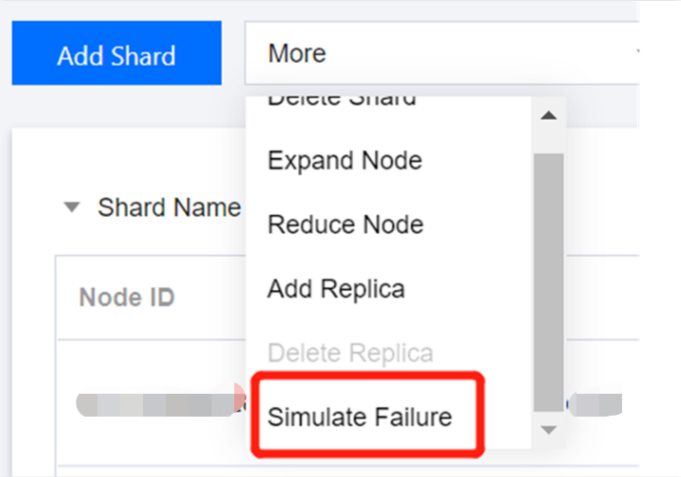


Node ID	Role	AZ	Monitoring	Status	Slots
1778117891a35753247017f1539c0518dbb58c27	Replica Node	Guangzhou Zone 6	ıl	Normal	["0-16383"]
d7f9e4e54f92df7ceecdd11647cd61833f39078e	Master Node	Guangzhou Zone 6	ıl	Normal	["0-16383"]

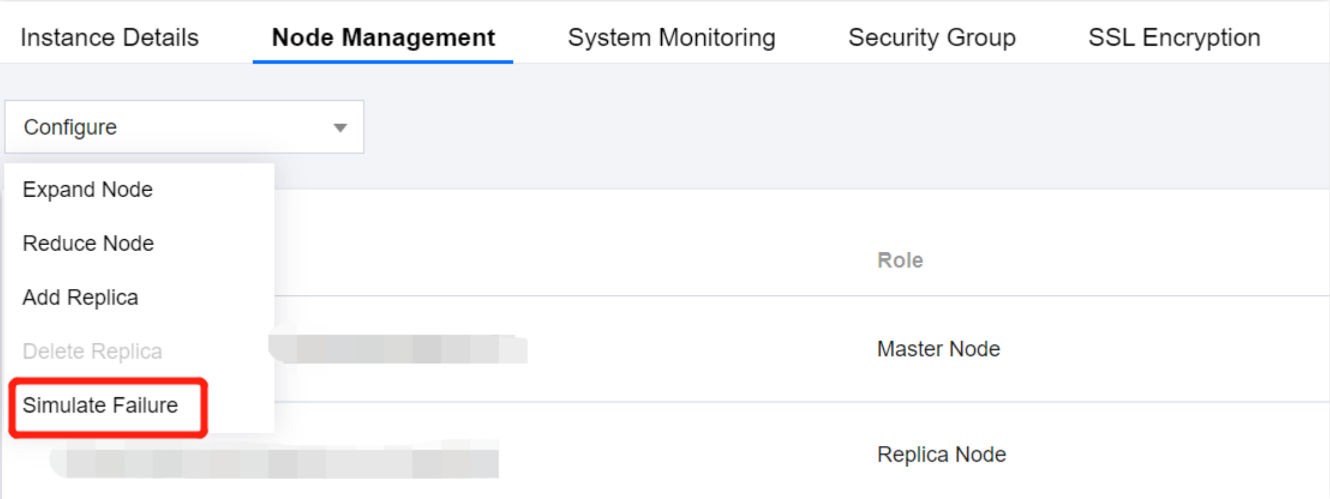
Simulate Failure

To facilitate failure simulation tests for your service, TencentDB for Redis provides a feature for simulating failures in instances deployed across multiple availability zones. You can use the failure simulation feature on the **Node Management** page. For specific operations, see Failover.

The entry for the failover simulation for **Cluster Architecture** instances:

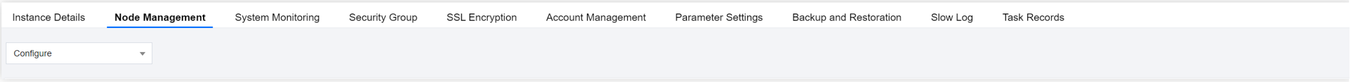


The entry for failover simulation in **Standard Architecture** instances:



Read-Only Replica

On the **Node Management** page, when the number of replica database instances is greater than or equal to 1, you can enable automatic read-write separation to improve the read performance vertically. For specific operations, see Enabling Read-Only Replica.



Related APIs

API Name	API Description
DescribeInstanceNodeInfo	Query instance node information
DescribeInstanceZoneInfo	Query Redis node detailed information

Promotion of Replicas

Last updated：2024-03-13 10:03:03

TencentDB for Redis allows you to promote a replica to the primary node/group based on your business deployment requirements. The principle of promoting a replica node to primary node (group) in TencentDB for Redis instance is to execute cluster failover command in the cluster to switch the role of the primary node to the secondary node.

Standard Architecture

In a standard architecture instance, an instance can only have one primary node. You can promote a specified node to primary node using promoting replica node to primary node. In a multi-availability zone deployed standard architecture instance, after switching the primary node, your visit may cross availability zones, leading to increased access latency and reduced QPS.

Cluster Architecture

For a cluster architecture instance with multiple replicas for a single shard, TencentDB for Redis divides the primary node and multiple replicas into different node groups for easier node management. Each node group has a node group name. For a multi-availability zone deployed cluster architecture instance, you can promote all nodes in a specified node group to a primary node group. After some of the primary nodes of the shards in the node group have switched to other node groups, you can revert all primary nodes back to the specified node group using promoting replica node to primary node. The execution process is as follows:

1. Promote all Nodes in the Node Group to Primary Nodes.
2. Mark this node group as the primary node group. After the failure recovery, the system will automatically promote the replicas in the group to primary nodes.
3. Mark this availability zone as the primary availability zone.

Notes

Service will have a brief access jitter during the promotion process. The jitter time is within a few seconds to 3 minutes. During the switching process, blocking commands such as BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE will fail once or multiple times.

During the replica promotion process, the instance will become read-only for less than one minute and a momentary disconnection will occur in your service; the execution process may fail, but retrying usually results in successful execution.

Prerequisites

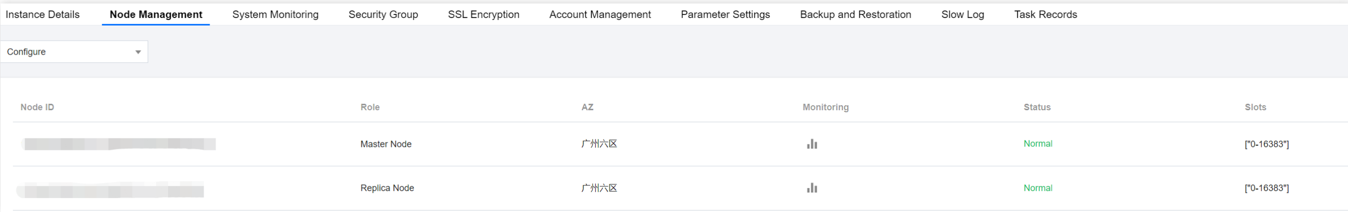
The instance has been Configuring Multi-AZ Deployment.

The database is on v4.0 or above.

Instance status is **Running**.

Directions

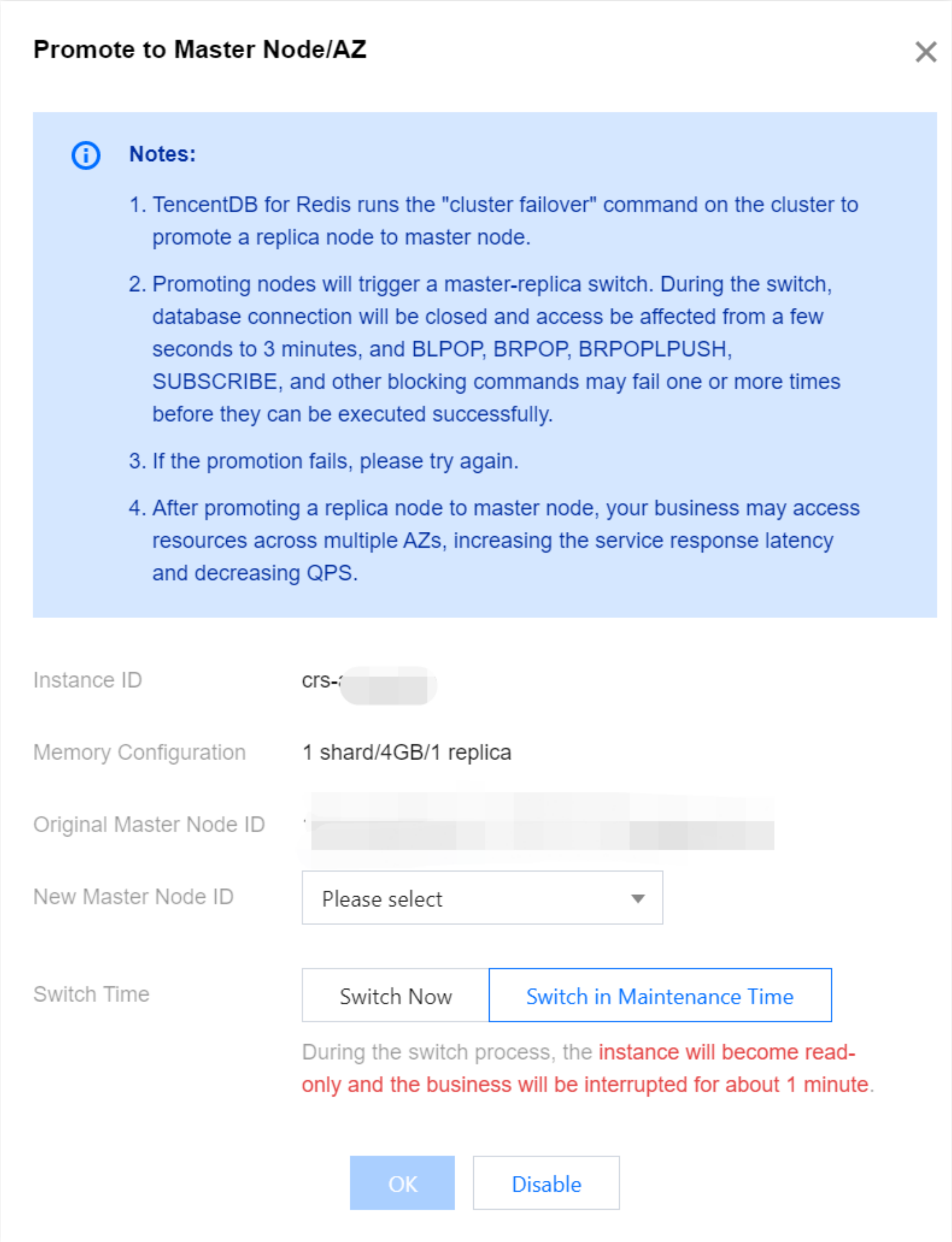
1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the **Instance List**, locate the Multi-AZ deployable instance that requires replica promotion.
4. Click the **Instance ID** to enter the **Instance Details** page.
5. On the **Instance Details** page, click the **Node Management** tab, click on the **Promote Replica to Master** at the top right of the page.



Node ID	Role	AZ	Monitoring	Status	Slots
[redacted]	Master Node	广州六区	▮	Normal	["0-16383"]
[redacted]	Replica Node	广州六区	▮	Normal	["0-16383"]

6. In the **Promote to Master Node**/**AZ** window, configure the new primary node and set the switch time, as shown in the figure below.

Standard Architecture



Promote to Master Node/AZ

Notes:

1. TencentDB for Redis runs the "cluster failover" command on the cluster to promote a replica node to master node.
2. Promoting nodes will trigger a master-replica switch. During the switch, database connection will be closed and access be affected from a few seconds to 3 minutes, and BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE, and other blocking commands may fail one or more times before they can be executed successfully.
3. If the promotion fails, please try again.
4. After promoting a replica node to master node, your business may access resources across multiple AZs, increasing the service response latency and decreasing QPS.

Instance ID: crs-[redacted]

Memory Configuration: 1 shard/4GB/1 replica

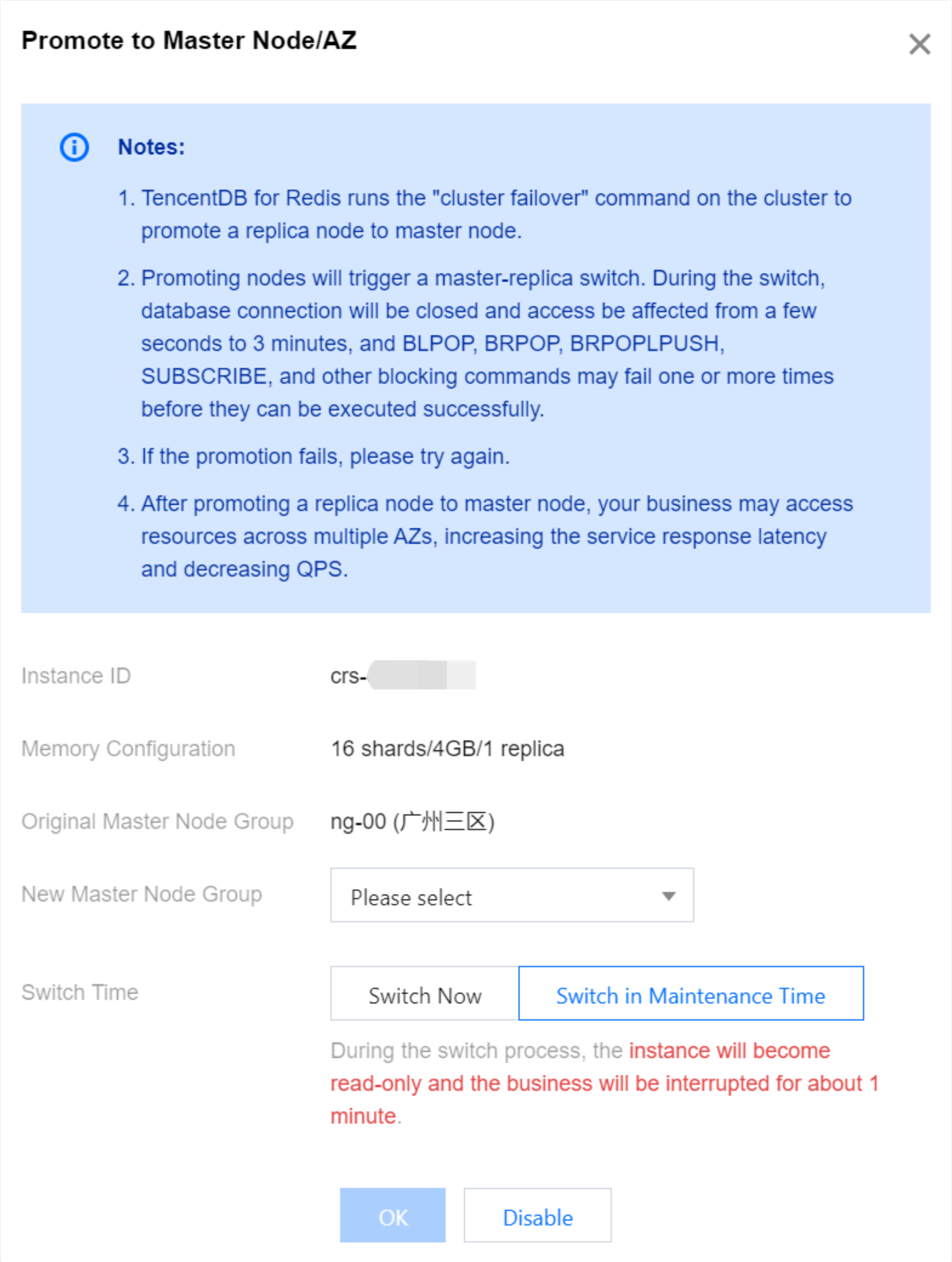
Original Master Node ID: [redacted]

New Master Node ID: Please select

Switch Time: Switch Now | Switch in Maintenance Time

During the switch process, the instance will become read-only and the business will be interrupted for about 1 minute.

Cluster architecture



Promote to Master Node/AZ

Notes:

1. TencentDB for Redis runs the "cluster failover" command on the cluster to promote a replica node to master node.
2. Promoting nodes will trigger a master-replica switch. During the switch, database connection will be closed and access be affected from a few seconds to 3 minutes, and BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE, and other blocking commands may fail one or more times before they can be executed successfully.
3. If the promotion fails, please try again.
4. After promoting a replica node to master node, your business may access resources across multiple AZs, increasing the service response latency and decreasing QPS.

Instance ID: crs-[redacted]

Memory Configuration: 16 shards/4GB/1 replica

Original Master Node Group: ng-00 (广州三区)

New Master Node Group: Please select

Switch Time: Switch Now | Switch in Maintenance Time

During the switch process, the instance will become read-only and the business will be interrupted for about 1 minute.

Interface Parameter	Parameter Meaning	Configuration Method
Instance ID	Current multi-availability zone instance ID awaiting replica promotion.	Confirm the correctness of the instance.
Memory Configuration	Current instance memory configuration specification.	Confirm the specification information of the instance.
Original Master Node ID	The standard architecture displays the current primary node's ID.	Confirm the current primary node's ID information.
New Master Node ID	The standard architecture shows this	Please select the replica node to be

	parameter, and the configuration needs to be switched to the replica node of the primary node.	promoted to primary node from the drop-down list.
Original Master Node Group	The cluster architecture shows this parameter, displaying the current name of the primary node set.	Confirm the current primary node set name.
New Master Node Group	The cluster architecture shows this parameter, and the configuration needs to be switched to the replica node set of the primary node set.	Please select the replica node set to be promoted to primary node set from the drop-down list.
Switch Time	Select the switch time period for promoting replica to primary.	**Switch Now**: The switch task will be initiated immediately. **Switch in Maintenance Time**: This indicates initiating a switch task in the maintenance window. For related introduction and configuration of the maintenance window, see Setting Maintenance Time.

7. After the configuration is complete, confirm that there are no errors, and click **OK**.

Related APIs

API Name	Description
ChangeReplicaToMaster	This API only supports the promotion of multi-AZ instance replica groups and single-AZ replicas.

Replica is read-only

Last updated：2024-03-13 10:36:47

Overview

For service scenarios with frequent reads and infrequent writes, TencentDB for Redis allows you to designate read requests for hot datasets to replica nodes. This is known as the read-only replica, where read operations and write operations are routed to different database nodes. This avoids read-write conflicts, enhances the database's concurrency processing capabilities, and improves the overall system's performance and stability. For more information, see Read/Write Separation.

Billing FAQs

The read-only replica feature is currently available for free trial.

Notes

Enabling the read/write separation feature of a read-only replica may cause data inconsistency (data on the replica node lagging behind the primary node). Please confirm first if your service allows for such inconsistency.
Disabling the read/write separation feature of a read-only replica may cause existing connections to be interrupted. It is recommended to perform this operation during off-peak business hours.

Prerequisites

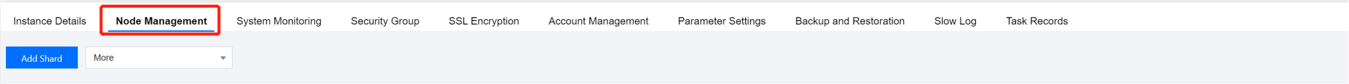
The database instance is on v4.0 or above.
The database instance is in **Running** status.

Directions

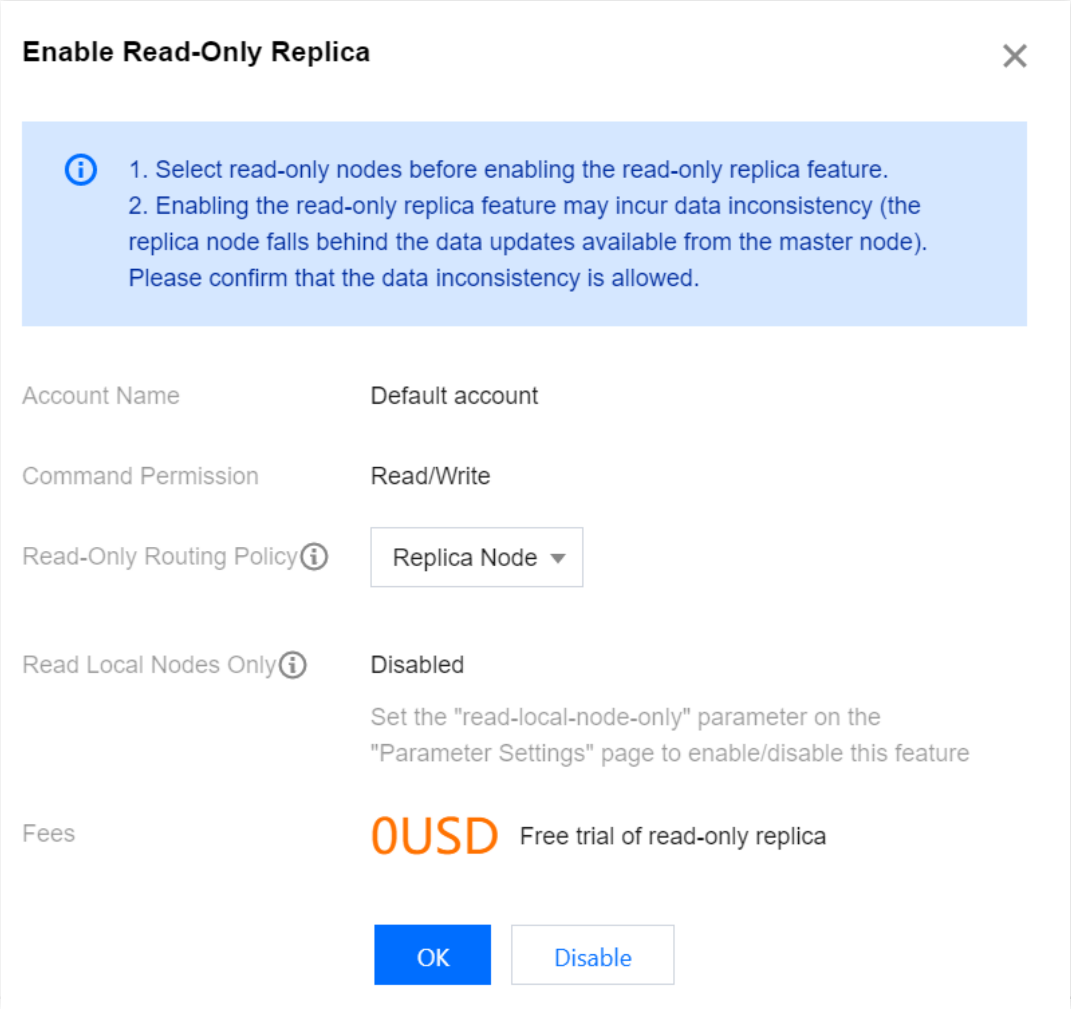
Enabling Read-Only Replica

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the **Instance List**, find the target instance.

4. Click the **Instance ID** to enter the **Instance Details** page, click the **Node Management** tab.
5. In the upper right corner of the **Node Management** page, click the icon next to the **Read-Only Replica**.



6. In the pop-up dialog box, configure the **Read-Only Replica** node. For specific parameter information, see the table below.



Enable Read-Only Replica

1. Select read-only nodes before enabling the read-only replica feature.
2. Enabling the read-only replica feature may incur data inconsistency (the replica node falls behind the data updates available from the master node). Please confirm that the data inconsistency is allowed.

Account Name: Default account

Command Permission: Read/Write

Read-Only Routing Policy: Replica Node

Read Local Nodes Only: Disabled

Set the "read-local-node-only" parameter on the "Parameter Settings" page to enable/disable this feature

Fees: 0USD Free trial of read-only replica

Parameter Name	Parameter Description
Account Name	Set as **Default account**, that is, the system only enables read-only replicas for the default account.
Command Permission	Set as **Read/Write** permissions. The default account has read and write permissions.

Read-Only Routing Policy	By default, it is set as **Replica Node**, you can also select **Master Node**, or select both Replica Node and Primary Node. Read requests will be automatically load-balanced to the configured read-only nodes.
Read-Only Local Node	This parameter is displayed when the instance is deployed in multiple availability zones. Enable/disable reading local nodes only feature is fixed as **Disabled**. You can enable/disable this feature by configuring the reading-local-node-only parameter on the **Parameter Configuration** page in the console.
Fees	Currently available for free trial.

7. Confirm the parameter configuration is correct, then click **OK**.
8. When the **Instance Status** changes to **Processing**, wait for it to be **Running**. Under **Instance Details** page in the **Specification Information** zone, you can see that **Read-Only Replica** is marked as **Enabled**, indicating that read/write separation is now available.

Disabling Read-Only Replica

Note:

If the read-only routing policy of the instance's custom account specifies that read requests be distributed to the replica node, you cannot directly disable the read-only replica feature. Please prioritize setting the read-only policy to primary node, then disable the read-only replica feature.

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the **Instance List**, find the target instance.
4. Click the **Instance ID** to enter the **Instance Details** page, click the **Node Management** tab.
5. In the upper right corner of the **Node Management** page, click the icon next to the **Read-Only Replica**.
6. In the **Disable Read-Only Replica** dialogue box, understand the impact of disabling read-only replica, confirm the disable, click **OK**.
7. **Instance Status** has been changed to **In progress**. Wait for it to be **Running**. In the **Specification Information** zone of the **Instance Details** page, you can see **Read-Only Replica** is marked as **Not Enabled**, indicating that the disabling process is complete.

Command

Once the Read-Only Replica is enabled, some commands will not be routed to the replica node, they will only be routed to the primary node. The specific commands are shown in the following table.

Architecture Category	Commands only routed to the primary node when Read/Write Separation is enabled.
Standard Edition	getdel,getex,set,setnx,setex,psetex,append,del,unlink,setbit,bitfield,setrange,incr,decr,rpush,lpus
Cluster Edition	getdel,getex,set,setnx,setex,psetex,append,del,unlink,setbit,bitfield,setrange,incr,decr,rpush,lpus

Related APIs

API Name	Description
EnableReplicaReadonly	Enabling Read/Write Separation
DisableReplicaReadonly	Disabling Read/Write Separation

Multi-AZ Deployment Management

Configuring Multi-AZ Deployment

Last updated：2024-03-13 11:25:54

This document describes how to configure multiple AZs for your TencentDB for Redis instance and how to view them in the console.

Overview

You can now deploy TencentDB for Redis master and replica nodes in different AZs of the same region as instructed in Multi-AZ Deployment. Multi-AZ deployed instances have higher availability and better disaster recovery capability than single-AZ deployed instances.

Read/Write separation disabled (that is, replicas can be written to and read from): Write/read requests in a replica AZ are routed by proxy to the master node, and the master node synchronizes with replica nodes to ensure consistent data across all nodes. In this process, only one cross-AZ access happens.

Read/Write separation enabled (that is, replicas can only be read from): Write requests are routed by proxy to the master node, but read requests are routed to the replica node in the same AZ as the proxy, so that read requests can get responded by the nearest node.

We recommend that you deploy one master node and one replica node in the master AZ, and another replica node in the replica AZ. If the master node fails, the replica node in the master AZ can be promoted quickly to avoid cross-AZ master-replica switch. Such a deployment solution can maximize service availability and reduce the delay caused by master node failures.

Note:

Currently, multi-AZ deployment is supported only for TencentDB for Redis 4.0, 5.0, and 6.2 standard architecture and cluster architecture.

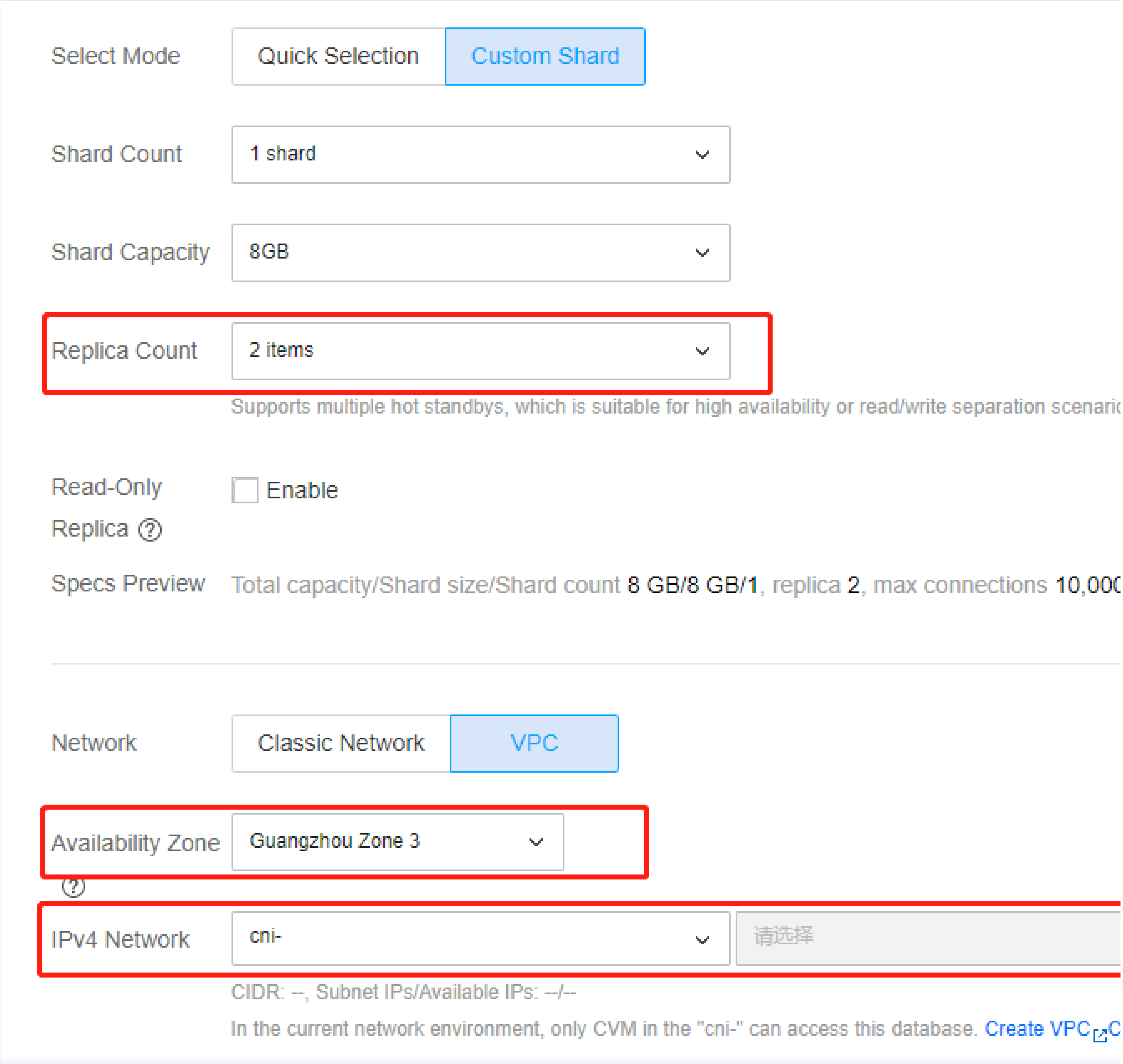
Configuring multi-AZ deployment

1. Log in to the TencentDB for Redis purchase page.
2. Set parameters such as billing mode, region, instance edition, compatible version, architecture, and memory. For more information, see Creating TencentDB for Redis Instance.
3. The **replica quantity** determines the maximum number of AZs, which is the replica quantity plus 1. Select a value based on your actual business needs.

Note:

Deploy Redis nodes in different AZs. Up to six AZs are supported.

4. When configuring **AZ**, select **Multi-AZ Deployment** and set the AZs of the master and replica nodes.



Select Mode: Quick Selection | Custom Shard

Shard Count: 1 shard

Shard Capacity: 8GB

Replica Count: 2 items

Supports multiple hot standbys, which is suitable for high availability or read/write separation scenario

Read-Only Replica: □ Enable

Specs Preview: Total capacity/Shard size/Shard count 8 GB/8 GB/1, replica 2, max connections 10,000

Network: Classic Network | VPC

Availability Zone: Guangzhou Zone 3

IPv4 Network: cni- 请选择

CIDR: --, Subnet IPs/Available IPs: --/--

In the current network environment, only CVM in the "cni-" can access this database. Create VPC C

5. Set information such as network, port, parameter template, instance name, and password.

6. Confirm the price. After making the payment, return to the instance list. After the status of the instance changes to **Running**, it can be used normally.

Log in to the TencentDB for Redis console. Instances with the M flag in the **AZ** column are multi-AZ deployed. You can view their AZ information.

In the instance list, click the instance ID to enter the management page. On the **Instance Details** tab, instances with the

M flag in the **AZ** section are multi-AZ deployed. You can view their AZ information.
In the instance list, click the instance ID to enter the management page. On the **Node Management** tab, you can view the details of nodes in different AZs.

Related APIs

API	Description
CreateInstances	Creates an instance (the `NodeSet.N` parameter defines whether the instance is multi-AZ deployed).

Change the Availability Zone

Last updated：2023-11-06 10:21:09

Overview

A multi-AZ deployment is created by combining multiple AZs in the same region. A multi-AZ deployed cluster has better disaster recovery capabilities than a single-AZ deployed cluster and can protect your database instances from IDC-level failure or AZ outages. An existing single-AZ deployed cluster can be automatically upgraded to a multi-AZ one through online data migration without business interruptions.

Billing overview

The multi-AZ feature is currently available free of charge; that is, a single-AZ deployed cluster can be upgraded to a multi-AZ one for free.

Use limits

This feature is currently unavailable in Jakarta and Mumbai regions and will be available in more regions and AZs in the future.

Notes

If Reading Local Nodes Only is not required, upgrading to multi-AZ deployment will involve metadata migration only without affecting the service, which generally take less than three minutes to complete.
If Reading Local Nodes Only is required, you will need to upgrade the proxy version and Redis kernel minor version, which will involve data migration and may take hours to complete. There will be one or several momentary disconnections within three minutes after the upgrade is completed, so make sure that your business has an automatic reconnection mechanism.

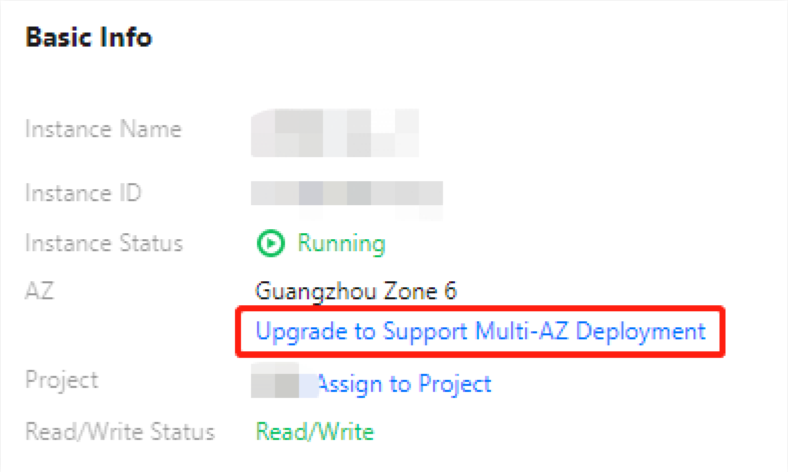
Prerequisites

The cluster region has at least two AZs.
The database is on v4.0 or later.

The instance is in **Running** status.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the **instance ID** of the target instance in blue to enter the **Instance Details** page.
5. In the **Basic Info** section on the **Instance Details** page, click **Upgrade to Support Multi-AZ Deployment** after **AZ**.



Basic Info

Instance Name [redacted]

Instance ID [redacted]

Instance Status Running

AZ Guangzhou Zone 6

Upgrade to Support Multi-AZ Deployment

Project [redacted] Assign to Project

Read/Write Status Read/Write

6. In the **Upgrade to Support Multi-AZ Deployment** window, learn more about the impact of upgrading to multi-AZ deployment, confirm whether to support Reading Local Nodes Only, and click **OK**.
7. The **Instance Status** in **Basic Info** becomes **Upgrading to support multi-AZ deployment**. Wait for the status to become **Running**.

Notes

After the upgrade, you need to manually change the AZs of replica nodes, so that they are in the AZs different from the master node. For detailed directions, see Adding Replicas to Multi-AZ Deployed Instance.

Related APIs

API	Description
UpgradeVersionToMultiAvailabilityZones	Upgrades an instance to support multi-AZ deployment

Adding Replicas to Multi-AZ Deployed Instance

Last updated：2024-03-13 11:27:07

Overview

A multi-AZ deployed instance uses the one-replica architecture by default. One master node and two replica nodes are recommended, with the master node and one replica node deployed in the master AZ and the other replica node deployed in the replica AZ. This deployment mode maximizes the service availability and greatly reduces the delay caused by failures of the master node. If the master node fails, the replica node in the master AZ can be first elected as the new master node, ensuring that the access latency in the master AZ will not be affected by the switch of the master node to the replica AZ otherwise. For more information, see Multi-AZ Deployment. We recommend you select the number of replicas based on your business needs.

Billing Description

Pay-as-you-go: The instance will be billed based on the new specification on the next hour under tier 1, and fees will be settled on each hour. For pricing details, see Pricing.

Notes

If the number of replicas is increased, the instance will be charged at the price of the new specification.
Increasing the number of replicas will not cause momentary disconnections or command execution failures or affect existing connections.

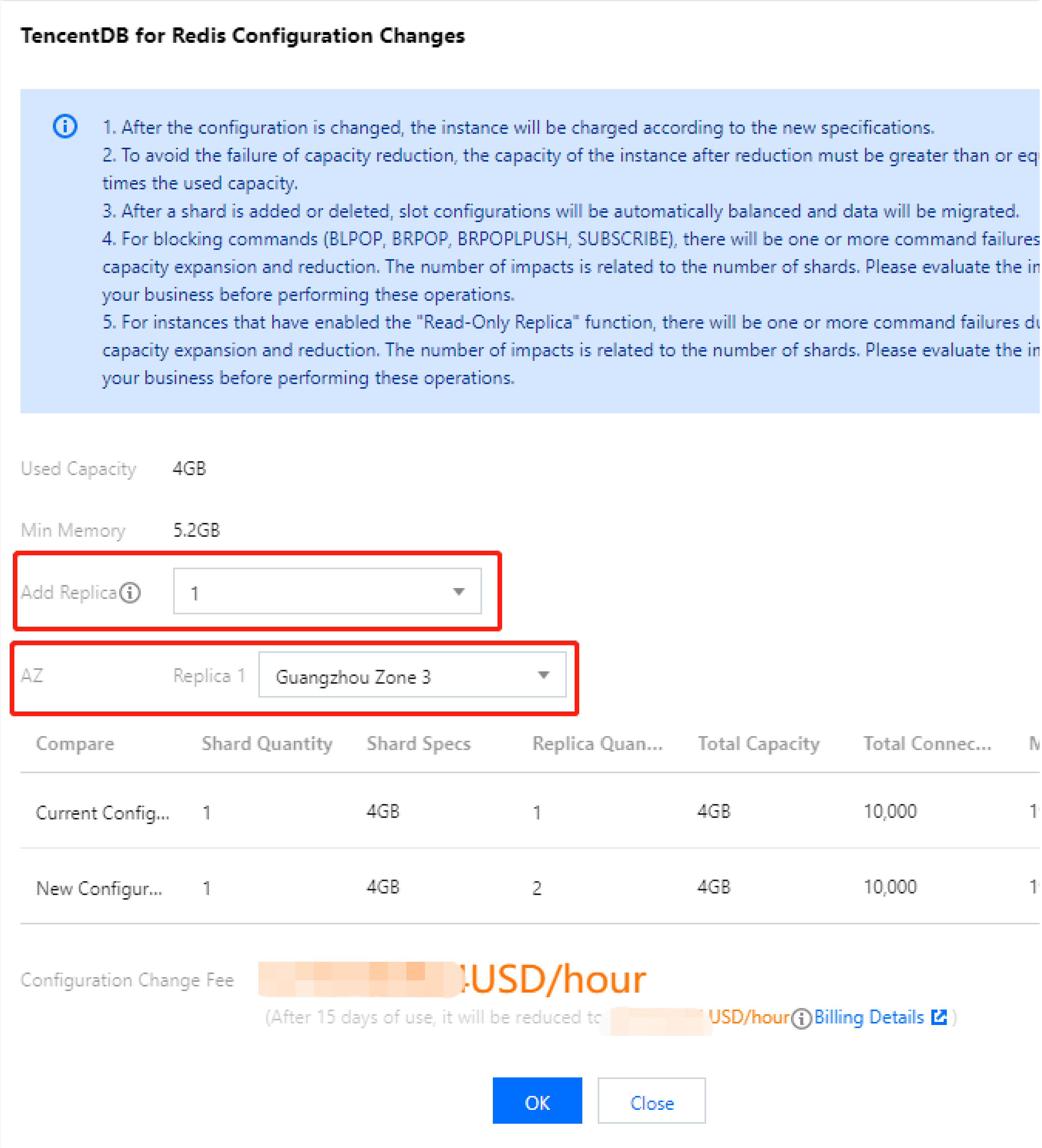
Prerequisites

The instance has been configured with multi-AZ deployment.
The database is on v4.0 or later.
The instance is in **Running** status.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the **instance list** on the right, select the region.
3. In the instance list, find the target instance.
4. In the **Operation** column of the target instance, select **Configure** > **Add Replica**.
5. In the **TencentDB for Redis Configuration Changes** pop-up window, select the number of replicas to be added from the **Add Replica** drop-down list. Then, select AZs for them in the **AZ** drop-down list. The parameters are as described below.

Parameter	Description
Used Capacity	Capacity used by the current instance.
Min Memory	Minimum memory specification of the current instance required to prevent the disk space from being used up.
Add Replica	Specify the number of replicas to be added based on the required security level of your business data.
AZ	Specify AZs for the added replicas. For more information on how to configure this parameter, see Multi-AZ Deployment.
Compare	Comparison of the specification before and after adding replicas. Check whether the new configuration meets your expectations.
Configuration Change Fee	Make sure that you understand the billing information of the new configuration. The hourly rate of the new configuration will be displayed for pay-as-you-go instances.



TencentDB for Redis Configuration Changes

1. After the configuration is changed, the instance will be charged according to the new specifications.
2. To avoid the failure of capacity reduction, the capacity of the instance after reduction must be greater than or eq
times the used capacity.
3. After a shard is added or deleted, slot configurations will be automatically balanced and data will be migrated.
4. For blocking commands (BLPOP, BRPOP, BRPOPLPUSH, SUBSCRIBE), there will be one or more command failures
capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the in
your business before performing these operations.
5. For instances that have enabled the "Read-Only Replica" function, there will be one or more command failures du
capacity expansion and reduction. The number of impacts is related to the number of shards. Please evaluate the in
your business before performing these operations.

Used Capacity 4GB

Min Memory 5.2GB

Add Replica 1

AZ Replica 1 Guangzhou Zone 3

Compare	Shard Quantity	Shard Specs	Replica Quan...	Total Capacity	Total Connec...	N
Current Config...	1	4GB	1	4GB	10,000	1
New Configur...	1	4GB	2	4GB	10,000	1

Configuration Change Fee [redacted]USD/hour
(After 15 days of use, it will be reduced tc [redacted] USD/hour Billing Details)

6. After completing the configuration, click **OK**. Then, the **instance status** will change to **Processing**. Wait for the status to become **Running**.

Related APIs

API	Description

UpgradeInstance	Upgrades instance specification, including the shard size, shard quantity, and replica quantity.

Accessing Multi-AZ Deployed Instances

Last updated：2023-03-21 16:19:46

Both single-AZ and multi-AZ deployed TencentDB for Redis instances can be easily accessed at a private IPv4 address.

Private IPv4 address

For a TencentDB for Redis instance, the master and replica nodes are deployed in different AZs. Subnets in different AZs can be created in the same VPC, which can communicate with each other over the private network by default. The system assigns a private IPv4 address according to the VPC selected for the instance, at which all clients in the entire region can access the instance.

The private IPv4 address serves for the failover of the Redis service. After a Redis service node fails and triggers the master-replica switch, the Redis service will automatically update the backend service processes associated with the private IPv4 address, so the business does not need to change this address.

Viewing the private IP of an instance

Log in to the TencentDB for Redis console and view the private IPv4 address assigned to the instance in the **Network** column in the instance list.

Go to the instance list, click the instance ID to enter the **Instance Details** page, and view the **private IPv4 address** of the instance in the **Network Info** section.

Accessing a multi-AZ deployed instance

You can access a TencentDB for Redis instance by using an SDK as instructed in Connecting to TencentDB for Redis Instance.

To reduce the access latency of a multi-AZ deployed instance, TencentDB for Redis allows you to read local nodes only.

Failure Simulation

Last updated：2022-11-08 17:18:57

Overview

TencentDB for Redis supports the automatic failover of proxy nodes and Redis servers (data storage nodes) to ensure service availability.
You can use the failure simulation feature in the TencentDB for Redis console to perform failure simulations or tests. The `shutdown` command is sent to all master nodes to trigger the automatic high availability (HA) logic to perform failure simulation.

Proxy failover

Proxy nodes are used in both standard and cluster architectures of TencentDB for Redis. The standard architecture has three proxy nodes, while the number of proxy nodes in the cluster architecture increases linearly with that of shards. The high availability design of proxy nodes is as follows:
Multiple proxy nodes support the high availability and load balancing of the proxy service.
Proxy nodes are deployed on three physical devices to ensure high availability.
If a proxy node fails, the testing system will detect the failed node and automatically add new one.

Redis server failover

TencentDB for Redis in standard architecture or cluster architecture adopts the same cluster management mechanism as the Redis Cluster, which uses the Gossip protocol to detect the status of nodes in a cluster. The `cluster-node-timeout` parameter is used to specify the maximum amount of time a Redis cluster node can be unavailable, without it being considered as failing. We recommend you set this parameter to its default value (1500 ms) and do not change it. For more information, see Scaling with Redis Cluster.

Notes

Only instances in the "Running" status can perform failure simulations.
-Only instances deployed in multi-AZ can perform failure simulations.

Use Limits

A failure simulation will make Redis unavailable for less than one minute until the failover is completed. The data written during the simulation may be lost.

The instance service downtime caused by failure simulations won't be counted into SLA

Prerequisites

The instance has been deployed in multi-AZ. For more information, see Configuring Multi-AZ Deployment.

The database is on v4.0 or later.

The instance is in the **Running** status.

Directions

1. Log in to the Redis Console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. On the instance management page, click **Node Management** tab, and select **Simulate Failure** in **More** drop-down list.
5. In the **Simulate Failure** pop-up window, check the instance name and ID, learn about how failure simulation works, and click **OK**. The instance status will change to **Processing**.
6. In the left sidebar, click **Task Management**, and wait for the task to be completed. The failure simulation is successfully performed when the instance status becomes **Running**.

##API

API	Description
KillMasterGroup	Performs a failure simulation
SwitchProxy	Simulates the failure of a proxy node

Auto-Failback

Last updated：2023-03-14 15:39:53

TencentDB for Redis provides the automatic failback feature for instances deployed across AZs. After the feature is enabled, if the master node is switched from the master AZ or master node group (cluster architecture) to another AZ or group after a failover occurs, it will be automatically switched back, simplifying subsequent Ops operations. You can enable or disable this feature by setting a database parameter.

Master AZ

You can deploy nodes of a TencentDB for Redis instance in master and replica AZs. The automatic failback is enabled by default. If the master node is switched from the master AZ or master node group (cluster architecture) to another AZ or group after a failover occurs, it will be automatically switched back after all of the failed nodes are replaced.

You can specify the master AZ when creating an instance. Or, you can manually promote a replica AZ to master after the instance is created (after a replica node/node group is promoted to master, its AZ will become the master AZ).

Enabling/Disabling automatic failback

You can manage the automatic failback feature with a database parameter. The feature is enabled by default. Enabling/disabling it does not affect your access to Redis.

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. On the **Instance Details** page, select the **Parameter Settings** tab.
6. On the **Parameter Settings** tab, click the **Modifiable Parameters** tab, find `auto-failback` in the parameter list, and set it to `yes` .

Reading Local Nodes Only

Last updated：2023-04-03 16:54:59

To reduce the access latency of a multi-AZ deployed instance, TencentDB for Redis allows you to read local nodes only. The principle is as follows:

Enable the "read-only replica" feature. Before you enable the feature, confirm that data delay of replicas is allowed.

Enable the "read local nodes only" feature by setting a database parameter.

If there is an available proxy node in the same AZ as the load balancing cluster, the cluster can perceive and access it only.

The proxy node can access the AZ information stored in Redis nodes and route read requests to a Redis node in the same AZ.

Enabling the "read local nodes only" feature

The "read local nodes only" feature is disabled by default. You can enable/disable the feature for an existing instance by setting a database parameter on the **Parameter Settings** page in the console. Or, you can create a parameter template in which the parameter is specified and apply the template when creating an instance.

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. On the **Instance Details** page, select the **Parameter Settings** tab.
6. On the **Parameter Settings** tab, click the **Modifiable Parameters** tab, find `read-local-node-only` in the parameter list, and set it to `yes` .

Reading local nodes only and read-only routing policy

After enabling **read-only replica** in TencentDB for Redis as instructed in Enabling/Disabling Read/Write Separation, you can specify the **read-only routing policy** as described in Managing Account to control the distribution of account read requests to the master or replica nodes.

After the **read local nodes only** feature is enabled, **it has a higher priority than the read-only routing policy**. In this case, nodes in the same AZ will be read first, and then the read-only routing policy will be executed.

Read-only replica is enabled for the instance, and **Read-only Routing Policy** is set to **Replica Node**.

The "read local nodes only" feature is enabled (`read-local-node-only` = `yes`).

Only one master node exists in the master AZ.

In this case, when the business in the master AZ accesses the proxy node in the master AZ, the proxy will ignore the read-only routing policy and route read requests to the master node in the master AZ to avoid reading across AZs.

Related APIs

API	Description
UpgradeVersionToMultiAvailabilityZones	Upgrades an instance to support multi-AZ deployment

Backup and Restoration

Overview

Last updated：2023-05-23 10:26:53

Overview

TencentDB for Redis supports data backup and restoration. The backend service can automatically back up the instance data on a regular basis in the console. Manual backup can also be performed at any time. The backup data is saved in RDB format corresponding to the Redis Engine Edition and stored in COS for high data reliability. TencentDB for Redis supports instance data restoration through clone.

Automatic Backup

By default, a full data backup will be performed every day between 02:00 AM and 08:00 AM. In order to avoid increasing the processing burden on the master database, the backup data comes from replica nodes. You can flexibly configure the start time period of automatic backup tasks based on your conditions. You can see the daily backup data in **Backup and Restoration** in the TencentDB for Redis console. For more information, see Backing up Data.

Manual Backup

In addition to automatic backup that is performed by the system backend regularly, you can manually back up your instance in the TencentDB for Redis console to meet your diversified needs. For more information, see Backing up Data。

Backup File

The backup list will display all the backup files of the instance and backup information. TecentDB for Redis provides two backup download addresses: private network and public network, and you can choose the one that best suits your needs.

Download from public network: You can download backup data anywhere you can access the Internet. The public network download is enabled by default, and you can disable it as needed.

Download from private network: In order to prevent database backup data from being dragged, TencentDB for Redis allows you to download backup files from specified servers on the allowed private network. You can’t download backup files across regions, and you can do so only in the region where the Redis instance resides.

Data Restoration

Only TencentDB for Redis 2.8 supports fast data restoration based on backup files.

Instance Clone

TencentDB for Redis Memory Edition (excluding v2.8) supports instance clone, i.e., creating a complete instance based on a backup file. The data of the instance is the same as that in the backup file. You can use the clone feature to analyze previous data. You can also roll back an instance by swapping the IPs of the new instance and the original instance.

Backing up Data

Last updated：2023-04-19 14:31:26

Overview

The backend service of TencentDB for Redis backs up the data in an instance periodically. By default, it performs a full data backup in any hour between 02:00 AM and 08:00 AM every day to generate a database backup file in RDB format and store it persistently in COS. In addition to periodic backups on the system backend, you can also perform manual backup in the TencentDB for Redis console to back up all data at any time outside the automatic backup time period. This helps meet your backup requirements in different scenarios.

Note

During data backup, in order not to increase the processing burden on the master database or affect the business, the backup source will be a replica database in the instance.

Configuring the automatic backup time

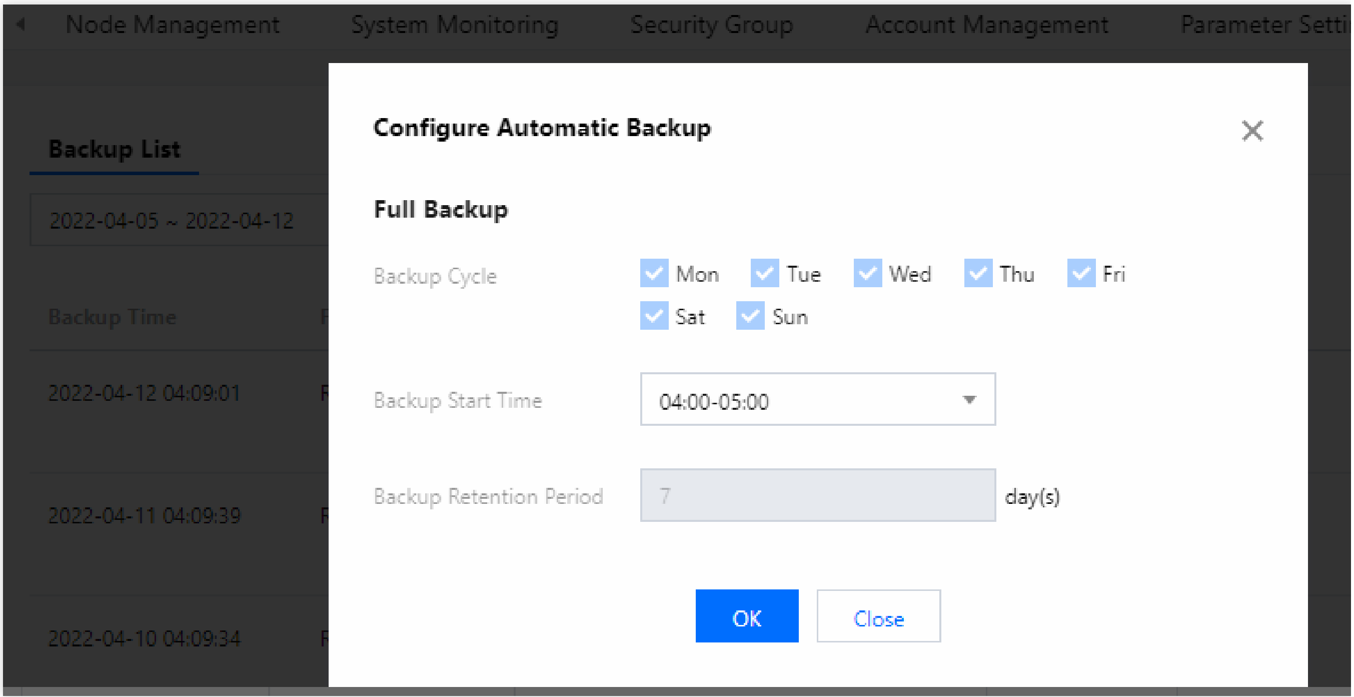
1. Log in to the TencentDB for Redis console.
2. In the instance list, click the instance ID to enter the **Instance Details** page.
3. On the **Instance Details** page, select the **Backup and Restoration** tab.
4. In the top-right corner of the backup list, click **Configure Automatic Backup**.
5. In the **Configure Automatic Backup** window, configure the following parameters and click **OK**:

Auto Backup: It is enabled by default. To disable it, submit a ticket.

Backup Cycle: All options are selected by default and cannot be modified.

Backup Start Time: Select an option in the drop-down list.

Backup Retention Period: It is 7 days by default. To change it, submit a ticket.



Configure Automatic Backup

Full Backup

Backup Cycle ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri ☑ Sat ☑ Sun

Backup Start Time 04:00-05:00

Backup Retention Period 7 day(s)

6. The backup task starts within the specified time period everyday. After backup is completed, you can view backup files in the following two ways:

In the **Backup List** on the **Backup and Restoration** tab, view the backup files with **Backup Mode** being **Auto Backup**.

On the left sidebar, select **Database Backup**. Then, view the backup files with **Backup Mode** being **Auto Backup** on the **Backup List** tab.

Note

After automatic backup is configured, the system backend service will perform a full backup once a day within the configured backup time period.

Backup start may be delayed if affected by relevant processes.

Manual backup

1. Log in to the TencentDB for Redis console.
2. In the instance list, click the instance ID to enter the **Instance Details** page.
3. In the top-right corner of the **Instance Details** page, click **Manual Backup**.
4. In the **Manual Backup** window, enter the **Remarks** for the backup task.
5. Click **OK**.

Note:

The backup task will start after around one minute.

During backup, you can select **Task Management** on the left sidebar to view the task progress and details.

After backup is completed, you can select **Database Backup** on the left sidebar and view the backup files with **Backup Mode** being **Manual Backup** on the **Backup List** tab. You can also view such files in the **Backup List** on the **Backup and Restoration** tab on the **Instance Details** page.

Related APIs

API	Description
ManualBackupInstance	Backs up an instance
DescribeInstanceBackups	Queries the list of backups of an instance

Downloading Backup Files

Last updated：2023-05-23 10:29:48

Overview

After the database loses some or all of the data due to some reasons, the backup file can help you retrieve the lost data.

Prerequisites

You have configured the backup file download method in the current region.

You have completed an automatic or manual backup task. For directions, see Backing up Data.

Directions

1. Log in to the TencentDB for Redis console.
2. Enter the **Backup List** page by any of the following methods.

On the top of the instance list, select a region, and find the target instance in the instance list. Then, click the instance ID to enter the **Instance Details** page, and select the **Backup and Restoration** tab to enter the backup list.

On the left sidebar, select **Database Backup** to enter the backup list.

3. In the backup list, find the backup file to be downloaded, and click **Download** in the **Operation** column.

(Optional) If you select **Database Backup** in the left sidebar to enter the backup list, before downloading the backup file, click **Details** in the **Operation** column to view the backup details. After confirming the backup file information is correct, click **OK**.

4. In the **Download Backup** pop-up window, copy the download address or click **Download** to download the backup file. Cross-AZ download is not supported.

Public network download is disabled

a. In **Download Conditions**, the default values are the VPC ID and IP address configured in the current region, which can also be changed based on your needs.

b. If the instance is on a standard architecture, click **Generate Download Address**, and click **Copy** in the **Download Address** area.

If the instance is on a cluster architecture, click **Generate Download Address**. In the **Download Address** area, the download address of each backup will be displayed in a list, click **Copy download URL** to get the download address of backup data for each shard. Click **Export Download Address** to get the shell script that contains the download URLs of all backup shards.

Note

To change **Download Conditions**, you need to click **Generate Download Address** to regenerate **Download Address**.

The download address is valid for 6 hours. Get a new one after it expires.

Downloaded backup shard lists the shard node information of each shard, which is in the format of `instance ID-node-shard ID`. The shard IDs are sequenced from 0.

c. Download the backup file over the private nework.

Standard architecture: Run the command `wget -c 'private network address' -O backup.tar` in a CVM instance to download the backup at a high speed over the private network. For detailed directions on how to log in to CVM, see Customizing Linux CVM Configurations.

Cluster architecture: Run the .sh script on the CVM to download the backup file.

Public network download is enabled

a. Click **Copy** next to the **Download Address**.

b. Copy the download address and open it in a browser to download the backup file.

Cloning Data

Last updated：2023-04-19 14:31:45

Overview

TencentDB for Redis Memory Edition master and replica instances allow you to clone a complete instance from a backup file in both single-AZ and multi-AZ deployments. The clone instance has the same data as the backup file. You can use the clone feature to analyze historical data. You can also roll back an instance by swapping the IPs of the clone and original instances.

Version description

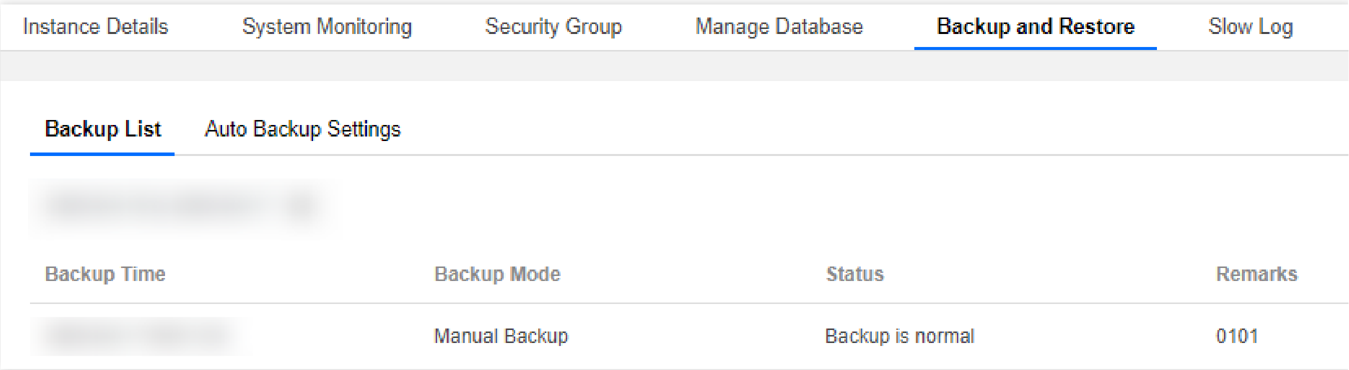
Currently, all Redis versions except 2.8 support instance cloning.

Prerequisites

You have backed up the data and generated a backup file. For more information, see Backing up Data.
The instance is in **Running** status with no ongoing tasks.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. On the **Instance Details** page, select the **Backup and Restoration** tab.
5. In **Backup List**, select the time point to roll back the clone instance to based on the backup time. Then, click **Clone Instance** in the **Operation** column.



[redacted]

Backup Time	Backup Mode	Status	Remarks
[redacted]	Manual Backup	Backup is normal	0101

6. On the **Clone TencentDB for Redis Instance** page, configure the clone instance as needed based on the parameter descriptions below:

Parameter	Required	Description
Billing Mode	Yes	The billing mode of the clone instance is independent of the source instance, so you can flexibly select **Monthly Subscription** or **Pay as You Go**. For billing details, see Billing Overview.
Region	Yes	The region of the clone instance is the same as that of the source instance and cannot be modified.
Instance Edition	Yes	The edition and architecture of the clone instance are the same as those of the source instance and cannot be modified.
Clone Backup	Yes	Confirm the source instance ID and name. Confirm the backup time. The clone instance will restore data based on the backup file at this time point.
Specs Preview	Yes	The specifications of the clone instance are the same as those of the source instance. You can preview the maximum number of connections and network throughput that the clone instance can support.
Network	Yes	Currently, only **VPC** is supported.
AZ	Yes	You cannot select **Multi-AZ Deployment** for the clone instance. Its deployment mode (single-AZ or multi-AZ) depends on the source instance. If the source instance is deployed in a single AZ, specify the AZ where the master node is located for the clone instance. If the source instance is deployed in multiple AZs, after selecting the master AZ, you need to specify an AZ for a replica in the drop-down list of replica x, where x is the replica number, such as replica 1 and replica 2.
IPv4 Network	Yes	Select the VPC and subnet. We recommend that you select the same VPC in the same region as the CVM instance to be connected to. VPCs are region-specific (e.g., Guangzhou), while subnets are AZ-specific (e.g., Guangzhou Zone 1). One VPC can be divided into one or multiple subnets, which are interconnected over the private network by default. Different VPCs are isolated

		over the private network by default, no matter whether they are in the same or different regions. You can switch the VPC after the instance is purchased. You can also click **Create VPCs** and **Create Subnets** to create a network environment as needed. For more information, see Creating VPC.
Port	Yes	Customize the port number, which is 6379 by default and ranges from 1024 to 65535. You can modify it after the purchase.
Parameter Template	Yes	Select a parameter template for the clone instance. The system automatically matches the appropriate default template based on the selected compatible version and architecture to configure parameters for the instance in batches. If you want to create your own template, click **Create Parameter Template**.
Project	Yes	Assign the clone instance to a project in the drop-down list for easy management. The default project is **Default Project**. You can also click **Create Project** to enter the **Project Management** page and customize the project name to manage your Tencent Cloud resources.
Tag	No	Add tags to the clone instance for easy classification and management. Click **Add** to select tag keys and values.
Security Group	Yes	Set security group rules to control the inbound traffic to the clone instance. You can either select a security group from the **Existing Security Groups** drop-down list or click **Custom Security Groups** to create one and set **inbound rules**. For more information, see Configuring Security Group.
Instance Name	Yes	Set the name for the clone instance, which can contain up to 60 letters, digits, hyphens, and underscores.
Set Password	Yes	Select the password authentication method, which can be **Password Authentication** (default value) or **Passwordless Authentication**.
Password	No	If you select **Password Authentication** for **Set Password**, you need to set an access password for the clone instance according to the following requirements: It can contain [8,30] characters. It must contain at least two of the following four types: lowercase letters, uppercase letters, digits, and symbols ((\)`~!@#$%^&*-+=_|{}[]:;<>,.?/). It cannot start with a slash (/).
Confirm Password	No	Enter the access password for the clone instance again.
Quantity	Yes	You can purchase up to 100 monthly subscribed instances or up to 30 pay-as-you-go instances at a time. You can purchase up to 100 instances in each region.

Validity Period	No	If you select the **Monthly Subscription** billing mode, you need to select the purchase period as needed. The longer the purchase period, the higher the discount.
Auto-Renewal	No	If you select the **Monthly Subscription** billing mode, you can select whether to enable the **Auto-Renewal** feature. If it is enabled, after the instance expires, fees will be automatically deducted monthly from your Tencent Cloud account.
Terms of Service	Yes	Click **TencentDB Terms of Service** to learn about the terms of service related to the use of TencentDB, service fees, usage rules, and intellectual property rights. Click **Service Level Agreement** to learn about the agreement that needs to be followed when you use TencentDB for Redis. Select **I have read and agreed to TencentDB Terms of Service and Service Level Agreement**.

7. Click **Buy Now**. After the purchase is completed, you will be redirected to the instance list. After the instance becomes **Running** in the status, you can use it normally. You can exchange the IPs of the new clone instance and the source instance by modifying the IP to achieve the purpose of data rollback.

Note

After the instance is cloned, the source instance can be retained or terminated as instructed in Returning and Isolating Instance based on your needs.

Configuring Backup File Download Method

Last updated：2023-05-23 10:29:19

Overview

In order to prevent database backup data from being dragged, TencentDB for Redis allows you to download backup files from the specific server on the allowed private network. The private network refers to a local area network (LAN), and cloud services access each other through internal links. The private network services have user attributes, and different users are isolated from each other, which has higher security. You can't download files across regions over the private download address, and can only download them in the region where the TencentDB for Redis instance is located.

Note

Currently, you can't configure backup file download methods in the following regions: Tianjin, Nanjing, Hangzhou, Shenzhen, Virginia, and Jakarta.

Directions

1. Log in to the TencentDB for Redis console.
2. In the left sidebar, select **Database Backup**.
3. At the top of the **Redis - Database Backup** page on the right, select a region, and click the **Download Settings** tab.
4. Click **Edit** next to **Download Settings** to configure download conditions.

Download from Public Network: You can choose **Enable** or **Disable**.

Enable Public Network Access: You can download backup files over both public and private networks.

Disable Public Network Access: You can download backup files over the private network only.

Download Conditions: Customize the VPC and IP address for downloading backup files.

Field is **IP**: In the **Operator** column, select **IN** or **NOT IN**, and enter the included or excluded IP addresses in the **Value** column.

Note

You should separate the values of an IP condition with commas.

If you don't specify any value, the condition won't take effect.

Field is **VPC**: **Operator** defaults to **IN**, and select the included VPC in the drop-down list of **Value**.

5. Click **OK** to complete the settings.

Restoring Data

Last updated：2023-10-20 11:00:49

Overview

TencentDB for Redis Memory Edition (v2.8) and CKV Edition support restoration of an entire instance from a backup file.

Note

TencentDB for Redis Memory Edition (v2.8) and CKV Edition restore data to the source instance based on backup files. TencentDB for Redis Memory Edition (excluding v2.8) clones a complete instance based on the current instance and allows you to roll back data by swapping the IPs of the clone and original instances.

Restoring an instance will interrupt the services provided by the instance.

After the instance is restored, the existing data will be overwritten and cannot be recovered.

If your instance has been ever downgraded, you need to make sure that the instance specification is higher than the restored data capacity; otherwise, the restoration will fail.

Prerequisite

You have backed up the instance data. For more information, see Backing up Data.

Directions

1. Log in to the TencentDB for Redis console and click an instance ID in the instance list to enter the instance management page.
2. On the **Backup and Restoration** tab, locate the desired backup in the backup list, and click **Restore Instance** in the **Operation** column.
3. In the pop-up window, confirm that everything is correct and click **OK**.

Note

If the instance is password-protected, you need to enter the instance password on this page, which is the password you set on the instance purchase page rather than the connection password in the format of **instance ID:instance password** used for instance access.

4. Return to the instance list, where the status of the instance is displayed as **Restoring backup by backup ID**. After the status changes to **Running**, it can be used normally.

Data Migration

Migration Scheme Overview

Last updated：2023-03-14 15:39:05

With the help of DTS as well as redis-sync, redis-dump, and redis-restore tools in the redis-port migration toolkit, TencentDB for Redis offers a variety of data migration schemes for diverse business scenarios.

Migration tools

DTS: It helps migrate your database to the cloud without interrupting your business. In its full + incremental data migration mode, historical data in the source database written before migration and incremental data written during migration can be migrated together.

redis-sync: It supports data migration between Redis instances. It is simulated as a replication node to sync data from the source instance and translate the replicated data into write commands to update the target instance.

redis-dump and redis-restore: redis-dump can be used to back up Redis data into RDB files in an offline environment, and then redis-restore can be used to import the RDB files into the specified Redis instance.

Migration schemes

Migration Scenario	Category	Migration Tools	Access Type	Migration Description
Migration from self-built Redis to TencentDB for Redis	Migration from IDC-based self-built Redis to TencentDB for Redis	DTS (recommended)	The following three access types are supported: Direct Connect: The source database can be interconnected with VPCs through Direct Connect. For detailed directions, see Getting Started. VPN Access: The source database can be interconnected with VPCs through VPN	The migration is online, with full + incremental data sync supported. Supported source Redis versions are 2.8, 3.0, 4.0, 5.0, and 6.2, and supported target Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. Supported architectures include single-node,

			Connections. For detailed directions, see VPN Connections. Intranet (only suitable for Tencent's internal businesses): You need to submit a ticket to add your account to the allowlist.	Redis cluster, Twemproxy, and Sentinel.
		redis-sync	The following two access types are supported: Public Network: The source database can be accessed through a public IP. Intranet: You need to submit a ticket for application.	The migration is online, with full + incremental data sync supported. The source Redis database must allow the `SYNC` or `PSYNC` command. Supported source Redis versions are 2.8, 3.0, and 4.0, and supported target Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
		redis-dump and redis-restore	Offline (the source and target databases don't interconnect over the network).	The migration is offline, with only full data sync supported. Business downtime will be caused. Supported source Redis versions are 2.8, 3.0, and 4.0, and supported target Redis versions are 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
	Migration from CVM-based self-built Redis to TencentDB for Redis	DTS (recommended)	Self-Build on CVM: The source database is deployed in a CVM instance.	The migration is online, with full + incremental data sync supported. Supported source Redis versions are 2.8, 3.0,

				4.0, 5.0, and 6.2, and supported target Redis versions are 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. Supported architectures include single-node, Redis cluster, Twemproxy, and Sentinel.
		redis-sync	VPC: The source database can be accessed through VPC.	The migration is online, with full + incremental data sync supported. The source node must allow the `SYNC` or `PSYNC` command. Supported source Redis versions are 2.8, 3.0, and 4.0, and supported target Redis versions are 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
		redis-dump and redis-restore	Offline (the source and target databases don't interconnect over the network).	The migration is offline, with only full data sync supported. Business downtime will be caused. Supported source Redis versions are 2.8, 3.0, and 4.0, and supported target Redis versions are 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
Migration from TencentDB for Redis to	Migration from TencentDB for Redis to self-built Redis	DTS (recommended)	The following five access types are supported: Public Network: The source database can be	The migration is online, with full + incremental data sync supported.

self-built Redis	(migration off the cloud and multi-cloud sync)		accessed through a public IP. Self-Build on CVM: The source database is deployed in a CVM instance. Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections. CCN: The source database can be interconnected with VPCs through CCN.	You need to submit a ticket for application. Supported target instance types include single-node, Redis cluster, and proxy cluster (which can be deployed by using the proxy provided by Tencent Cloud).
Migration between TencentDB for Redis instances	Migration between TencentDB for Redis instances in different regions	DTS (recommended)	CCN: The source database can be interconnected with VPCs through CCN.	The migration is online, with full + incremental data sync supported. Supported Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
	Migration between TencentDB for Redis instances in the same region	DTS (recommended)	Database: The source database is a TencentDB instance.	The migration is online, with full + incremental data sync supported. Supported Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. This scheme is suitable for large-shard cluster architecture upgrade as it can shorten the downtime.

	Migration between TencentDB for Redis instances on different versions	DTS (recommended)	Database: The source database is a TencentDB instance. VPC: The source database can be accessed through VPC.	The migration is online, with full + incremental data sync supported. Supported Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. For detailed directions, see Version Upgrade with DTS.
	Migration across Tencent Cloud accounts	DTS (recommended)	Database: The source database is a TencentDB instance.	The migration is online, with full + incremental data sync supported. Supported Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. You need to submit a ticket for application.
	Migration from Tencent Cloud standard architecture to cluster architecture	DTS (recommended)	Database: The source database is a TencentDB instance.	The migration is online, with full + incremental data sync supported. Supported Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version. Check the command compatibility as instructed in Check on Migration from Standard Architecture to Cluster Architecture in advance to avoid service execution errors after the upgrade.
	Migration from legacy	redis-restore	Offline (VPC)	Only full data sync is supported. Business

	TencentDB for Redis Cluster Edition instance (purchased before January 1, 2018)			downtime will be caused. For detailed directions, see Migration Guide for Legacy Cluster Edition.
Migration from another cloud to TencentDB for Redis	Migration from Redis in another cloud to TencentDB for Redis	DTS (recommended)	The following three access types are supported: Public Network: The source database can be accessed through a public IP. Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections.	The migration is online, with full + incremental data sync supported. The third-party cloud vendor must allow the `SYNC` or `PSYNC` command. Supported source Redis versions are 2.8, 3.0, 4.0, 5.0, and 6.2, and supported target Redis versions are 2.8, 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
		redis-dump and redis-restore	Offline (the source and target databases don't interconnect over the network).	The migration is offline, with only full data sync supported. Business downtime will be caused. Supported source Redis versions are 2.8, 3.0, and 4.0, and supported target Redis versions are 4.0, 5.0, and 6.2. The target version must be later than or equal to the source version.
Migration from another type of database to TencentDB for Redis	Migration from SSDB to TencentDB for Redis	Siphon	The following four access types are supported: Direct Connect: The source database can be interconnected with	The migration is online, with full + incremental data sync supported. All SSDB kernel versions are supported.

			VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections. VPC: The source database can be accessed through VPC. Intranet: You need to submit a ticket for application.	
	Migration from Pika to TencentDB for Redis	pika-migrate	The following four access types are supported: Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with VPCs through VPN Connections. VPC: The source database can be accessed through VPC. Intranet: You need to submit a ticket for application.	The migration is online, with full + incremental data sync supported. Supported Pika versions are 2.2, 2.3, 3.0, 3.1, and 3.2. Only standalone single-database Pika instances are supported.
	Migration from Codis to TencentDB for Redis	DTS	The following four access types are supported: Direct Connect: The source database can be interconnected with VPCs through Direct Connect. VPN Access: The source database can be interconnected with	The migration is online, with full + incremental data sync supported. The source node must allow the `SYNC` or `PSYNC` command. All versions are supported. For migration of Tencent's internal businesses to the cloud, you need to submit a ticket for application.

			VPCs through VPN Connections. VPC: The source database can be accessed through VPC. Intranet: You need to submit a ticket for application.	
	Migration from Tencent istore to TencentDB for Redis	DTS	Intranet: You need to submit a ticket for application.	The migration is online, with full + incremental data sync supported. The source node must allow the `SYNC` or `PSYNC` command. All istore versions are supported.
	Migration from Memcached to TencentDB for Redis	-	-	Contact Tencent Cloud to customize a migration scheme.
	Migration from Tencent Cloud CKV to TencentDB for Redis	redis-restore	Offline (the source and target databases don't interconnect over the network).	Only full data sync is supported. Business downtime will be caused. For detailed directions, see Migration Guide for Legacy Cluster Edition.

Migration with DTS

Last updated：2023-05-23 10:36:24

Basic Information

Tencent Cloud Data Transmission Service (DTS) is a data transmission service that integrates such features as data migration, sync, and subscription. It helps you migrate your databases to the cloud without interrupting your business and build a high-availability database disaster recovery architecture through real-time sync channels. Its data subscription feature meets your requirements for commercial data mining and async business decoupling.
DTS for Redis currently supports the data migration feature for you to migrate data to TencentDB in a non-stop manner at a time. In addition, in its full + incremental data migration mode, historical data in the source database written before migration and incremental data written during migration can be migrated together.

Overview

The source and target databases supported for data migration through DTS are shown in the table below.

Source	Target	Note
Self-built Redis database in IDC and CVM	TencentDB for Redis and TencentDB for KeeWiDB	Migration to TencentDB for KeeWiDB
TencentDB for Redis	Self-built Redis database in IDC and CVM	Migration from TencentDB for Redis to self-built Redis (migration off the cloud and multi-cloud sync)
Third-party Redis database	TencentDB for Redis and TencentDB for KeeWiDB	Migration to TencentDB for KeeWiDB
TencentDB for Redis	TencentDB for Redis and TencentDB for KeeWiDB	Migration between TencentDB for Redis instances include the following scenarios: Migration between TencentDB for Redis instances in different regions Migration between TencentDB for Redis instances in the same region Migration between TencentDB for Redis instances on different versions Migration across Tencent Cloud accounts Migration from Tencent Cloud standard architecture to the cluster architecture, or vice

		versa. Cluster architecture data is stored in a distributed manner, and there are command compatibility issues with the standard architecture. Before migration, see Check on Migration from Standard Architecture to Cluster Architecture.
Database of other types	TencentDB for Redis	Migration from Codis to TencentDB for Redis Migration from Tencent istore to TencentDB for Redis

Version and Architecture

The supported source databases includes Redis 2.8, 3.0, 3.2, 4.0, 5.0, and 6.2. We recommend that you migrate from an earlier version to a later one to avoid incompatibility.

Supported cluster architectures include single-node, Redis cluster, Codis, and Twemproxy.

The source instance supports the SYNC or PSYNC command.

Restrictions

The number of databases in the source database must be less than or equal to that in the target database.

The **memory** space of the target database must be greater than or equal to 1.5 times the space occupied by the data to be migrated in the source database.

The target database must support the **overwrite** mode. If this mode is not enabled, the target database must be empty.

Only instances in **Running** status are allowed to be migrated. Instances with uninitialized passwords or other ongoing tasks cannot be migrated.

The source database must be a secondary node; otherwise, a warning will be reported for the verification item. The alarm can be canceled based on the business conditions of the primary node.

Note

DTS can’t migrate data in a loop where data is continuously written, resulting in a write storm.

DTS doesn’t support resuming syncing from where it is paused as there is no binlog file. The incremental data is synced in the connected memory. Once disconnected, the incremental data will be lost from the memory. Therefore, the task can’t be paused once it is initiated.

Migration Preparations

Check whether the status of the source and target database instances is normal.

Check whether the access types of the source network environment are interconnected.

Public Network: The source database can be accessed through a public IP.

Self-Build on CVM: The source database is deployed in a CVM instance.

Direct Connect: The source database can be interconnected with VPCs through Direct Connect.

VPN Access: The source database can be interconnected with VPCs through VPN Connections.

CCN: The source database can be interconnected with VPCs through CCN.

Check the memory space: The memory space of the target database must be greater than or equal to 1.5 times the space occupied by the data to be migrated in the source database.

Check the number of databases: The number of databases in the source database must be less than or equal to that in the target database.

The target database must support the **overwrite** mode, and you need to enable it when configuring a migration task. If this mode is not enabled, the target database must be empty.

Note

DTS will perform verification before starting the migration task. For troubleshooting methods, see Check Item Overview.

Check whether there are big keys in the source database.

During the migration, big keys may cause the buffer (client-output-buffer-limit) to overflow, leading to a migration failure.

For TencentDB databases, you can use DBbrain’s performance optimization feature to quickly analyze big keys. For more information, see Memory Analysis.

For non-TencentDB databases, use RDBTools to analyze big keys in Redis.

Evaluate large keys for splitting or cleaning. If you need to retain them, set the source buffer size (client-output-buffer-limit) to infinite.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

Check the limit on the number of TCP connections in the source Linux kernel

If the number of concurrent business requests is high, check the limit on the number of connections in the Linux kernel before the migration. If this value is exceeded, the Linux server will actively disconnect from DTS.

```
echo "net.ipv4.tcp_max_syn_backlog=4096" >> /etc/sysctl.conf
echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
echo "net.ipv4.tcp_abort_on_overflow=0" /etc/sysctl.conf
sysctl -p
```

Check the access permission of the source RDB file directory

Before the migration, check and make sure that the directory where RDB files are stored in the source database is readable; otherwise, the migration will fail.

If the RDB file directory is not readable, run the following command in the source database to set "diskless replication". Then, RDB files will be directly sent to DTS for storage, with no need to be stored in the source database first and then sent.

```
config set repl-diskless-sync yes
```

Check the command compatibility issues when perform migration from standard architecture to the cluster architecture. For directions, see Check on Migration from Standard Architecture to Cluster Architecture for static and dynamic evaluations.

Migration Directions

Step 1. Create a migration task

1. Log in to the DTS console, select **Data Migration**, and click **Create Migration Task**.

2. On the **DTS** page, create a migration task based on the description of the configuration items in the table below, and click **Buy Now**.

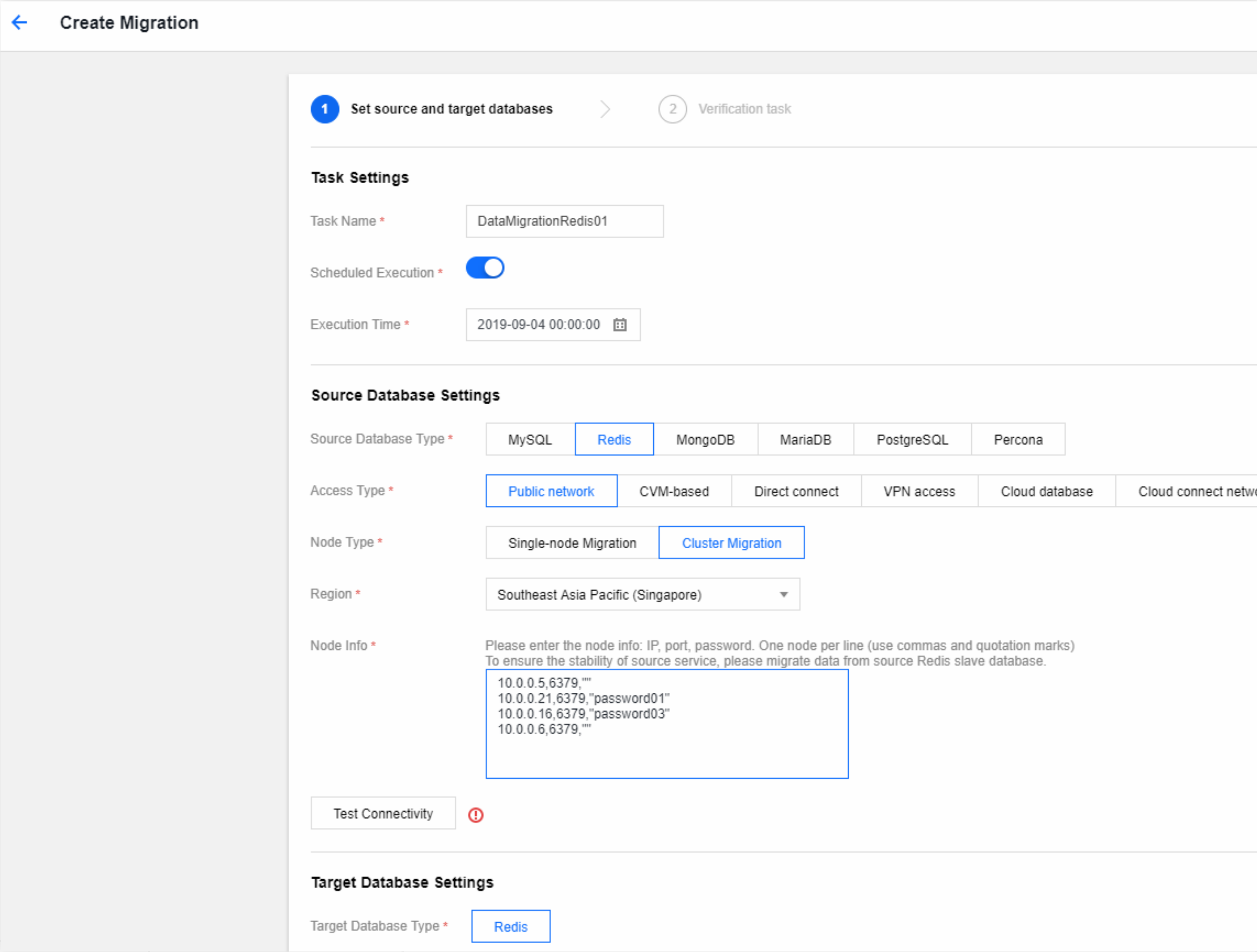
Configuration Items	Note
Service Type	Select **Data Migration**.
Creation Mode	Select a task creation mode. The following two modes are supported. **Create task**: Create a new task. **Create similar task**: Quickly create a similar task based on an existing migration task.
Task ID	When **Create Mode** selects **Create similar task**, this parameter needs to be configured. In the drop-down list, select a similar migration task ID. After configuring this parameter, the configuration items below will be updated based on the configuration in the selected task ID.
Billing Mode	Only **pay-as-you-go** is supported.
Source Instance Type	Select the source database type, which cannot be changed after purchase. Here, select **Redis**.
Source Instance Region	Select the source database region. If the source database is a self-built one, select a region nearest to it.
Target Instance Type	Select the target database type, which cannot be changed after purchase. Here, select **Redis**.
Target Instance Region	Select the target database region.
Specification	Currently, only **Xlarge** is supported.
Tag	Set tag key and value for a migration task, which facilitates efficient task management through tags.
Task Name	**Name After Creation**: After creating the task, you can change its name. **Name Now**: Set a name for the migration task in the input box below. Any special symbols other than hyphens are not supported. Lowercase letters, uppercase letters, and digits are supported. The length limit is as prompted in the console.

	It is recommended to set a name that is easy to identify.
Terms of Service	Be sure to check **I have read and agree to**.
Quantity	The number of created tasks, which ranges from 1 to 10.
Link Configuration Fees	Free of charge currently.

3. Return to the data migration task page. In the migration task list, you can view the task status and progress, and wait for the task to be created.

Step 2. Configure the migration task

1. In the data migration list, find the created migration task.
2. In its **Operation** column, click **Configure**.
3. On the **Set source and target databases** tab of the **Modify Migration Task** configuration wizard, configure the source database network access type, host address, port, and target database instance information based on the table below. Then, test the connectivity and click **Save**.



Task Settings

Task Name * DataMigrationRedis01

Scheduled Execution *

Execution Time * 2019-09-04 00:00:00

Source Database Settings

Source Database Type * MySQL [✓ Redis] MongoDB MariaDB PostgreSQL Percona

Access Type * [✓ Public network] CVM-based Direct connect VPN access Cloud database Cloud connect netw

Node Type * Single-node Migration [✓ Cluster Migration]

Region * Southeast Asia Pacific (Singapore)

Node Info * Please enter the node info: IP, port, password. One node per line (use commas and quotation marks)
To ensure the stability of source service, please migrate data from source Redis slave database.

```
10.0.0.5,6379,""
10.0.0.21,6379,"password01"
10.0.0.16,6379,"password03"
10.0.0.6,6379,""
```

Target Database Settings

Target Database Type * [✓ Redis]

Configuration Category	Configuration Items	Note
Task Configuration	**Task Name**	If the task name has been set when creating a new migration task, this field defaults to the created name. If no task name is configured when creating a new task, you need to set an identifiable task name in the input box. Any special symbols other than hyphens are not supported. Lowercase letters, uppercase letters, and digits are supported. The length limit is as prompted in the console.
	Running Mode	Select **Immediate execution** or **Scheduled execution**. **Immediate execution**: Run immediately after the migration task is configured. **Scheduled execution**: You can set a specific **execution time**, and the task will start running at the set execution time. This method is recommended. After the migration task is configured, you can modify the scheduled execution time before the execution time.

		After the scheduled execution time is set for the migration task, but the task needs to be run immediately before the scheduled time, you can click **Immediate start** in the **Operation** column of the migration task list.
Source Database Settings	**Source Database Type**	The source database type selected during purchase, which cannot be changed.
	Region	The region selected during purchase, which cannot be changed.
	Access Type	Select the type of network connected to the source database. For a third-party cloud database, you can select **Public Network** generally or select **VPN Access**, **Direct Connect**, or **CCN** based on your actual network conditions. **Public Network**: The source database can be accessed through a public IP. **Self-Build on CVM**: The source database is deployed in a CVM instance. **Direct Connect**: The source database can be interconnected with VPCs through Direct Connect. **VPN Access**: The source database can be interconnected with VPCs through VPN Connections. **Database**: The source database is a TencentDB database. CCN: The source database can be interconnected with VPCs through CCN.
	Node Type	Select the cluster deployment type for the source database, which can be **Single-Node Migration** or **Cluster Migration**. **Single-Node Migration**: Refers to the cluster architecture with one master and multiple replicas. **Cluster Migration**: Refers to the cluster architecture comprised of several shards and multiple replica nodes on each shard.
	Host Address	This parameter is displayed when **Single-Node Migration** is selected for **Node Type**. You need to enter the IP address of the host replica node of the source database in the input box.
	Port	This parameter is displayed when**Single-Node Migration** is selected for **Node Type**. You need to enter the port of the source database in the input box.
	Password	This parameter is displayed when **Single-Node Migration** is selected for **Node Type**. You need to enter the access password of the source database in the input box.
	Node Info	This parameter is displayed when **Cluster Migration** is selected for **Node Type**.

		Enter all the shard node addresses and passwords of the source database cluster ("IP:port:password" or "IP:port"), and separate multiple nodes by line breaker. It is recommended to migrate data from the replica node (secondary node) of the source database to avoid affecting the business access of the source database.
Target Database Settings	**Target Database Type**	The target database type selected at the time of purchase, which cannot be changed and is fixed as **Redis**.
	Region	The target database region selected during purchase, which cannot be changed.
	Access Type	Select the network type for the target database access.
	Node Type	Select the deployment type for target database cluster. **Single-Node Migration**: Refers to the cluster architecture with one master and multiple replicas. **Redis Cluster Migration**: Refers to the native Redis cluster architecture, which does not support proxy architecture. **Proxy Cluster Migration**: Refers to the cluster architecture with proxy, such as Codis and Twemproxy.
	Host Address	This parameter is displayed when **Single-Node Migration** is selected for **Node Type**. You need to enter the IP address of the host node of the target database in the input box.
	Port	This parameter is displayed when **Single-Node Migration** is selected for **Node Type**. You need to enter the port of the target database in the input box.
	Password	This parameter is displayed when **Single-Node Migration** is selected for **Node Type**. You need to enter the access password of the target database in the input box.
	Node Info	This parameter is displayed when **Redis Cluster Migration** or **Proxy Cluster Migration** is selected for **Node Type**. Enter all the shard node addresses and passwords of the target database cluster ("IP:port:password" or "IP:port"), and separate multiple nodes by line breaker.

4. On the **Set migration options and select migration objects** tab, confirm that the **Migration Type** is **Full + incremental migration** and the **Migration Object** is **Entire instance**. Then, you can check whether **Enable Overwrite Mode**, click **Save.**

Note

Once the replace mode is enabled, the data will be updated to the target database in a replacing manner without clearing the database, and the data is written from the beginning of the file.

5. On the **Verify task** tab, task verification will be performed automatically. The verification items include network, instance access password, instance capacity, and read-only target instance.

If the verification result is **Failed**, it indicates that a check item failed and the task is blocked. You need to fix the problem and run the verification task again.

If the verification result is **Alarm**, it indicates that a check item doesn't completely meet the requirements, and the task can be continued, but the business will be affected. Before continuing the task, you need to assess whether to ignore the alarm or fix the problem as prompted.

6. After all verification tasks are in **Passed** status, click **Start Task** or **Start Later**.

Start Task: Immediately start the migration task, even if a scheduled execution time is set, the data migration will be started immediately without waiting.

Start Later: If you have set the scheduled execution time when configuring the task, you can select **Start Later**, and the task will start at the execution time.

7. Return to the data migration task list, and the task will be in “Preparing” status. After running for 1-2 minutes, the data migration task will be started.

Step 3. View and manage data migration tasks

In the migration task list, you can view the progress of the migration task. When the status of the migration task changes to **Task successful**, verify the data on the target database. If the verification is correct, determine the cutover time, and route the business system to the target database

If the keys of the source and target databases are the same, click **Complete** in the **Operation** column to stop the data migration task.

Manage migration tasks: In the **Operation** column of the task list, you can perform operations such as retrying, terminating, and pausing the task. For directions, see [Task Management](https://www.tencentcloud.com/document/product/571/42637!cf9a9bfbaa744e05fc9c8d74b2dec935).

Event Alarming and Metric Monitoring

DTS can automatically report event alarms triggered upon migration interruption to keep you informed of any exceptions. For directions, see Configuring Alarm Policy for Data Migration.

DTS allows you to view the monitoring data of various metrics during migration to understand the performance metrics of the system. For more information, see Viewing Monitoring Metric.

FAQs

For common error messages and handling methods during DTS migration, see Common Error Messages.

Version Upgrade with DTS

Last updated：2022-04-28 15:25:41

Overview

TencentDB for Redis 4.0 and 5.0 Memory Edition instances in standard architecture or cluster architecture feature more flexible specification configuration, higher performance, and better functionality. If you are using an earlier version of Redis, we recommend that you upgrade to Redis 4.0 or 5.0 for a better TencentDB service experience. TencentDB for Redis instance version can now be upgraded through Data Transmission Service (DTS) with hot migration, which guarantees instance service continuity during the upgrade process and can update incremental data in real time.

Term	Description
Source instance	Source instance for version upgrade
Target instance	Target instance for version upgrade

Supported versions

Source Instance Version	Target Instance Version			
	4.0 Memory Edition (standard architecture)	4.0 Memory Edition (cluster architecture)	5.0 Memory Edition (standard architecture)	5.0 Memory Edition (cluster architecture)
2.8 Memory Edition (standard architecture)	✓	✓	✓	✓
4.0 Memory Edition (standard architecture)	N/A	✓	✓	✓
4.0 Memory Edition (cluster architecture)	N/A	N/A	✓	✓
5.0 Memory Edition (standard architecture)	N/A	N/A	N/A	✓

Prerequisites

The source TencentDB for Redis instance should be running properly.
You have purchased TencentDB for Redis 4.0 or 5.0 Memory Edition instances in standard architecture or cluster architecture.

Note:

If your existing data is less than 12 GB with incremental data of no more than 60 GB and QPS of no more than 40,000, or your business requires transactional support, TencentDB for Redis 4.0 or 5.0 Memory Edition (standard architecture) is recommended; otherwise, TencentDB for Redis 4.0 or 5.0 Memory Edition (cluster architecture) is your best choice.

Directions

1. Use DTS to migrate data from a source TencentDB for Redis instance to a Redis 4.0 or 5.0 Memory Edition instance in standard architecture or cluster architecture. For more information, see Migration with DTS.
2. After the data sync is completed and the data is verified by your business, you can select the time to disconnect the source Redis instance based on metrics such as business QPS and connect to the destination Redis instance. There are two switching methods:

Log in to the console for switch:

3. Note down the old IP address of the source Redis instance and modify it.
4. Change the network information of the target Redis instance to the VPC subnet of the source Redis instance, and change the IP address of the target instance to the old IP address of the source instance to complete the switchover on the business side. For more information on how to modify network information and IP addresses, see Configuring Network.

Log in to the instance for switch: update the IP of the source Redis instance in the code to the IP of the target Redis instance.

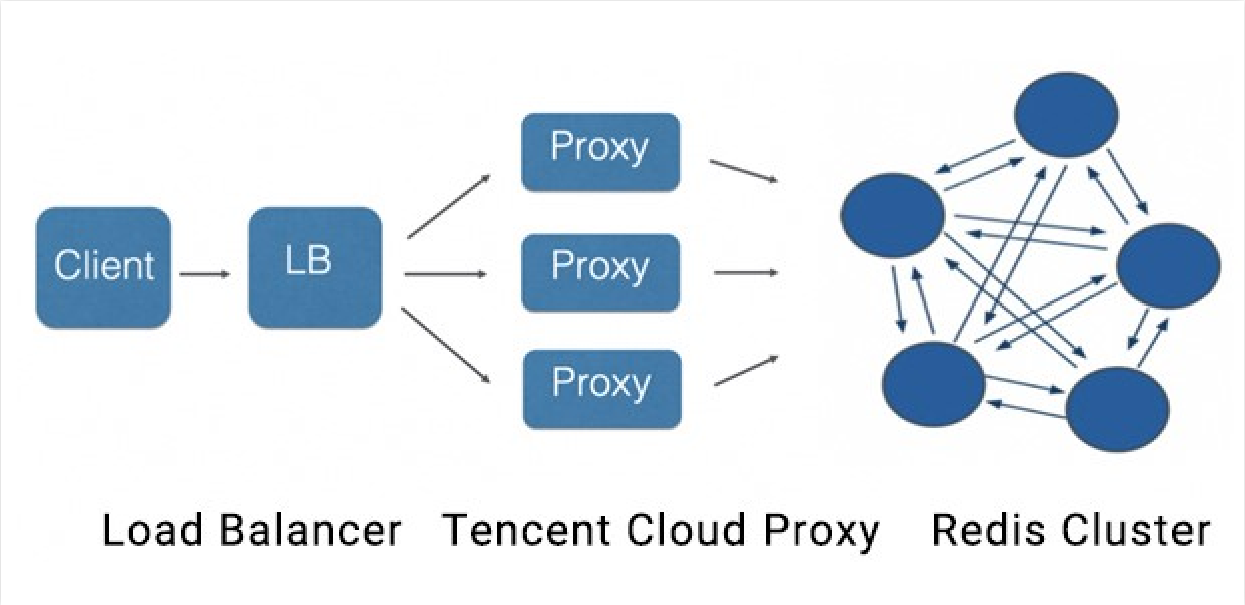
Check on Migration from Standard Architecture to Cluster Architecture

Last updated：2023-05-23 10:56:56

Standard Edition can be your self-created Redis Standalone Edition, master/replica mode, or TencentDB for Redis Memory Edition (Standard Architecture). This document describes the compatibility issues in migrating data from Redis Standard Edition to TencentDB for Redis Memory Edition (Cluster Architecture).

Compatibility Description

TencentDB for Redis Memory Edition (Cluster Architecture) adopts the cluster architecture consisting of Tencent Cloud's proprietary proxy and Redis Community Cluster Edition, which is 100% compatible with Redis Community Cluster Edition commands.



The most challenging problem in migrating data from Standard Edition to Memory Edition (Cluster Architecture) is the command compatibility with usage specifications of Memory Edition (Cluster Architecture). You need to pay attention to the following usage specification issues:

Multikey operation

TencentDB for Redis Memory Edition (Cluster Architecture) uses the hash algorithm to distribute keys to 16,384 slots. For more information on the principle, please see Redis Cluster Specification.

Redis Community Cluster Edition: it does not support any cross-slot multi-key access commands.

TencentDB for Redis Memory Edition (Cluster Architecture): it supports cross-slot multikey access of the `MGET`, `MSET`, and `DEL` commands. This mainly works by using Tencent Cloud's proprietary proxy to implement aggregated command computing among multiple nodes.

Hash tag: in your business, keys that need to engage in multi-key computing can be aggregated into the same slot through a hash tag. For more information on how to use hash tags, see Redis Cluster Specification.

Cross-slot command list:

Command Group	Command	Cross-slot support in Memory Edition (Cluster Architecture)
Keys Group	del	✓
	exists	✓
	rename	x
	renamenx	x
	unlink	x
List Group	rpoplpush	x
	blpop	x
	brpop	x
	brpoplpush	x
Sets Group	sdiff	x
	sdiffstore	x
	sinter	x
	sinterstore	x
	smove	x
	sunion	x
	sunionstore	x
Sorted Sets Group	zinterstore	x
	zunionstore	x
Strings Group	bitop	x

	mget	✓
	mset	✓
	msetnx	x
Hyperloglog Group	pfcount	x
	pfmerge	x
Scripting Group	eval	x
	evalsha	x
	script exists	x
Stream Group	xread	x
	xreadgroup	x

Support for Lua

Memory Edition (Cluster Architecture) supports Lua commands, but cross-slot access to keys in Lua scripts is not supported.

The `Key` parameter must be passed in for the `EVAL` and `EVALSHA` commands; otherwise, they cannot be executed.

The subcommands `LOAD` , `FLUSH` , `KILL` , and `EXIST` of `SCRIPT` will be distributed to all master nodes in the cluster through the proxy.

```
> eval "return {KEYS[1],KEYS[2],ARGV[1],ARGV[2]}" 2 key1 key2 first second
1) "key1"
2) "key2"
3) "first"
4) "second"
```

Note

The `key1` and `key2` parameters must be passed in when you use Lua.

Transaction support

Memory Edition (Cluster Architecture) supports transactions, but cross-slot access to keys in transactions is not supported.
You need to first run the `watch key` command and then the `multi` and `exec` commands in the current version. This operation will be optimized in future versions to eliminate need to run `watch key` first.

Custom commands

Through VIP encapsulation, TencentDB for Redis Memory Edition (cluster architecture) provides a user experience in cluster mode comparable to the standard edition, making it much easier for use in different scenarios. To increase the transparency to OPS, custom commands can be used. Access to each node in the cluster is supported by adding a parameter "node ID" on the right of the original command parameter list, such as `COMMAND arg1 arg2 ... [node ID]` . The node ID can be obtained through the `cluster nodes` command or in the console.

```
10.1.1.1:2000> cluster nodes25b21f1836026bd49c52b2d10e09fbf8c6aa1fdc 10.0.0.15:6379

Native command: `info server`
Custom command:
info server ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171SCAN
Sample:
scan 0 238b45926a528c85f40ae89d6779c802eaa394a2
scan 0 match a* 238b45926a528c85f40ae89d6779c802eaa394a2KEYS
Sample:
keys a* 238b45926a528c85f40ae89d6779c802eaa394a2
```

Client access method

We recommend you use a Standard Edition (e.g., Jedis but not JedisCluster) client to access TencentDB for Redis Memory Edition (Cluster Architecture), as this access method is more efficient and simpler. You can also access through cluster clients, such as JedisCluster.

Codis compatibility

TencentDB for Redis Memory Edition (Cluster Architecture) is 100% compatible with Codis Server commands with no modification to your business required. You can use DTS to quickly migrate data to TencentDB for Redis, which has the following advantages over Codis:

Compatibility with more versions. Codis supports only Redis 3.2 or below, while TencentDB for Redis Memory Edition (Cluster Architecture) supports Redis 4.0 and 5.0 and will be continuously updated in sync with the Redis Community.

TencentDB for Redis Cluster Edition is compatible with more commands. Codis does not support blocking commands such as `BLPOP` and `SUBSCRIBE` .

If a big key occurs in data migration with Codis, the service may become unavailable. In contrast, TencentDB for Redis supports lossless expansion with no fear for big keys.

Compatibility Check

Currently, no tools can be used to exactly determine whether there will be compatibility problems in data migration from Standard Edition to Cluster Edition. You can use the following two tools to evaluate the compatibility before migration. We recommend you perform static evaluation, dynamic evaluation, and business verification before migration to ensure that data can be smoothly migrated.

Static evaluation

1. Download the cluster_migrate_online_check.py static verification tool and use it to run the `info commandstats` command to check whether Standard Edition has ever executed cross-slot commands in order to determine whether there is compatibility problem.

```
Usage:
./cluster_migrate_check.py host port password
```

Note

Enter the corresponding Redis Standard Edition information for `host` , `port` , and `password` .

2. Check whether each item can pass as instructed in Compatibility Description above.

Dynamic evaluation

Download the cluster_migrate_online_check dynamic verification tool and use it to simulate the execution of the `psync` command on the client so as to sync incremental data from Standard Edition to the TencentDB for Redis Memory Edition (Cluster Architecture) in real time. By performing real-time sync, you can check whether there is compatibility problem in write commands. This tool cannot test the compatibility of read commands.

The steps are as follows:

1. Activate TencentDB for Redis Memory Edition (Cluster Architecture) in the console.
2. Use the tool to sync data from Standard Edition to TencentDB for Redis Memory Edition (Cluster Architecture) in real time.
3. After a period of verification (such as 6 or 24 hours), if the tool does not report any errors, the write commands do not have compatibility problems; otherwise, you can get the information of incompatible commands in the error message.

```
Usage:
./cluster_migrate_online_check srcip:srcport srcpasswd dstip:dstport dstpasswd
Environment variable parameters:
export logout=1  // It is used to print command in the console, which is disabled b
export pipeline = 2000  // Number of concurrent pipelines, which is 1,000 by defaul
```

Note

`srcip:srcport` : Redis Standard Edition address information, which is required.

`dstip:dstport` : TencentDB for Redis Memory Edition (Cluster Architecture) address information, which is optional. If it is left empty, the tool can be used as a monitor.

4. Check whether each item can pass as instructed in Compatibility Description above.

Business verification

To ensure successful data migration, we recommend you test the business in the test environment. You can connect the business in the testing environment to the TencentDB for Redis Memory Edition (Cluster Architecture) and confirm whether all features can work properly before data migration.

Migrating Data Online with DTS

For detailed directions, see Migration with DTS.

Self-Created Instance Migration Failure

The `client-output-buffer-limit` parameter value is too small. You are recommended to set it to 512 MB or 1,024 MB by running the following command:

```
config set client-output-buffer-limit "slave 1073741824 1073741824 600"
```

Parameters have not been passed in for the `EVAL` command.

Migration Guide for Legacy Cluster Edition

Last updated：2021-09-13 10:49:34

Overview

The legacy TencentDB for Redis Cluster Edition instances (purchased before January 1, 2018) are on an older version and architecture, which may pose a risk to instance stability. We recommend that you migrate their data to TencentDB for Redis 4.0 Memory Edition (standard or cluster architecture).

TencentDB for Redis 4.0 comes with more flexible configuration, higher performance, and more comprehensive features. This document describes how to migrate data from your legacy instances to TencentDB for Redis 4.0 Memory Edition instances (standard or cluster architecture).

Note:

To avoid data loss, you need to stop writing to your legacy instances before migrating their data to TencentDB for Redis Memory Edition instances (standard or cluster architecture), as hot migration is unsupported in this case. You can configure a security group or reset the password to block all access requests. You can check QPS on the monitoring page: if it is zero, all access requests are blocked successfully.

Prerequisites

You have purchased TencentDB for Redis Memory Edition instances (standard or cluster architecture).

Note:

If your existing data is less than 12 GB with expected incremental data of no more than 60 GB and QPS of no more than 40,000, or you need to use transaction commands, TencentDB for Redis 4.0 Memory Edition (standard architecture) is recommended; otherwise, TencentDB for Redis 4.0 Memory Edition (cluster architecture) is your best choice. The cluster architecture is compatible with all commands of the legacy instances except that it does not support transaction commands.

You have a CVM instance ready for data import, which needs to have sufficient disk capacity to accommodate the existing data.

The data import tool, redis-port, has been installed. For more information on the tool usage and download address, see Migration with redis-port.

Directions

1. Stop writing to the legacy Redis Cluster Edition instance.

2. Back up its data in the TencentDB for Redis console. The time backup takes depends on the amount of data. After backup is completed, an RDB file will be generated.
3. After backup is completed, you can view the backup file in the backup list. Click **Export** to generate an RDB file with a download link, copy the link, and use it to download the backup file from the CVM instance over a private network. Cross-AZ download is unsupported.
4. Initialize the password of the newly purchased TencentDB for Redis 4.0 Memory Edition (standard or cluster architecture) instance, and use redis-port to import the RDB file to it.

```
./redis-restore dump.rdb -t 127.0.0.1:6379
```

5. After the data import is completed, check whether it is successful by viewing the memory usage in the **Specs Info** block on the instance details page.

6. Migrate your application to the new instance by replacing the IP of the legacy instance in the code with that of the new instance.

Pika-to-Redis Data Migration Scheme

Last updated：2023-05-23 10:54:15

Migration Principle

Data can be migrated from Pika to Redis online with support for full sync and incremental sync. The pika-migrate tool is virtualized as the Pika replica to get data from the master and then forward the data to Redis. Incremental sync is supported to implement online hot migration.

1. pika-migrate requests the full database data from the master and the corresponding binlog offsets through DBSync.
2. After getting the current full data from the master, the tool scans the database and then packages and forwards the data to Redis.
3. The tool performs incremental sync from the master through the obtained binlog offsets. During the incremental sync, it reassembles the binlog obtained from the master into a Redis command and forward the command to Redis.

Supported versions

This tool applies to Pika 3.2.0 or later in standalone mode with a single database. If the Pika version is earlier than 3.2.0, you need to upgrade the kernel version to 3.2.0.

Note

Pika supports different data structures using keys with the same name, while Redis does not. Therefore, in scenarios involving data structures using keys with the same name, the first data structure migrated to Redis will apply, and other data structures will be discarded.

This tool supports hot migration from Pika only in standalone mode with a single database. It will report an error and exit in case of the cluster mode or a multi-database scenario.

To prevent the tool from triggering repeated full syncs due to the clearing of binlogs from the master and writing dirty data to Redis, the tool is self-protected by reporting an error and exiting when full sync is triggered for the second time.

Migration Directions

1. Run the following command in the Pika master to keep 10,000 binlog files.

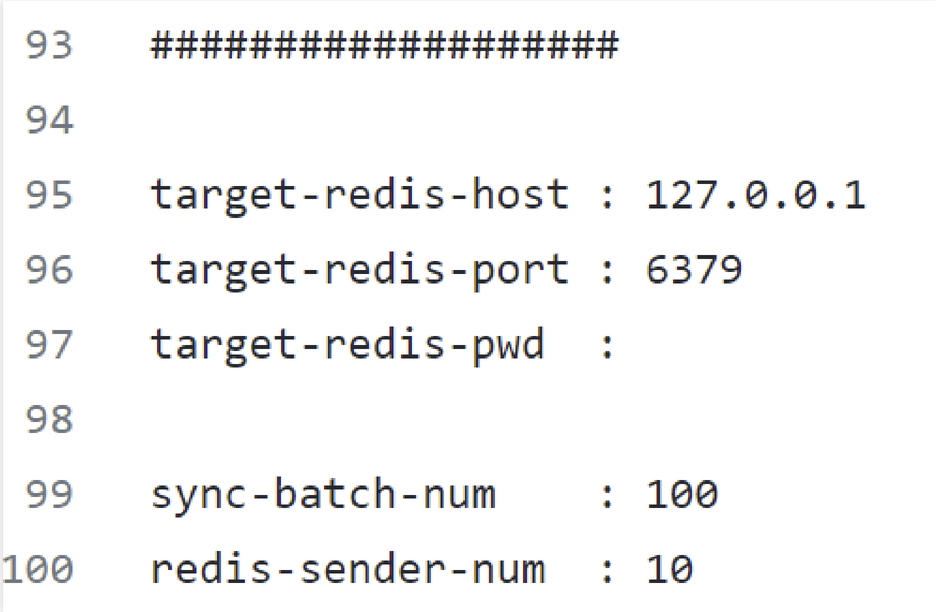
```
config set expire-logs-nums 10000
```

Note

It may take a long time for pika-port to write full data to Redis, which will cause the original binlog offsets to be cleared from the master. It is necessary to keep 10,000 binlog files in the Pika master and ensure that they still exist when the tool requests incremental sync.

As binlog files use disk capacity, you can determine the number of binlog files to be retained as needed.

2. Modify the following parameters in the `pika.conf` configuration file of the migration tool.



```
###################

target-redis-host : 127.0.0.1
target-redis-port : 6379
target-redis-pwd  :

sync-batch-num    : 100
redis-sender-num  : 10
```

target-redis-host: Specify the IP address of Redis.

target-redis-port: Specify the port number of Redis.

target-redis-pwd: Specify the password of the default Redis account.

sync-batch-num: Specify `sync-batch-num` data entries received by pika-migrate from the master for packaging and sending to Redis to increase the forwarding efficiency.

redis-sender-num: Specify `redis-sender-num` threads for forwarding packets. The forwarding command distributes the data to different threads based on the hash value of the key for sending, so you don't need to worry about messy data caused by multi-threaded sending.

3. Run the following command in the path of the toolkit to start the pika-migrate tool and view the returned information.

```
pika -c pika.conf
```

4. Run the following command to disguise the migration tool as the slave and request sync from the master, and then observe whether an error is reported.

```
slaveof ip port force
```

5. After confirming that the master/slave relationship is successfully established, pika-migrate will forward data to the target Redis database. Run the following command to check for the delay in the master/slave sync. You can write a special key and then check in Redis whether the key can be obtained immediately, so as to determine whether the data sync is completed.

```
info Replication
```

SSDB-to-Redis Data Migration Scheme

Last updated：2022-09-14 18:36:29

How migration works

The Siphon migration tool developed based on Go is disguised as the SSDB slave to subscribe to data and sync the data to Redis.
Siphon is automatically connected to the SSDB server upon startup to perform key addressing. It starts sync from the starting position until all the existing data is synced and then syncs the incremental data. That is to say, the tool establishes a persistent connection after startup and keeps running.

Tool and version descriptions

Migration tool: Siphon. It applies to all SSDB kernel versions.
If SSDB involves big keys or over 100 million keys, you need to submit ticket to obtain the modified Siphon V2 version to improve data synchronization efficiency

Note:

The modified tool solves the problem of inefficiency of the native edition in data sync. In particular, it increases the efficiency in syncing big keys such as hashes and sorted sets (zsets) by about 12 times.

Notes

Migration from SSDB in single-instance mode to Redis Cluster Edition involves logic compatibility issues, such as cross-slot transactions and pipelines.

Migration directions

1. Collect the parameters required to run the migration command as shown below:
-p: Specifies the number of concurrent threads.
-f: Specifies the address of the SSDB server.
-t: Specifies the address of the Redis server.
-T: Specifies the password of the Redis database.
2. Start the migration tool with `siphon_v2 sync` and view the migration log.

```
./siphon_v2 sync -p 1 -f X.X.X.X:8888 -t X.X.X.x:6379 -T XXX
```

The status is displayed as follows after the command is executed:

Copy Start: Indicates the start of full data sync.

Copy Stop: Indicates the end of full data sync.

3. Wait for new data to be generated and incrementally synced to Redis without exiting the process.

Common Error Messages

Last updated：2023-05-23 10:57:56

Issue 1

Description

The following error message is displayed during migration with DTS:

```
[launch]state:6 #rdb rdbfile:./tmp1600869159_89068.rdb rdbsize:2753701723 rdb_write
```

Causes

Check the log of the source Redis database. If it contains the following message, the `client-output-buffer-limit` configured for the source database is exceeded.

```
psync scheduled to be closed ASAP for overcoming of output buffer limits
```

Solution

Run the following command to set the `client-output-buffer-limit` to infinite and initiate the DTS task again.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

Issue 2

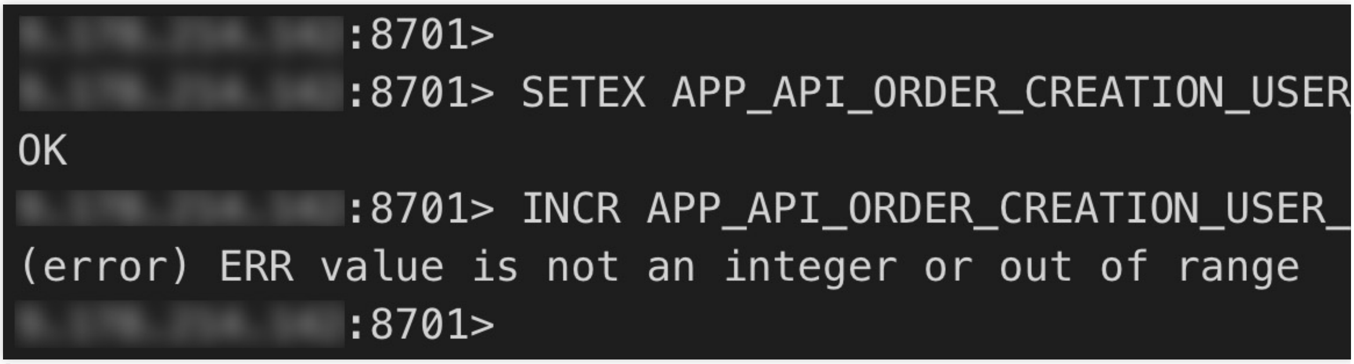
Description

The following error message is displayed during migration with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1600395232_34851.rdb rdbsize:107994104 rdb_writed
```

Causes

By capturing packets on two DTS Syncer instances in the region, it was found that the value of the key was characters rather than a number, causing the INCR execution to fail.



```
[redacted]:8701>
[redacted]:8701> SETEX APP_API_ORDER_CREATION_USER
OK
[redacted]:8701> INCR APP_API_ORDER_CREATION_USER_
(error) ERR value is not an integer or out of range
[redacted]:8701>
```

Solution

Delete the relevant key and initiate the DTS migration again.

Issue 3

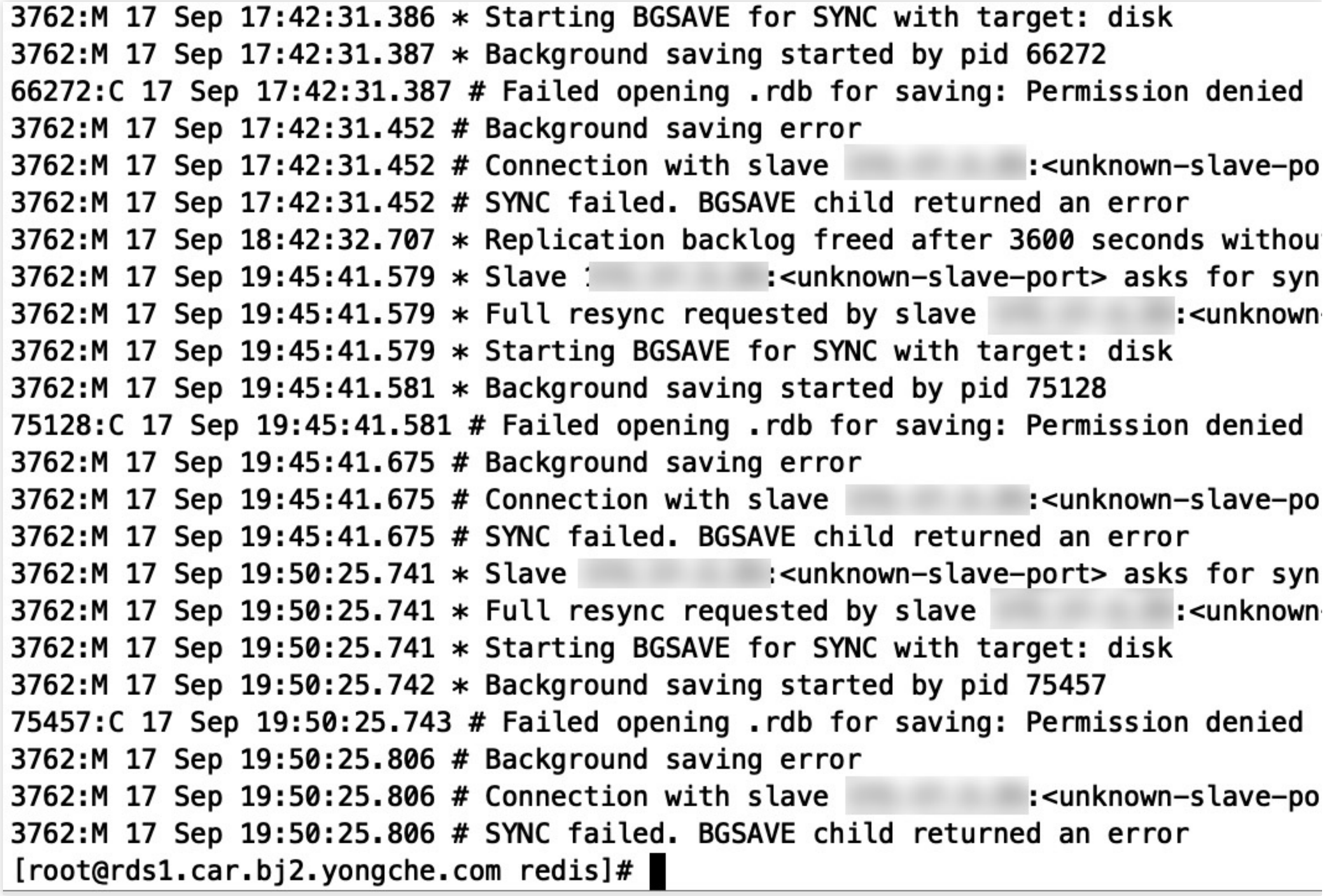
Description

The following error message is displayed during migration with DTS:

```
errmsg:Error reading bulk length while SYNCing:Operation now in progress read rdb l
```

Causes

The error message of the source instance reveals that the RDB file does not have permission to access the directory.



```
3762:M 17 Sep 17:42:31.386 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 17:42:31.387 * Background saving started by pid 66272
66272:C 17 Sep 17:42:31.387 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 17:42:31.452 # Background saving error
3762:M 17 Sep 17:42:31.452 # Connection with slave [redacted]:<unknown-slave-po
3762:M 17 Sep 17:42:31.452 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 18:42:32.707 * Replication backlog freed after 3600 seconds withou
3762:M 17 Sep 19:45:41.579 * Slave [redacted]:<unknown-slave-port> asks for syn
3762:M 17 Sep 19:45:41.579 * Full resync requested by slave [redacted]:<unknown
3762:M 17 Sep 19:45:41.579 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:45:41.581 * Background saving started by pid 75128
75128:C 17 Sep 19:45:41.581 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:45:41.675 # Background saving error
3762:M 17 Sep 19:45:41.675 # Connection with slave [redacted]:<unknown-slave-po
3762:M 17 Sep 19:45:41.675 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 19:50:25.741 * Slave [redacted]:<unknown-slave-port> asks for syn
3762:M 17 Sep 19:50:25.741 * Full resync requested by slave [redacted]:<unknown
3762:M 17 Sep 19:50:25.741 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:50:25.742 * Background saving started by pid 75457
75457:C 17 Sep 19:50:25.743 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:50:25.806 # Background saving error
3762:M 17 Sep 19:50:25.806 # Connection with slave [redacted]:<unknown-slave-po
3762:M 17 Sep 19:50:25.806 # SYNC failed. BGSAVE child returned an error
[root@rds1.car.bj2.yongche.com redis]# █
```

Solution

Run the following command to set diskless replication and initiate the DTS task again.

```
config set repl-diskless-sync yes
```

Issue 4

Description

The following error message is displayed during migration with DTS:

```
[launch]state:6 #rdb rdbfile:./tmp1597977351_20216.rdb rdbsize:24282193511 rdb_writ
```

Causes

The disk space in the DTS Syncer instance is insufficient.

Solution

Clear the disk in the DTS Syncer instance or mount a new disk and then initiate the DTS task again.

Issue 5

Description

The following error message is displayed during migration with DTS:

```
[launch]state:5 #rdb rdbfile: rdbsize:0 rdb_writed_size:0 rdb_parsed_size:0 rdb_par
```

Solution

Disconnect the source database and adjust the limit on the connections to the source system kernel.

```
echo "net.ipv4.tcp_max_syn_backlog=4096" >> /etc/sysctl.conf
echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
echo "net.ipv4.tcp_abort_on_overflow=0" /etc/sysctl.conf
sysctl -p
```

Issue 6

Description

The following error message is displayed during migration from Redis Memory Edition (Standard Architecture) to Cluster Architecture with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1645683629_34614.rdb rdbsize:781035471 rdb_writed
```

Causes

The database involves multi-key, transactional, or cross-slot operations. For more information, see Check on Migration from Standard Architecture to Cluster Architecture.

Solution

Migrate the data to a Standard Architecture instance in the cloud, or change the business logic to clear multi-key operations.

Issue 7

Description

The following error message is displayed during migration with DTS:

```
[launch]state:7 #rdb rdbfile:./tmp1633836033_79441.rdb rdbsize:1008499748 rdb_write
```

You can see the following information in the execution log of the source node:

```
44:M 05 Jun 03:31:06.728 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:31:06.978 * Background saving started by pid 89
89:C 05 Jun 03:32:08.417 # Error moving temp DB file temp-89.rdb on the final desti
44:M 05 Jun 03:32:08.698 # Background saving error
44:M 05 Jun 03:32:08.698 # Connection with slave 10.xx.xx.119:<unknown-slave-port>
44:M 05 Jun 03:32:08.698 # SYNC failed. BGSAVE child returned an error
44:M 05 Jun 03:50:24.626 * Slave 10.xx.xx.119:<unknown-slave-port> asks for synchro
```

```
44:M 05 Jun 03:50:24.626 * Full resync requested by slave 10.xx.xx.119:<unknown-sla
44:M 05 Jun 03:50:24.626 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:50:24.880 * Background saving started by pid 90
90:C 05 Jun 03:51:22.585 * DB saved on disk
90:C 05 Jun 03:51:22.739 * RDB: 280 MB of memory used by copy-on-write
44:M 05 Jun 03:51:23.008 * Background saving terminated with success
44:M 05 Jun 03:51:27.898 * Synchronization with slave 10.xx.xx.119:<unknown-slave-p
44:M 05 Jun 03:52:19.531 # Connection with slave client id #317862457 lost.
```

Causes

This is often because the connection of the DTS task to the source node timed out due to network environment issues, big keys contained in the database, or `client-output-buffer-limit` overflows on the source node.

Solution

Check the source network environment for any issue. For detailed directions, see Database Connection Check.

Clear big keys in the source database. You can quickly locate, assess, and delete big keys as instructed in Memory Analysis.

Run the following command to set the `client-output-buffer-limit` to infinite on the source node.

```
config set client-output-buffer-limit 'slave 0 0 0'
```

Issue 8

Description

The following error message is displayed during migration with DTS:

```
[launch]state:7 #rdb rdbfile:./tmp1654365384_70581.rdb rdbsize:1664871634 rdb_write
```

Causes

If this error message is displayed when you retry a failed DTS task, it generally indicates that the target node is not empty or the memory is full.

Solution

Clear the target node and try again, or submit a ticket to apply for DTS overwrite.

Issue 9

Description

The following error message is displayed during migration with DTS:

```
[launch]state:8 #rdb rdbfile:./tmp1653290250_19158.rdb rdbsize:1721160435 rdb_write
```

Causes

The Redis node of the target instance experienced an HA master/replica switch, or the proxy node experienced a failover, causing the sync task to fail.

Solution

Create a new DTS task and configure the new node after the HA switch as the target node for data migration.

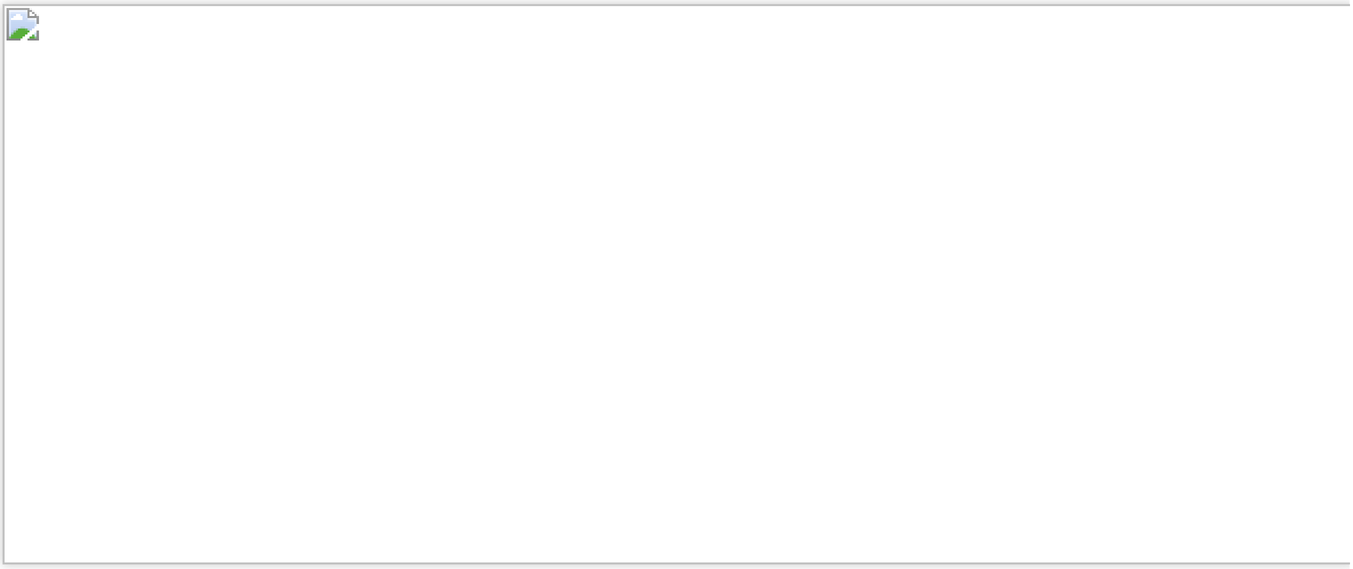
Issue 10

Description

When the memory eviction policy of the target instance is set to `allkey-lru` during migration with DTS, the following error message is displayed:

```
[launch]state:8 #rdb rdbfile:./tmp1638263556_29975.rdb rdbsize:597343276 rdb_writed
```

Or, the following error is reported:



Causes

The memory of the target instance is smaller than the memory used by the data to be migrated from the source database.

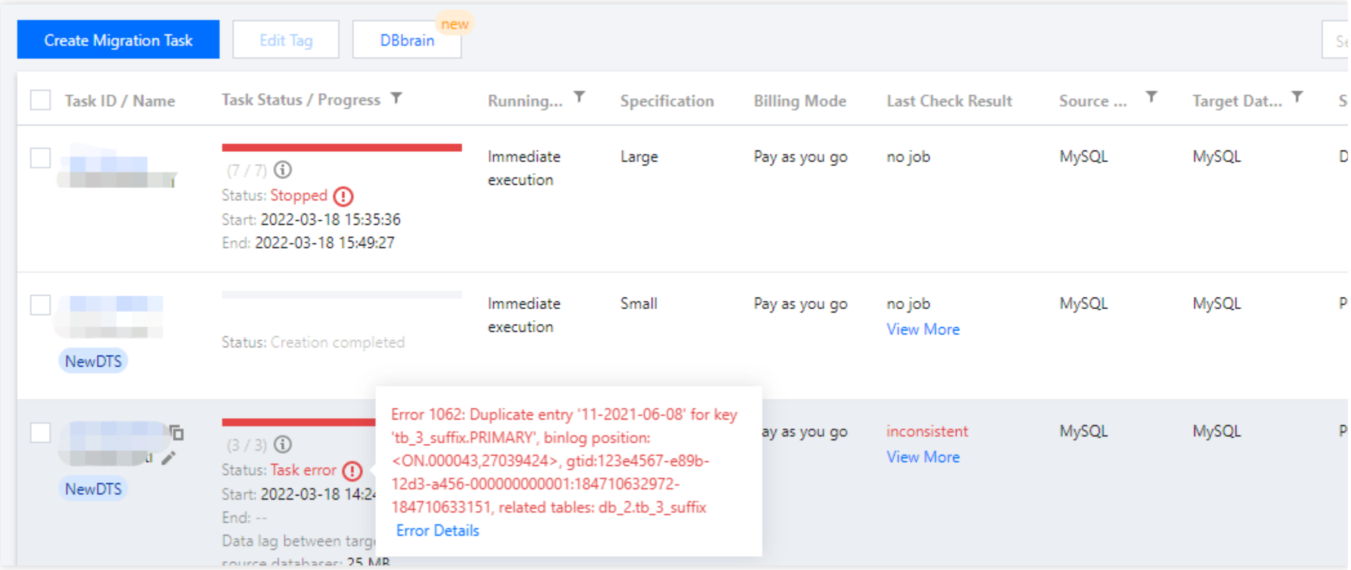
Solution

Expand the memory of the target instance and then initiate a new DTS migration task. For detailed directions, see Changing Instance Specification.

Issue 11

Description

The following error message is displayed during migration with DTS:



Task ID / Name	Task Status / Progress	Running...	Specification	Billing Mode	Last Check Result	Source ...	Target Dat...	S
[redacted]	(7 / 7) Status: Stopped Start: 2022-03-18 15:35:36 End: 2022-03-18 15:49:27	Immediate execution	Large	Pay as you go	no job	MySQL	MySQL	D
[redacted] NewDTS	Status: Creation completed	Immediate execution	Small	Pay as you go	no job View More	MySQL	MySQL	P
[redacted] NewDTS	(3 / 3) Status: Task error Start: 2022-03-18 14:2 End: -- Data lag between targ source databases: 25 MB			ay as you go	inconsistent View More	MySQL	MySQL	P

Error 1062: Duplicate entry '11-2021-06-08' for key 'tb_3_suffix.PRIMARY', binlog position: <ON.000043,27039424>, gtid:123e4567-e89b-12d3-a456-000000000001:184710632972-184710633151, related tables: db_2.tb_3_suffix
Error Details

Causes

If an error is reported when you start a DTS migration task with a proxy, it is usually because the bandwidth of the proxy is insufficient.

Solution

Expand the bandwidth of the proxy or perform migration tasks serially in sequence. For detailed directions, see Bandwidth Adjustment.

Issue 12

Description

The following error message is displayed during migration with DTS:

```
[launch]SrcInstance nodes has changed.
```

Causes

The source node experienced an HA master/replica switch, causing the DTS task sync to fail.

Solution

Create a new DTS task and configure the new node after the HA switch as the target node for data migration.

FAQs

Last updated：2023-05-23 10:51:51

Will the data in the source database be deleted after migration with DTS?

No. Data migration with DTS essentially replicates the data from the source database to the target database without affecting such data.

How does data migration with DTS affect the target database?

When data is migrated to the target database, the system will verify whether the source and target databases have tables with the same name, and if so, the verification will fail, and you will be prompted to make changes first.

Does DTS support migration from one offline database to another?

No. DTS only supports migration from self-built databases, databases in other clouds, or TencentDB databases to TencentDB databases.

Does DTS support data migration between TencentDB instances under two different Tencent Cloud accounts?

Yes. For migration between TencentDB instances under two different Tencent Cloud accounts, you need to log in to DTS with the Tencent Cloud account of the target instance. For detailed directions, see Cross-Account TencentDB Instance Migration.

Can I configure multiple DTS tasks for migration from the same source database to different TencentDB instances?

Yes. You can migrate data from the same source database to multiple target databases and vice versa, but multiple concurrent tasks may increase the access pressure on the source and target databases and thus slow down the migration. If you need to create multiple migration tasks for the same source database, then after creating the first task, you can quickly create similar tasks by clicking **More** > **Create similar task** in the **Operation** column.

Does DTS support scheduled automatic migration?

Yes. When modifying the configuration for a created data migration task, you can select scheduled migration and specify the start time.

Can I monitor the task progress during migration?

Yes. You can log in to the DTS console and view the migration task progress on the **Data Migration** page.

Why is there a 15-day limit on incremental migration?

Currently, incremental migration is performed through the nearest proxy server via Tencent Cloud Direct Connect, which eliminates network jitters and ensures the quality of data transfer. The 15-day limit can reduce the connection pressure on the proxy server and is only intended for reasonable utilization of resources for migration. Currently, connections will not be force closed after 15 days.

How is the data accuracy ensured during data migration?

DTS uses Tencent Cloud's proprietary data migration architecture to verify the data accuracy in real time and quickly detect and correct errors. This guarantees the reliability of the transferred data.

Why does data verification require that the source database instance not be read-only?

This is because data verification requires creating a new database `__tencentdb__` in the source instance and writing the checksum table to the database. If the instance is read-only, data verification will be skipped.

Can I specify tables for migration with DTS?

Yes. You can select the entire instance or specify tables as the migration object.

When does data migration stop?

When you select incremental migration, if it takes a long time before the task stops, you may need to stop it by yourself.

If you select **Structural migration** or **Full migration** as the **Migration Type**, the task will automatically stop upon completion.

Select **Full + Incremental migration**: after full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Complete** to manually stop the incremental data sync.

Manually complete incremental data sync and business switchover at appropriate time.

Check whether the migration task is in the incremental sync stage without any lag. If so, stop writing data to the source database for a few minutes.

Manually complete incremental sync when the data gap between the target and the source databases is 0 MB and the time lag between them is 0 seconds.

Why does the data size change before and after full migration?

This is because the fragmented spaces of the source and target databases are different, and the source database may contain data holes. In this case, after full migration is completed, the table storage space in the target database may be smaller than that in the source database. We recommend you perform a data consistency check as instructed in Creating Data Consistency Check Task after the migration is completed to check whether the contents of the source database and the target database are consistent.

Is double write supported during data migration?

No. Writing data to both the source and target databases during migration may cause data inconsistency.

Does DTS support cross-region database migration?

Yes. You can implement cross-region data transfer over the public network.

What Redis versions does DTS support migrating?

The target database version must be later than or equal to the source database version. Supported source and target versions are as follows:

Source	Target
Self-built database (including CVM-based ones) on Redis 2.8, 3.0, 3.2, 4.0, or 5.0	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, or 5.0
TencentDB for Redis 2.8, 3.0, 3.2, 4.0, or 5.0	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, or 5.0
Databases in other clouds on Redis 2.8, 3.0, 3.2, 4.0, or 5.0	TencentDB for Redis 2.8, 3.0, 3.2, 4.0, or 5.0

Can I create multiple DTS tasks for migration from the same source database to different TencentDB instances?

Yes. You can migrate data from the same source database to multiple target databases and vice versa, but multiple concurrent tasks may increase the access pressure on the source and target databases and thus slow down the migration. If you need to create multiple migration tasks for the same source database, then after creating the first task, you can quickly create similar tasks by clicking **Create similar task** in the **Operation** column.

Migration with redis-port

Last updated：2023-04-18 11:29:06

Tools

redis-port (Linux 64-bit) is a collection of open-source tools mainly used for database sync, data import, and data export between Redis nodes and supports cross-version Redis data migration. The toolkit contains the following tools:

redis-sync: It is used for data migration between Redis instances.

redis-restore: It supports importing Redis backup files (in RDB format) to the specified Redis instance.

redis-dump: It supports backing up Redis data in RDB format.

redis-decode: It supports decoding Redis RDB backup files into readable files.

Compatible Versions

Source instances on Redis 2.8, 3.0, and 4.0 are supported.

Target instances on Redis 2.8, 3.0, 3.2, and 4.0 and in all editions of TencentDB are supported, including Redis Memory Edition and CKV Edition.

Online Migration with redis-sync

How it works

redis-sync has two modules which are simulated as replication nodes to continuously sync data from the source instance and translate the replicated data into write commands to update the target instance.

Data replication is done in two phases: full sync and incremental sync.

Notes:

The database capacity of the target instance should be greater than that of the source instance; otherwise, the migration will fail.

If migration is interrupted for causes such as network failure, you need to empty the target instance first and then perform migration again; otherwise, there may be dirty data.

The progress of migration is displayed in the log, where "sync: rdb = 9063349 - [100.00%]" indicates that full data has been synced and incremental data sync is in progress, while "speed=(0/0,0/0,0)" indicates that incremental data has been synced.

You can stop data sync and migration by pressing Ctrl + C or through other means.

Parameter description

-n: Number of concurrent write tasks. We recommend that you leave it empty or set it to CPU core quantity * 2.

-m: Source instance address in the format of `"password"@ip:port` or `ip:port` (in password-free mode).

-t: Target instance address in the format of `"password"@ip:port` or `ip:port` (in password-free mode).

--tmpfile=FILE: Temporary filename.

--tmpfile-size=SIZE: Maximum size of the temporary file.

--help: Help command.

Sample

```
./redis-sync -m 127.0.0.1:6379 -t "xxx2018"@10.0.5.8:6379
```

Output log

```
[root@VM_5_16_centos bin]# ./redis-sync -m 127.0.0.1:6379 -t "xxx2018"@10.0.5.8:637
2019/02/21 09:56:00 sync.go:76: [INFO] sync: master = "127.0.0.1:6379", target = "x
2019/02/21 09:56:01 sync.go:103: [INFO] +
2019/02/21 09:56:01 sync.go:109: [INFO] sync: runid = "f63e2ad58e2fcc15c8cc122f1577
2019/02/21 09:56:01 sync.go:110: [INFO] sync: rdb file = 9063349 (8.64mb)
2019/02/21 09:56:01 sync.go:208: [INFO] sync: (r/f,s/f,s) = (read,rdb.forward,rdb.s
2019/02/21 09:56:02 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)
```

```
2019/02/21 09:56:03 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)
2019/02/21 09:57:54 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)
```

Data Import with redis-restore

redis-restore supports importing Redis backup files (in RDB format) on Redis 2.8, 3.0, 3.2, and 4.0 as well as AOF files into the specified Redis instance.

Parameter description

-n: Number of concurrent write tasks. We recommend that you leave it empty or set it to CPU core quantity * 2.
-i: RDB file path.
-t: Target instance address in the format of `"password"@ip:port` or `ip:port` (in password-free mode).
-a: AOF file path.
--db=DB: Database ID of the target Redis instance for backup file import, which should be the same as that of the source instance.
--unixtime-in-milliseconds=EXPR: The key expiration time value is updated in the process of data import.
--help: Help command.

Sample

```
./redis-restore dump.rdb -t 127.0.0.1:6379
```

Data Backup with redis-dump

redis-dump supports backing up Redis data into RDB files and incremental data into AOF files.

Notes

TencentDB for Redis currently does not support backing up data through redis-dump. You can back up and download data in the TencentDB for Redis console or through APIs. However, you can use redis-dump to back up your self-built Redis instances.

Parameter description

-n: Number of concurrent write tasks. We recommend that you leave it empty or set it to CPU core quantity * 2.

-m: Redis instance address in the format of `"password"@ip:port` or `ip:port` (in password-free mode).

-o: Path of the output RDB file.

-a: Path to the outputted AOF file.

--help: Help command.

Sample

```
./redis-dump  127.0.0.1:6379 -o dump.rdb
```

Account and Password

Password-Free Access

Last updated：2024-03-13 11:35:29

TencentDB for Redis supports password-enabled and password-free access.

Note:

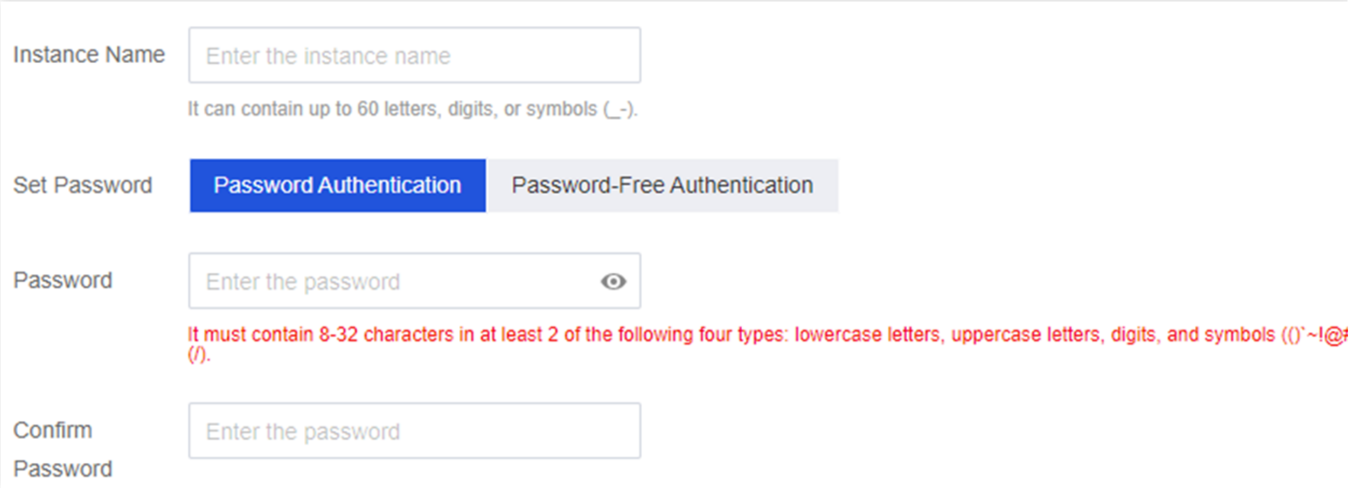
For the sake of data security, you are not recommended to enable password-free access.

After password-free access is enabled, you are recommended to limit the number of accessing servers using a security group.

Setting Password-Free Access

Select password-free access when creating an instance

1. Log in to the TencentDB for Redis console, and click **Create Instance** in the instance list.
2. In the **Set Password** section on the purchase page, select **Password-Free Authentication**. After the instance is created successfully, it can be accessed without password.



Instance Name: Enter the instance name

It can contain up to 60 letters, digits, or symbols (_-).

Set Password: Password Authentication | Password-Free Authentication

Password: Enter the password

It must contain 8-32 characters in at least 2 of the following four types: lowercase letters, uppercase letters, digits, and symbols (()`~!@#
(/).

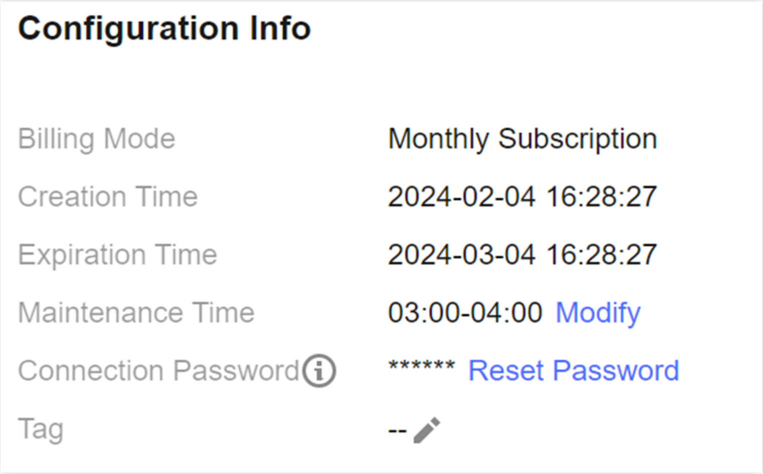
Confirm Password: Enter the password

Enable password-free access for existing instances

In the instance list, click an instance name to enter the **Instance Details** page, and click **Password-Free Access** in **Configuration Info** > **Connection Password**.

Viewing Password-Free Access Status

In the instance list, click an instance name to enter the **Instance Details** page, and check whether **Password-Free Access** is enabled in **Configuration Info** > **Connection Password**.



Configuration Info

Billing Mode	Monthly Subscription
Creation Time	2024-02-04 16:28:27
Expiration Time	2024-03-04 16:28:27
Maintenance Time	03:00-04:00 Modify
Connection Password ⓘ	****** Reset Password
Tag	--

Disabling Password-Free Access

In **Configuration Info** > **Connection Password**, you can disable **Password-Free Access** by resetting your password in **Reset Password**.

Related APIs

API	Description
ResetPassword	Resets password. If you leave the `Password` parameter empty, password-free access will be enabled.

Managing Account

Last updated：2024-03-13 11:37:07

Overview

TencentDB for Redis provides read-write permission control and routing policy control through the account mechanism, which helps meet the needs of business permission management in complex scenarios. Currently, only the TencentDB for Redis Memory Edition (excluding Redis 2.8) supports account settings.

Account types

Default account: An account with only a password. When creating an instance, set the password for accessing the database as instructed in Creating TencentDB for Redis Instance.

Custom account: An account with an account name. The authentication method of a custom account is `account name@password` , which is used as the password parameter for accessing Redis, such as `redis-cli -h 1.1.1.1 -p 6379 -a readonlyuser@password` .

Account match priority

When there is a default account with the @ separator, it will be matched first before a custom account. Custom accounts will be matched with the first @ symbol as the separator.

TencentDB for Redis uses a passwordless authentication method different from that of Redis Community Edition. Specifically, after password exemption access is enabled for an instance, if the password in the access parameter is not empty, authentication will fail in the former but will succeed in the latter.

Permission settings

Read-only permission: The account has the permission to read but not modify data.

Read-write permission: The account has the permission to read and write data.

Read-only routing policy

By configuring a read-only routing policy, you can distribute **read requests** from the specified account to the specified (master or replica) node.

If **read-only replica** is not enabled for an instance, the instance will not support routing to replica nodes. This feature can be enabled on the **Node Management** page.

If an instance has an account accessing a replica node, the **read-only replica** feature cannot be disabled. To disable it, you need to delete the account first.

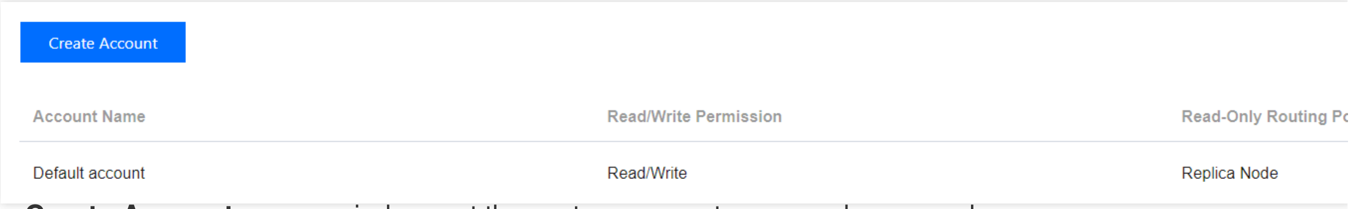
Prerequisites

You have created a TencentDB for Redis instance, and the instance is running.

You have planned the accounts to be defined and their passwords, read-write permissions, and read-only routing policies.

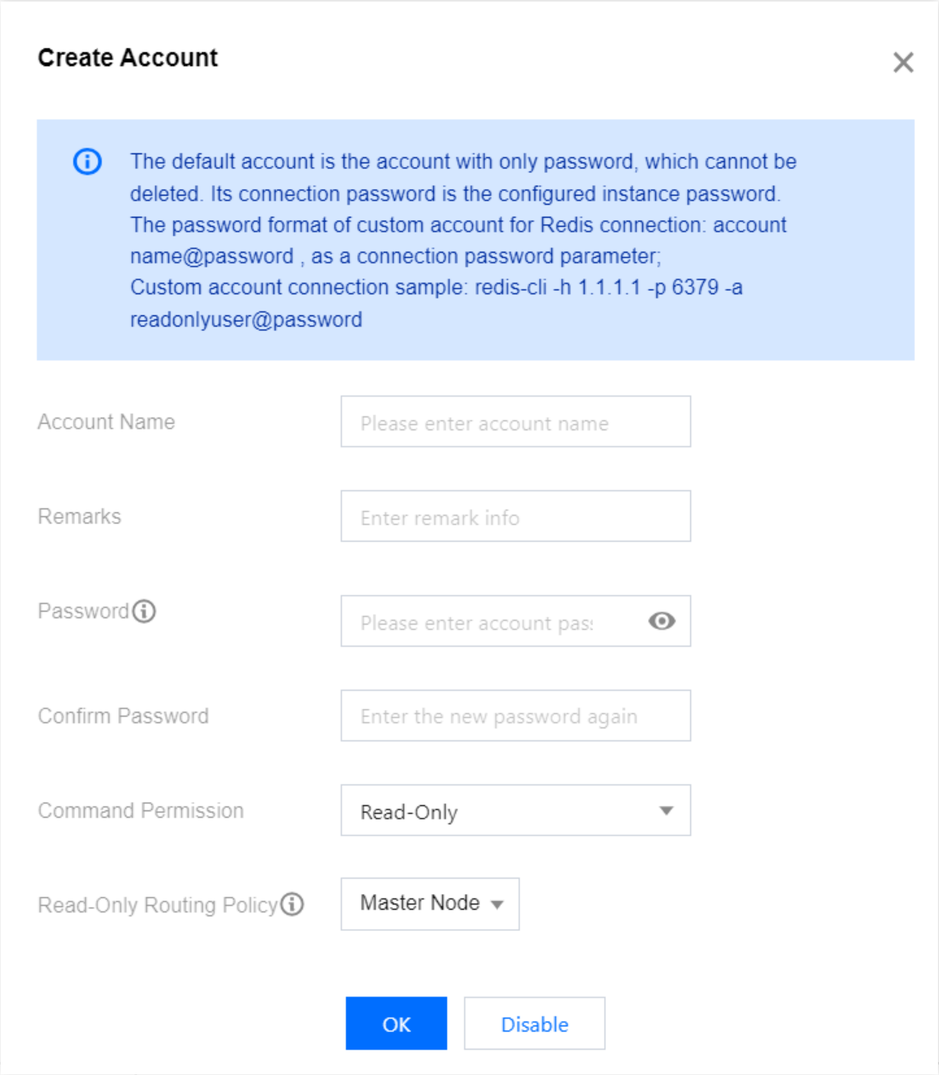
Creating a custom account

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. Select the **Account Management** tab and click **Create Account**.



Account Name	Read/Write Permission	Read-Only Routing Po
Default account	Read/Write	Replica Node

6. In the **Create Account** pop-up window, set the custom account name and password.



Create Account

The default account is the account with only password, which cannot be deleted. Its connection password is the configured instance password.
The password format of custom account for Redis connection: account name@password , as a connection password parameter;
Custom account connection sample: redis-cli -h 1.1.1.1 -p 6379 -a readonlyuser@password

Account Name: Please enter account name

Remarks: Enter remark info

Password: Please enter account pas:

Confirm Password: Enter the new password again

Command Permission: Read-Only

Read-Only Routing Policy: Master Node

Parameter	Description
Account Name	Set the custom account name. Contain only letters, digits, or symbols (-_). Contain up to 32 characters.
Password	Set the custom account password. Contain [8,30] characters. Contain at least two of the following four types: lowercase letters, uppercase letters, digits, and symbols (()`~!@#$%^&*-+=_|{}[]:;<>,.?/). Cannot start with a slash (/).
Confirm Password	Enter the password again.
Command Permission	Select the account permission in the drop-down list, which can be read-only or read-write.
Read-only Routing	Specify to distribute the read requests of the account to the master or replica node. If the read-only replica feature is not enabled, you cannot select replica node here. For detailed directions,

Policy	see Enabling/Disabling Read/Write Separation.

7. Click **OK**. Then, you can use the created custom account to access the database as instructed in Connecting to TencentDB for Redis Instance.

Deleting a custom account

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the account list on the **Account Management** tab, find the target custom account.
6. Click **Delete** in the **Operation** column.
7. In the **Delete Account** pop-up window, confirm the information of the account to be deleted and click **Confirm Deletion**.

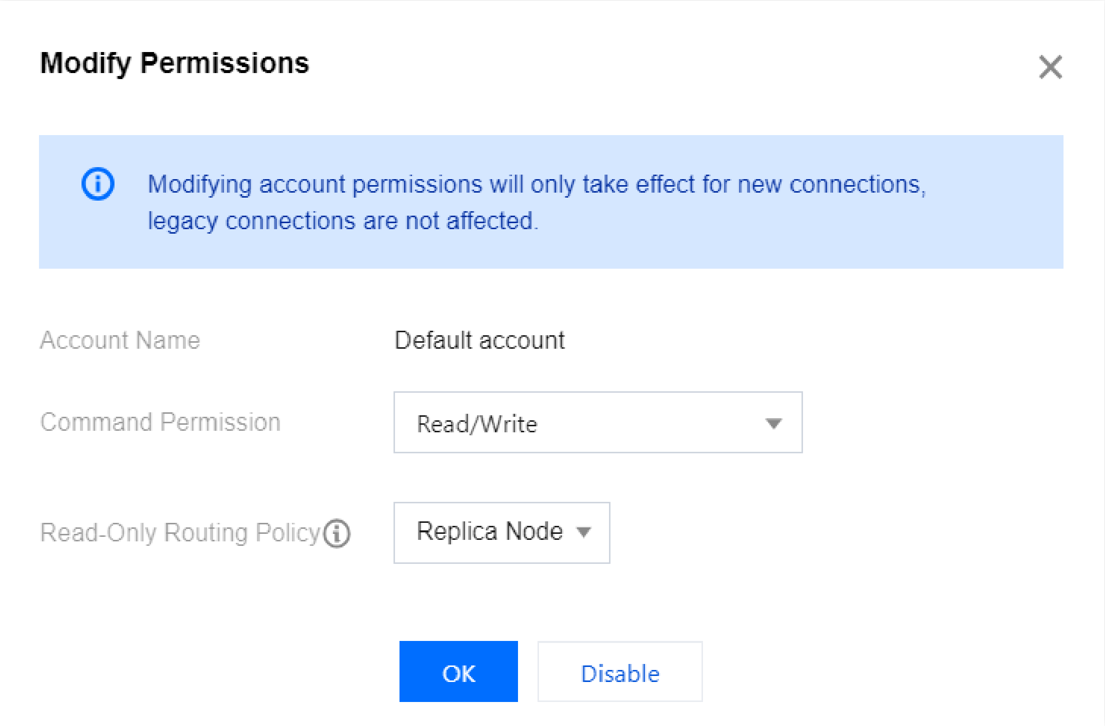
Note:

The default account (i.e., the account with a password only) cannot be deleted.

After the account is deleted, existing connections using the account will not be closed. New connections using the account cannot be verified.

Modifying the account permission

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the account list on the **Account Management** tab, find the target custom account.
6. Click **Modify Permissions** in the **Operation** column.
7. In the **Modify Permission** pop-up window, select the read-write permission of the account in the drop-down list after **Command Permission**, and select the target node for read-only routing in the drop-down list after **Read-only Routing Policy**.



Modify Permissions

Modifying account permissions will only take effect for new connections, legacy connections are not affected.

Account Name: Default account

Command Permission: Read/Write

Read-Only Routing Policy: Replica Node

Note:

The command permission of the default account can only be read-write but not read-only.

If the read-only replica feature is not enabled, you cannot select replica node for the read-only routing policy. For detailed directions, see Enabling/Disabling Read/Write Separation.

8. Click **OK**.

Resetting Password

Last updated：2023-05-23 10:48:12

Overview

If you forgot or want to change your password, you can reset it directly in the console.

Prerequisites

You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.
The database instance is in **Running** status.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID and enter the **Reset Password** window in any of the following ways:

In the **Configuration** section on the **Instance Details** page, click **Reset Password** on the right of **Connection Password**.

In the account list on the **Account Management** page, find the **Default account** or custom account for which to reset the password, and click **Reset Password** in the **Operation** column.

5. In the **Reset Password** pop-up window, enter the **New Password** and **Confirm Password**. The password must meet the following requirements:

It must contain 8-30 characters, preferably 12 or more.

Cannot start with a slash (/)

Must contain characters in at least two of the following types:

Lowercase letters (a-z)

Uppercase letters (A-Z)

Digits (0-9)

`()~!@#$%^&*-+=_|{}[]:;<>,.?/`

6. Click **OK**, and the new password will take effect immediately.

Related APIs

API Name	Description
ResetPassword	Resets password

Parameter Configuration

Setting Instance Parameters

Last updated：2023-03-14 15:49:33

This document describes how to configure instance parameters in the TencentDB for Redis console.

Overview

You can view and modify certain instance parameters and query parameter modification logs in the TencentDB for Redis console.

Notes

To ensure instance stability, only some parameters can be modified in the console. These parameters are displayed on the **Parameter Settings** page.

Editing Parameters

Editing one parameter

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. On the **Instance Details** page, select the **Parameter Settings** tab.
5. Select the target parameter, hover over the parameter value in the **Current Value** column, and click to modify the parameter value.
6. Modify the value within the restrictions stated in the **Acceptable Values** column and click to save the modification. You can click to cancel the operation.

Editing parameters in batches

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.

3. Click the ID of the target instance in the instance list and enter the **Instance Details** page.
4. Select the **Parameter Settings** tab and click **Modify Current Value**.
5. Find the target parameters and modify their values in the **Current Value** column. After confirming that everything is correct, click **OK**.

Importing a parameter template into the current instance

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. Select the **Parameter Settings** tab and click **Import from Template**.
5. In the **Import from Parameter Template** window, select a created parameter template in the drop-down list after **Select Parameter Template**.
6. Click **Import and Overwrite Original Parameters**, and all parameters of the current instance will be configured according to the parameter values in the template.

Exporting the Parameters of an instance

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. Select the **Parameter Settings** tab and click **Export Parameters**.
5. You can see the exported parameter file at the bottom of the page, which can be opened and viewed locally.

Saving Parameters as a Template

You can save the existing parameters of the current instance as a fixed template, so that these parameters can be easily applied to other instances.

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. Select the **Parameter Settings** tab and click **Save as Template**.
5. In the **Save as Parameter Template** window, set **Template Name** and **Template Description** as prompted.
6. Click **Create and Save**.

Viewing Parameter Modification Logs

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. View the parameter modification logs on the **Parameter Settings** > **Modification Log** tab.

Supported Custom Parameters

Parameter	Description	Supported Versions
disable-command-list	Disable commands that have high time complexity or high risks. The disabled commands will not be allowed to run in this instance. To disable multiple commands, separate them by comma, such as `flushdb,keys` .	Redis 2.8, 4.0, 5.0
maxmemory-policy	Select one of the following eviction policies used to evict data when the Redis in-memory cache was used up. volatile-lru: Evict keys that have the TTL set by trying to remove the LRU keys first. allkeys-lru: Evict keys by trying to remove the LRU keys first. volatile-random: Randomly evict keys with the TTL set. allkeys-random: Randomly evict keys. volatile-ttl: Evict keys with the TTL set, and try to evict keys with a shorter TTL first. noeviction: Do not evict keys but return errors when the memory limit was reached. LRU (least recently used) and TTL (time to live) are implemented by randomized and approximation algorithms.	Redis 2.8, 4.0, 5.0
cluster-node-timeout	Set the timeout threshold for a cluster node. If a cluster node remains unreachable longer than the threshold, it will be deemed as a failed node.	Redis 4.0, 5.0
hash-max-ziplist-entries	Hashes that meet both of the following conditions will be encoded as ziplist. The biggest hash entry is smaller than the value (in bytes) of `hash-max-ziplist-value` . The number of hash entries is smaller than the value of `hash-max-ziplist-entries` .	Redis 2.8, 4.0, 5.0
hash-max-ziplist-value	Hashes that meet both of the following conditions will be encoded as ziplist. The biggest hash entry is smaller than the value (in bytes) of `hash-max-ziplist-value` .	Redis 2.8, 4.0, 5.0

	The number of hash entries is smaller than the value of `hash-max-ziplist-entries` .	
proxy-slowlog-log-slower-than	Set the proxy slow log threshold (in milliseconds). In the proxy, queries that are executed longer than the threshold will be logged.	Redis 2.8, 4.0, 5.0
set-max-intset-entries	Sets that meet both of the following conditions will be encoded as intset. All set members are composed of just strings. All set members can be interpreted as base-10 integers within the range of 64-bit signed integers.	Redis 2.8, 4.0, 5.0
slowlog-log-slower-than	Set the slow log threshold (in milliseconds). Queries that are executed longer than the threshold will be logged.	Redis 2.8, 4.0, 5.0
timeout	Set the timeout threshold (in seconds) for connections. Client connections that remain idle longer than the threshold will be closed.	Redis 2.8, 4.0, 5.0
zset-max-ziplist-entries	Sorted sets that meet both of the following conditions will be encoded as ziplist. The biggest sorted set element is smaller than the value (in bytes) of `zset-max-ziplist-value` . The number of sorted set elements is smaller than the value of `zset-max-ziplist-entries` .	Redis 2.8, 4.0, 5.0
zset-max-ziplist-value	Sorted sets that meet both of the following conditions will be encoded as ziplist. The biggest sorted set element is smaller than the value (in bytes) of `zset-max-ziplist-value` . The number of sorted set elements is smaller than the value of `zset-max-ziplist-entries` .	Redis 2.8, 4.0, 5.0
notify-keyspace-events	Specify the type of notifications sent by the server. The value of this parameter is composed of multiple characters listed as follows: Character: Notification type >K: Keyspace events, published with `__keyspace@<db>__` prefix. E: Keyevent events, published with `__keyevent@<db>__` prefix. g: Generic commands (non-type specific) like DEL, EXPIRE, RENAME, etc. $: String commands l: List commands s: Set commands h: Hash commands z: Sorted set commands x: Expired events (events generated every time a key expires) e: Evicted events (events generated when a key is evicted for maxmemory)	Redis 2.8, 4.0, 5.0

	A: Alias for `g$lshzxe` . Enabling keyspace event notifications consumes CPU resources, so this type of notification is disabled by default. To configure the server to send notifications, the parameter value must include `K` or `E` . For example, to subscribe to evicted event notifications of the keyevent events, set the parameter to `Ee` ; to subscribe to all types of notifications, set the parameter to `AKE` .	
list-max-ziplist-entries	Lists that meet both of the following conditions will be encoded as ziplist. The biggest list element is smaller than the value (in bytes) of `list-max-ziplist-value` . The number of list elements is smaller than the value of `list-max-ziplist-entries` .	Redis 2.8
list-max-ziplist-value	Lists that meet both of the following conditions will be encoded as ziplist. The biggest list element is smaller than the value (in bytes) of `list-max-ziplist-value` . The number of list elements is smaller than the value of `list-max-ziplist-entries` .	Redis 2.8

Related APIs

API	Description
ApplyParamsTemplate	Applies a parameter template
CreateParamTemplate	Creates a parameter template
DeleteParamTemplate	Deletes a parameter template
DescribeInstanceParamRecords	Queries the list of parameter modifications
DescribeParamTemplates	Queries the list of parameter templates
ModifyInstanceParams	Modifies the parameters of an instance
ModifyParamTemplate	Modifies a parameter template

Managing Parameter Template

Last updated：2023-05-09 16:03:45

In addition to the various system parameter templates provided by TencentDB for Redis, you can also create custom parameter templates to configure parameters in batches as needed.

You can apply a parameter template to configure and manage the parameters of a database engine. A template is like a container of the values of database engine parameters, which can be applied to one or more database instances.

Parameter templates support the following features:

Support the default parameter templates.

Create custom templates by modifying the default parameters.

Save parameter configurations as templates.

Import parameters from templates to apply to one or more instances.

Notes

If the parameters in the template are updated, the instance parameters are not updated unless they are manually re-applied to the instances.

You can apply the parameter changes to single or multiple instances by importing a template.

Creating a custom parameter template

To use your own database parameter template, you can create a parameter template, modify the parameter values, and apply the template to instances.

1. Log in to the TencentDB for Redis console, select **Parameter Templates** on the left sidebar, and click **Create Template**.

2. In the pop-up window, configure the following parameters and click **Create and Set Parameters**.

Template Name: Enter a unique template name.

Database Version: Select a database version.

Template Description: Enter a brief description of the parameter template.

3. On the displayed parameter configuration page, make sure that all parameter values are correct and the parameter template is created successfully.

Default template

This page displays the default templates applicable to Redis 2.8 standard architecture, Redis 4.0 standard architecture, Redis 4.0 cluster architecture, Redis 5.0 standard architecture, and Redis 5.0 cluster architecture.

Note:

When you purchase an instance, the system automatically matches the appropriate default template based on the selected compatible version and architecture. For detailed directions, see Creating Redis Instance.

Notes

On the **Default Template** page, you can click **View Details** to view a template, but on the displayed details page, you cannot do any operation.

Applying a parameter template to instances

Applying to existing instances

Applying a parameter template to one or more instances

1. Log in to the TencentDB for Redis console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, locate the desired template, and click **Apply to Instance** in the **Operation** column. Or, click **View Details**, enter the template details page, and click **Apply to Instance**.
3. On the displayed page, specify the execution mode and instances, make sure that all parameter values are correct, and click **Submit**.

Redis Instance: Select one or more instances that need to apply the parameter template in the specified region.

Parameter Comparison: View the changed parameter values of the selected instance.

Notes

Before applying a parameter template to multiple instances, make sure that the instances do support those parameters.

Importing a parameter template into the current instance

1. Log in to the TencentDB for Redis console.
2. Click the instance ID on the instance list page to enter the instance management page.
3. On the instance management page, select the **Parameter Settings** tab.
4. On the **Modifiable Parameters** tab, click **Import from Template**.
5. In the **Import from Parameter Template** window, select a created parameter template in the drop-down list after **Select Parameter Template**.
6. Click **Import and Overwrite Original Parameters**, and all parameters of the current instance will be configured according to the parameter values in the template.

Applying to new instances

On the instance purchase page, select a default or custom parameter template for the new instances.

Click the selection box next to **Parameter Template** and a drop-down list with a search box will pop up. The list shows the names and IDs of all existing parameter templates.

Enter the parameter template name or ID in the search box to search for a template.

Copying a parameter template

To include most of the custom parameters and values of an existing parameter template in a new template, you can copy the existing template and modify it.

Option 1

1. Log in to the TencentDB for Redis console.
2. Select **Parameter Templates** on the left sidebar, click the parameter template name or **View Details** in the **Operation** column in the template list, and enter the parameter template details page.
3. Click **Save as Template**.
4. In the pop-up window, specify the following configurations:

Template Name: Enter a unique template name.

Template Description: Enter a brief description of the parameter template.

5. After confirming that everything is correct, click **Save**.

Option 2

1. Log in to the TencentDB for Redis console, select **Instance List** on the left sidebar, click an instance ID, and enter the instance management page.
2. On the **Parameter Settings** tab, click **Save as Template**.
3. In the pop-up window, specify the following configurations:

Template Name: Enter a unique template name.

Template Description: Enter a brief description of the parameter template.

4. After confirming that everything is correct, click **Create and Save**.

Modifying parameter values in a parameter template

1. Log in to the TencentDB for Redis console, select **Parameter Templates** on the left sidebar, click the parameter template name or **View Details** in the **Operation** column in the template list, and enter the parameter template details page.
2. Click **Batch Modify Parameters** or

✎ in the **Current Value** column to modify parameter values.

Notes

Default templates do not support the **Batch Modify Parameters** feature, and the **Current Value** column cannot be modified.

Deleting a Parameter Template

If a parameter template is created redundantly or no longer needed, it can be easily deleted.

1. Log in to the TencentDB for Redis console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, locate the target template and click **Delete** in the **Operation** column.
3. Click **OK** in the pop-up window.

Related APIs

API	Description
ApplyParamsTemplate	Applies a parameter template
CreateParamTemplate	Creates a parameter template
DeleteParamTemplate	Deletes a parameter template
DescribeInstanceParamRecords	Queries the list of parameter modifications
DescribeParamTemplates	Queries the list of parameter templates
ModifyInstanceParams	Modifies instance parameters
ModifyParamTemplate	Modifies a parameter template

Disabling Commands

Last updated：2023-02-07 16:08:02

Overview

TencentDB for Redis supports disabling some commands that may cause service instability or accidentally delete data.

You can configure the `disable-command-list` parameter to disable such commands. If this parameter is not displayed in your console, submit a ticket to upgrade the minor kernel version. A momentary disconnection will occur during the version upgrade, and you can reconnect after the upgrade is completed.

Directions

Disabling a command

1. Log in to the TencentDB for Redis console.
2. Above the instance list on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. On the **Instance Details** page, select the **Parameter Settings** tab.
6. On the **Parameter Settings** tab, click the **Modifiable Parameters** tab, find `disable-command-list` in the parameter list, and configure the list of disabled commands.

Notes

Commands that can be disabled include `flushall` , `flushdb` , `keys` , `hgetall` , `eval` , `evalsha` , and `script` , subject to the acceptable values in the actual parameter list.

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

Enabling a command

On the **Modifiable Parameters** tab, remove a command from the list of disabled commands in **Current Value** to enable it.

Parameter modification log

View the parameter modification history on the **Modification Log** tab.

Applying Parameter Templates

Last updated：2023-02-24 10:32:02

Besides the system parameter templates provided by TencentDB for Redis, you can create custom parameter templates to configure parameters in batches as needed.

You can apply a parameter template to configure and manage the parameters of a database engine. A template is like a container of the values of database engine parameters, which can be applied to one or more database instances.

Parameter templates support the following features:

Support the default parameter templates.

Create custom templates by modifying the default parameters.

Save parameter configurations as templates.

Import parameters from templates to apply to one or more instances.

Notes

If the parameters in the template are updated, the instance parameters are not updated unless they are manually re-applied to the instances.

You can apply the parameter changes to single or multiple instances by importing a template.

Creating a custom parameter template

To use your own database parameter template, you can create a parameter template, modify the parameter values, and apply the template to instances.

1. Log in to the TencentDB for Redis console, select **Parameter Templates** on the left sidebar, and click **Create Template**.

2. In the pop-up window, configure the following parameters and click **Create and Set Parameters**.

Template Name: Enter a unique template name.

Database Version: Select a database version.

Template Description: Enter a brief description of the parameter template.

3. On the displayed parameter configuration page, make sure that all parameter values are correct and the parameter template is created successfully.

Default Template

This page displays the default templates applicable to Redis 2.8 standard architecture, Redis 4.0 standard architecture, Redis 4.0 cluster architecture, Redis 5.0 standard architecture, and Redis 5.0 cluster architecture.

Notes

On the **Default Template** page, you can click **View Details** to view a template, but on the displayed details page, you cannot do any operation.

Applying a Parameter Template to Instances

During instance use

Applying a parameter template to one or more instances

1. Log in to the TencentDB for Redis console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, locate the desired template, and click **Apply to Instance** in the **Operation** column. Or, click **View Details**, enter the template details page, and click **Apply to Instance**.
3. On the displayed page, specify the execution mode and instances, make sure that all parameter values are correct, and click **Submit**.

Redis Instance: Select one or more instances that need to apply the parameter template in the specified region.

Parameter Comparison: View the changed parameter values of the selected instance.

Notes

Before applying a parameter template to multiple instances, make sure that the instances do support those parameters.

Importing a parameter template into the current instance

1. Log in to the TencentDB for Redis console.
2. Click the instance ID on the instance list page to enter the instance management page.
3. On the instance management page, select the **Parameter Settings** tab.
4. On the **Modifiable Parameters** tab, click **Import from Template**.
5. In the **Import from Parameter Template** window, select a created parameter template in the drop-down list after **Select Parameter Template**.
6. Click **Import and Overwrite Original Parameters**, and all parameters of the current instance will be configured according to the parameter values in the template.

During instance purchase

On the instance purchase page, select a default or custom parameter template for the new instances.

Click the selection box next to **Parameter Template** and a drop-down list with a search box will pop up. The list shows the names and IDs of all existing parameter templates.

Enter the parameter template name or ID in the search box to search for a template.

Copying a Parameter Template

To include most of the custom parameters and values of an existing parameter template in a new template, you can copy the existing template and modify it.

Option 1

1. Log in to the TencentDB for Redis console.
2. Select **Parameter Templates** on the left sidebar, click the parameter template name or **View Details** in the **Operation** column in the template list, and enter the parameter template details page.
3. Click **Save as Template**.
4. In the pop-up window, configure the following parameters:

Template Name: Enter a unique template name.

Template Description: Enter a brief description of the parameter template.

5. After confirming that everything is correct, click **Save**.

Option 2

1. Log in to the TencentDB for Redis console, select **Instance List** on the left sidebar, click an instance ID, and enter the instance management page.
2. On the **Parameter Settings** tab, click **Save as Template**.
3. In the pop-up window, configure the following parameters:

Template Name: Enter a unique template name.

Template Description: Enter a brief description of the parameter template.

4. After confirming that everything is correct, click **Create and Save**.

Modifying Parameter Values in a Parameter Template

1. Log in to the TencentDB for Redis console, select **Parameter Templates** on the left sidebar, click the parameter template name or **View Details** in the **Operation** column in the template list, and enter the parameter template details page.
2. Click **Batch Modify Parameters** or in the **Current Value** column to modify parameter values.

Notes

Default templates do not support the **Batch Modify Parameters** feature, and the **Current Value** column cannot be modified.

Deleting a Parameter Template

If a parameter template is created redundantly or no longer needed, it can be easily deleted.

1. Log in to the TencentDB for Redis console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, locate the target template and click **Delete** in the **Operation** column.
3. Click **OK** in the pop-up window.

Related APIs

API	Description
ApplyParamsTemplate	Applies a parameter template.
CreateParamTemplate	Creates a parameter template.
DeleteParamTemplate	Deletes a parameter template.
DescribeInstanceParamRecords	Queries the list of parameter modifications.
DescribeParamTemplates	Queries the list of parameter templates.
ModifyInstanceParams	Modifies instance parameters.
ModifyParamTemplate	Modifies a parameter template.

Slow Log

Last updated：2023-11-15 17:57:01

Overview

A slow query is defined as a query statement that takes more time than the specified value, and the statement is called a slow query statement. The slow log analysis feature of TencentDB for Redis offers slow logs in two dimensions: **instance** and **proxy**.

In the instance (Redis database instance) dimension, you can clearly view the CPU utilization, number of slow queries, consumed time statistics by log segment, and information of the entire slow log list.

In the proxy (middleware cluster node) dimension, you can view the proxy's slow log statistics, consumed time statistics by segment, and details of the slow log list.

Slow log analysis is often used as an important basis for database performance optimization. DBbrain is a cloud database autonomy service provided by Tencent Cloud for database performance optimization, security, and management, and the slow logs of TencentDB for Redis are connected to it for statistical analysis.

Billing Details

The slow log analysis feature of DBbrain is currently in beta test and free of charge. For more information, see Purchase Guide.

Note

Redis Slow Log operations with an execution time of 1 ms, 5 ms, 10 ms, 20 ms, 40 ms, and 50 ms.

Proxy Slow Log operations with an execution time of 10 ms, 20 ms, 40 ms, and 50 ms.

The slow logs are retained for 15 days.

A default account can perform up to 15,000 queries per second, while a custom account can perform up to 20 queries per second.

If the query is slow, you can narrow down the query time period.

Prerequisites

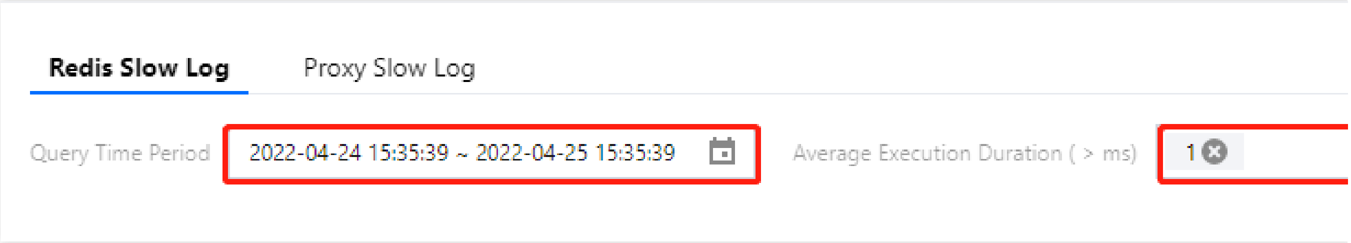
You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.

The database instance is in **Running** status.

Directions

Querying slow logs

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. Click the **Slow Log** tab to enter the **Redis Slow Log** page. To query the slow logs generated by the proxy, select the **Proxy Slow Log** tab.
6. Select the time period after **Query Time Period**, select the threshold for command request duration from the drop-down list after **Average Execution Duration (> ms), click** Query, and slow logs within the time period will be displayed.

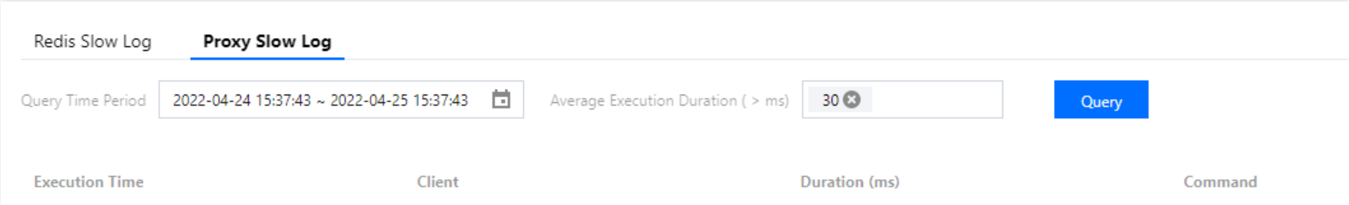


Query Time Period 2022-04-24 15:35:39 ~ 2022-04-25 15:35:39
Average Execution Duration (> ms) 1

Related Operations

Exporting slow log

1. Click **Export** in the top-right corner of the **Redis Slow Log** or **Proxy Slow Log** page.



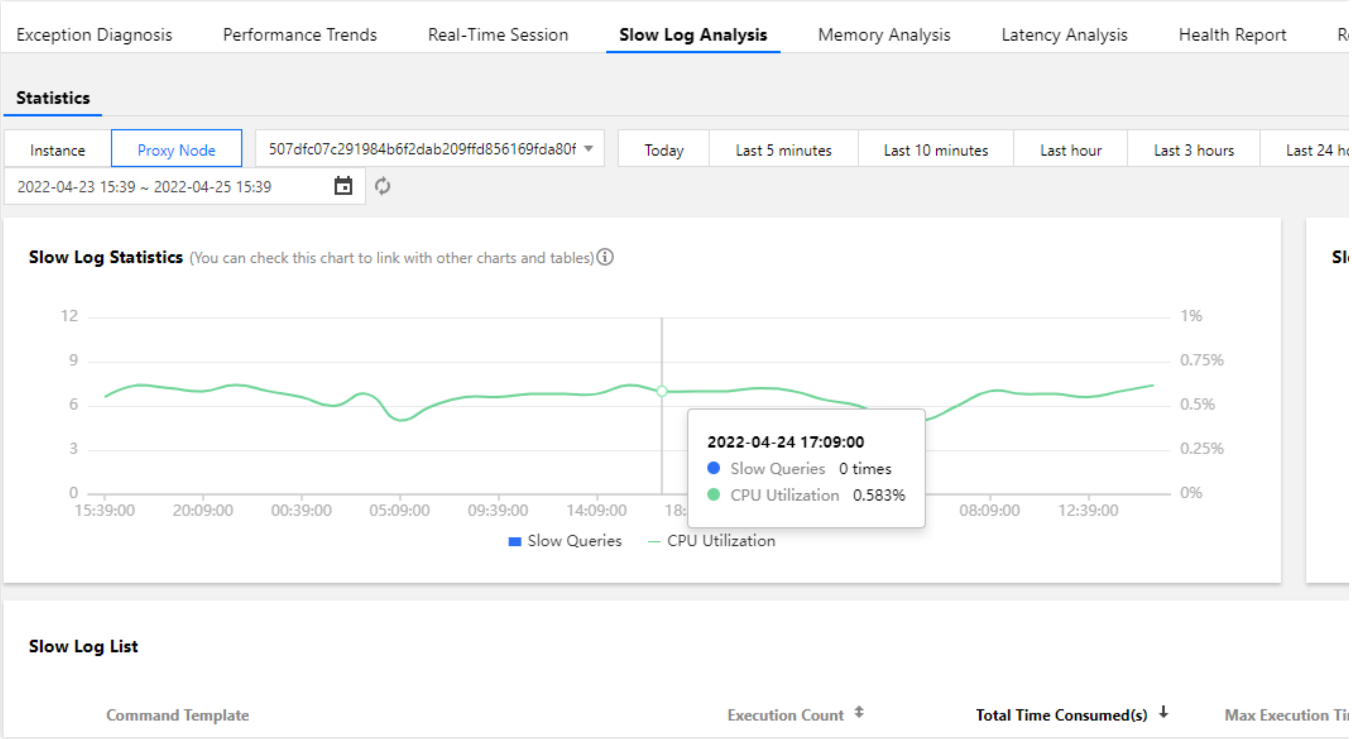
Query Time Period 2022-04-24 15:37:43 ~ 2022-04-25 15:37:43
Average Execution Duration (> ms) 30

Execution Time	Client	Duration (ms)	Command

2. In the pop-up window, view the limit on the number of slow logs that can be exported at a time, and click **Export**.
3. Open the exported slow logs in Excel.

Analyzing slow log

1. Click **Slow Log Analysis** in the top-right corner of the **Redis Slow Log** or **Proxy Slow Log** tab.
2. On **DBbrain** > **Slow Log Analysis** > **Performance Optimization**, select the query time period, and view the slow log information at the instance level or proxy level as needed. For more analysis, see Slow Log Analysis.



Slow Log Statistics (You can check this chart to link with other charts and tables)

Related APIs

API Name	Description
DescribeSlowLog	Queries the slow logs of an instance
DescribeProxySlowLog	Queries the slow queries of proxy

Network and Security

Configuring Network

Last updated：2023-03-14 15:44:21

Network overview

Tencent Cloud offers VPC (recommended) and classic network environments. A VPC is a logically isolated network space that can be customized in Tencent Cloud. Similar to the traditional network run in an IDC, a VPC is where your Tencent Cloud service resources are managed, such as Cloud Virtual Machine, Cloud Load Balancer, and TencentDB for MySQL.

The difference between classic network and VPC is as follows:

The classic network is a public network resource pool for all Tencent Cloud users. The private IPs of all CVMs are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses.

A VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies, making it more suitable for use cases requiring custom configurations.

A subnet is a network space in a VPC, which carries all the cloud resource deployments. A VPC has at least one subnet. A subnet will be created together with the VPC. You can also create more subnets in a VPC that suit your specific requirements to deploy different businesses.

A subnet is AZ-specific. You can create subnets in different AZs in the same VPC, which communicate with each other over the private network by default.

Notes

You can switch from classic network to VPC but not vice versa.

You can switch from one VPC to another or between subnets in the same VPC.

After the network switch, the new IP address will take effect immediately, and all connections to the original IP address will be closed. The original IP address will be retained for up to 15 days.

Notes

To ensure the service availability and keep your businesses uninterrupted during network switch, update the IP address promptly as needed and be cautious about releasing the old IP address.

Prerequisites

You have created a database instance and configured the classic network or VPC as instructed in Creating TencentDB for Redis Instance.

You have created the target VPC as instructed in Creating VPC.

The database instance is in **Running** status, with no ongoing tasks.

Changing the network

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Network Info** section on the **Instance Details** page, you can see the network and private IP of the current TencentDB for Redis instance. Click **Change Network** next to **Network**.

In this way, you can switch from the classic network to a VPC or from the current VPC to another VPC.

6. In the **Change Network** window, configure the new network information and click **OK**.

Network: Select a VPC and a subnet in the drop-down lists next to **Network** respectively.

New IP: Select **Auto Assign** or **Designate Address**. If you select the latter, then enter a custom IP address in the input box.

Old IP: Set the release time of the old IP. You can select **Release Now**, **Release after 1 day**, **Release after 2 days**, **Release after 3 days**, **Release after 7 days**, or **Release after 15 days** as needed to ensure the business continuity during the network change.

Changing the subnet

Notes

The subnet change feature is available only when the **Network** is **VPC**.

1. Log in to the TencentDB for Redis console.
2. Above the instance list, select the region.
3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. In the **Network Info** section, you can see the network and private IP of the current TencentDB for Redis instance. Click **Change Subnet**.
6. In the pop-up window, select the target subnet and click **OK**.

Related APIs

API	Description
ModifyNetworkConfig	Modifies the network configuration of instance

Configuring Security Group

Last updated：2023-05-23 10:26:23

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, see Security Groups Overview. You can bind a security group directly during instance purchase or bind one in the console after instance purchase.

Note

TencentDB for Redis security groups currently only support network access control for VPCs and public networks but not the classic network.

As TencentDB doesn't have any active outbound traffic, outbound rules don't apply to it.

TencentDB for Redis security groups support master instances, read-only instances, and disaster recovery instances.

Configuring Security Groups 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 above the instance list on the right, and click **Create**.
3. In the pop-up window, set the following configuration items, confirm that everything is correct, and click **OK**.

Template: Select a security group template in the drop-down list.

Open all ports: All ports are opened to the public and private networks. This may present security issues. Security group rules are added by default. You can click a security group template below to view its **Outbound Rules* and** Inbound Rules**.

- **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. Security group rules are added by default.

Custom: You can create a security group and then add custom rules.

Name: Custom name of the security group.

Project: Select a project for easier management. By default, **Default Project** is selected.

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

Advanced Configuration: You can add tags for the security group.

4. If you select **Custom** for **Template**, click **Set Now** in the **Note** window and perform the following steps.

Step 2. Set inbound rules in the security group

1. On the **Inbound Rule** tab of the **Security Group Rules** page, click **Add Rules**.

2. In the **Add Inbound Rules** window, set the rules.

Type: Select **Custom** as the default type.

Source: Set the source for database access, i.e., the inbound source, in the following formats:

Source Format	Format description
CIDR notation	A single IPv4 address or an IPv4 range is represented 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 is represented 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.
Security group ID	Reference a security group ID to match the IP address of the server associated with the security group.
Parameter template	Reference an IP address object or IP address group object in a parameter template.

Protocol Port: Enter the protocol type and port for the client to access TencentDB for Redis. You can view the port information in the **Private IPv4 Address** in the **Network Info** section on the Instance Details page. The default port is 6379. If the access protocol is TCP, you can enter `TCP:6379` .

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

Allow: Access requests of this port are allowed.

Reject: Data packets will be discarded without any response.

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

3. Click **Complete**.

Step 3. Configure the security group for a database instance

Note

When purchasing an instance, you can directly select the security group to be associated with the instance from the **Security Group** drop-down list on the purchase page. For more information, see Creating TencentDB for Redis Instance.

After purchase, you can change or add a security group for an instance based on the following steps.

1. Log in to the TencentDB for Redis console.

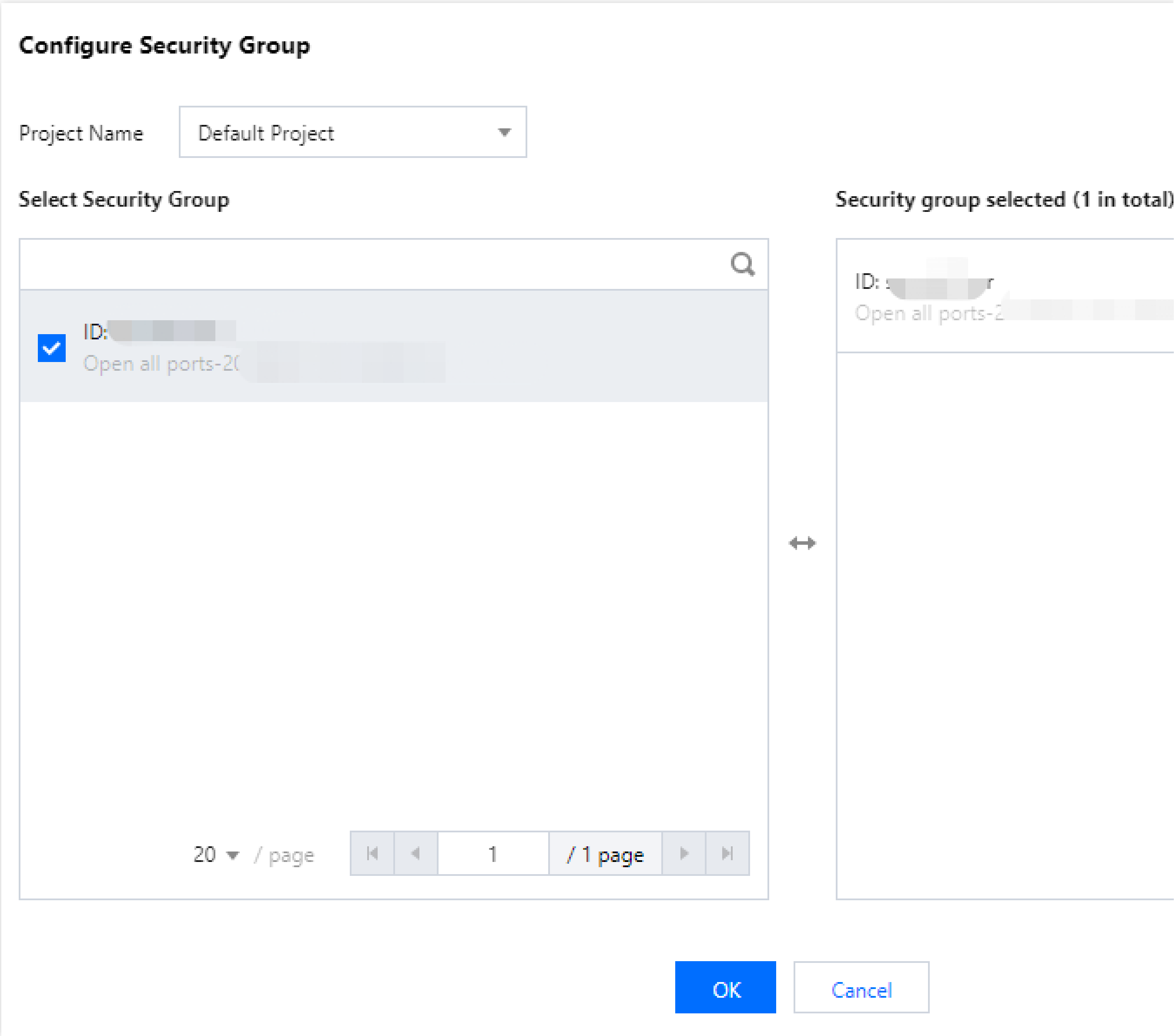
2. Above the **Instance List** on the right, select the region.

3. In the instance list, find the target instance.

4. Click the instance ID to enter the instance management page.

5. On the **Security Group** tab, click **Configure Security Group**.

6. In the **Configure Security Group** pop-up window, select a created security group. You can filter security group by project name.



Configure Security Group

Project Name: Default Project

Select Security Group

☑ ID:
Open all ports-2(

Security group selected (1 in total)

ID:
Open all ports-2

7. Click **OK**.

In the **Associated Security Group** section, you can view the list of security groups associated with the database instance.

You can click a security group ID to enter the **Security Group Rules** page and edit the rules as instructed in Modifying a Security Group Rule. For more operations, see Viewing a Security Group Rule.

In the **Priority** column, you can view the priorities of security groups. When two or more security groups are selected, click **Edit** above the security group list, click

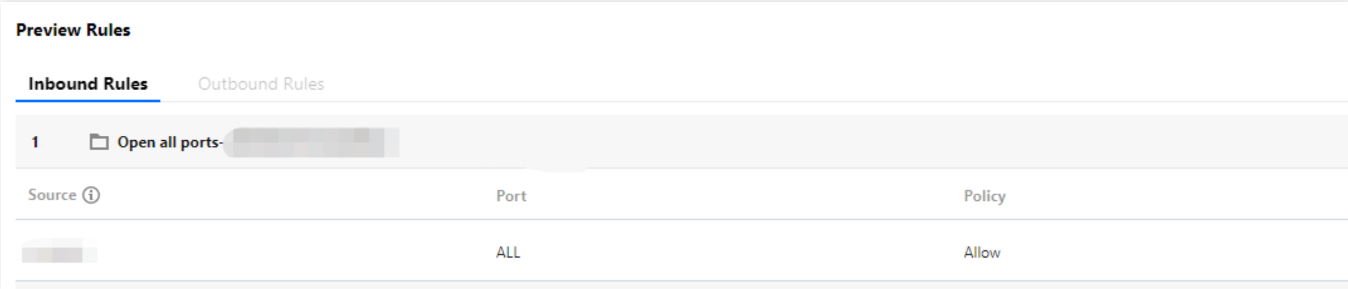
or ↑

↓ to adjust the priorities of security groups.

Configure two or more security groups and click

⊗ to delete the bound security groups.

On the **Preview Rules** page, you can view the inbound source information of the security group on the **Inbound Rules** tab.



Preview Rules

Inbound Rules Outbound Rules

1	Open all ports-[redacted]	
Source ⓘ	Port	Policy
[redacted]	ALL	Allow

More operations

For more security group operations, see Viewing a Security Group.

For more security group rule operations, see Viewing a Security Group Rule.

For security group APIs, see DeleteSecurityGroup.

Related APIs

API Name	Description
DescribeProjectSecurityGroup	Queries the security group information of a project
DescribeInstanceSecurityGroup	Queries the security group information of an instance
ModifyDBInstanceSecurityGroups	Modifies the security groups bound to an instance
AssociateSecurityGroups	Binds a security group
DisassociateSecurityGroups	Unbinds a security group from instances in batches

Configuring Public Network Address

Last updated：2023-03-14 15:48:49

This document describes how to enable/disable the public network address in the TencentDB for Redis console. You can use the system-assigned domain name and port to Accessing Database over Public Network, making it easier for you to test, manage, use, and develop the database on a daily basis.

Notes

The instance service downtime caused by public network errors won't be counted into the "Single Instance Service Downtime" in TencentDB for Redis Service Level Agreement (SLA).

Public network access may expose your instances to security threats, and service availability is not guaranteed by SLA. Therefore, we recommend that you access Redis over the public network only when testing, managing, or assisting in managing databases. In the production environment, access Redis over the private network.

Notes

When it is enabled, you can use the system-assigned domain name and port to access TencentDB for Redis via public network. It takes about 5 minutes to take effect.

After the public network access is enabled, it will be controlled by the security group policy. You should configure the database access source in the security group's inbound rules and open the protocol ports (both the private network port (6379 by default) and public network port) as instructed in Configuring Security Group.

Use Limits

Only instances in VPCs can enable the public network address. If an instance is in the classic network, switch it to VPC first before enabling public network access.

Currently, the public network address can be enabled for the instances in Chengdu, Beijing, Shanghai, and Guangzhou regions. To use it in other regions, you can access the TencentDB for Redis instance through iptables-based forwarding. For more information, see iptables Forwarding.

Prerequisites

To enable public network access, you need to disable password-free access.

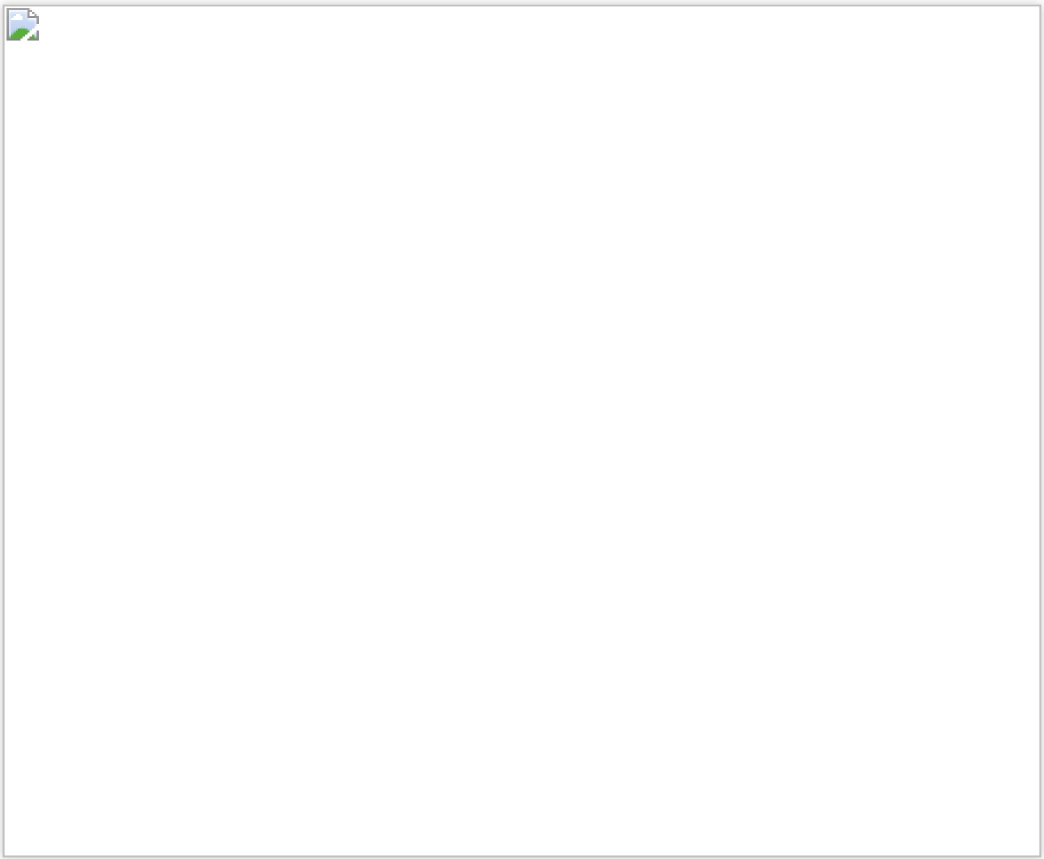
Enabling the Public Network Address

1. Log in to the TencentDB for Redis console, click an instance ID in the instance list, and enter the instance details page.
2. Click **Enable** next to **Public Network Address** in the **Network Info** section.

Notes

After the public network access is enabled, it will be controlled by the security group policy. You should configure the database access source in the security group's inbound rules and open the protocol ports (both the private network port (6379 by default) and public network port) as instructed in Configuring Security Group.

3. In the pop-up window, confirm that everything is correct and click **OK**.



4. Return to the instance details page, where you can see the instance in the **Enabling public network** status. If the status stays the same for a long time, refresh the page.
5. If **Public Network Address** shows an address comprising a domain name and port, the address is enabled successfully. Now you can use it to access Redis over the public network.

Disabling the Public Network Address

1. Log in to the TencentDB for Redis console, click an instance ID in the instance list, and enter the instance details page.
2. Click **Disable** next to **Public Network Address** in the **Network Info** section.

3. In the pop-up window, confirm that everything is correct and click **OK**.

4. Return to the instance details page, where you can see the instance in the **Disabling public network** status and **Public Network Address** display nothing.

Related APIs

API	Description
AllocateWanAddress	Enables public network access
ReleaseWanAddress	Disables public network access

iptables Forwarding

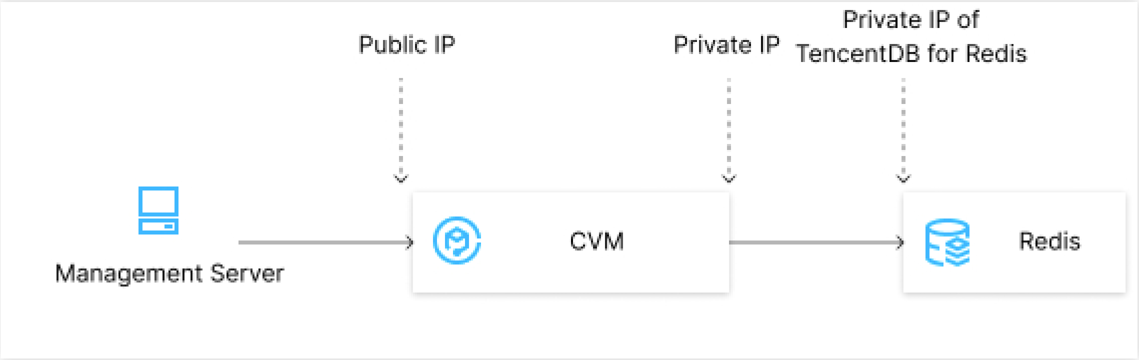
Last updated：2022-10-31 10:29:37

Overview

TencentDB for Redis supports public network access in Chengdu, Beijing, Shanghai, and Guangzhou regions. To use public network access in other regions, you can use a CVM instance with a public IP for port forwarding to access TencentDB for Redis over the public network.

Note:

Because iptables-based forwarding may be unstable, we recommend that you do not access instances over the public network in the production environment.



Directions

1. Log in to a CVM instance and enable the IP forwarding feature. For more information, see Logging In to Linux Instance (Web Shell).

Note:

The CVM and TencentDB instances must be under the same account and in the same VPC in the same region, or both in the classic network.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Configure the forwarding rule. The sample code below forwards access requests originally to `26.xx.x.2:10001` (CVM instance public address with a custom port as desired) to the TencentDB for Redis instance with the private address `10.0.0.5:6379` :

```
iptables -t nat -A PREROUTING -p tcp --dport 10001 -j DNAT --to-destination 10.0.0.
iptables -t nat -A POSTROUTING -d 10.0.0.5 -p tcp --dport 6379 -j MASQUERADE
```

3. Configure the security group to open the public port of the CVM instance. We recommend that you configure a security group rule to allow only the source which needs to connect to the Redis instance. For more information, see Adding Security Group Rules.

4. To connect to the Redis instance in the private network using a public network address (`26.xx.xx.2:10001` in the sample code), you can use the same command as the private network connection command. For more information, see Connecting to TencentDB for Redis Instances > Connecting via Client Tool.

5. After connecting to the TencentDB for Redis instance, run the `info` command. If the database information is returned, the connection is successful.

Accessing Database over Public Network

Last updated：2023-05-23 10:53:44

Connecting to Database from Local Windows at Public Network Address

redis-cli is the native command line tool offered by Redis. You can install it on your local device and use it to connect to TencentDB for Redis at a public network address for data management.

Connect via redis-cli

1. Download redis-cli and decompress the package to the installation directory, such as `D:\\Temp\\Redis-x64-3.2.100` .
2. On the local device, press **Windows + R** to open the **Run** window, enter **cmd**, and click **OK** to open Windows Command Prompt.
3. Run the following command to enter the installation directory of redis-cli.

```
cd /d <path>
```

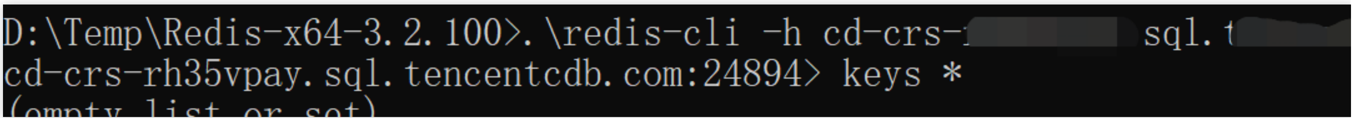
Here, `path` is the installation directory of redis-cli. For example, run `cd /d D:\\Temp\\Redis-x64-3.2.100`.

4. Run the following command to access the database:

```
redis-cli -h <hostname> -p <port> -a <password>
```

Here, `hostname` is the public network address of the database instance, `port` is the port of the public network address, and `password` is the default password of the instance account. If you use a custom account for connection, use `account name@password` as the password parameter for authentication.

An execution example is as shown below:



```
D:\Temp\Redis-x64-3.2.100>.\redis-cli -h cd-crs-[redacted] sql.t[redacted]
cd-crs-rh35vpay.sql.tencentcdb.com:24894> keys *
(empty list or set)
```

Connect via Redis client

Download the Redis client for Windows, configure the following parameters, and click **Test Connection** to connect to the database instance.

Parameter	Description
Name	The name of the connection to the database instance.
Address	Public network address and port number of the database instance.
Verification	Database instance connection password. If you use the default account, directly enter the instance password. If you use a custom account, enter `account name@password` for authentication.

Connecting to Database from Local Linux at Public Network Address with redis-cli

1. Download the latest stable version of the source code package. 6.2.6 is used as an example here.

```
wget https://download.redis.io/releases/redis-6.2.6.tar.gz
```

2. Run the following command to decompress the source code package:

```
tar -zxvf redis-6.2.6.tar.gz
```

3. Enter the source code directory and compile source code files.

```
cd redis-6.2.6/
```

4. Wait patiently as the compilation time varies by server configuration.

```
make
```

5. Run the following command to connect to the database at the public network address. The following is the default path of redis-cli.

```
src/redis-cli -h <hostname> -p <port> -a <password>
```

Here, `hostname` is the public network address of the database instance, `port` is the port of the public network address, and `password` is the default password of the instance account. If you use a custom account for connection, use `account name@password` as the password parameter for authentication.

Bandwidth Adjustment

Last updated：2023-05-23 10:58:55

Overview

The network bandwidth required varies depending on the instance specifications. If the traffic exceeds the bandwidth cap, it may cause congestion and affect the service performance. For example, to handle business traffic peaks during flash sales, or to eliminate the impact of the bandwidth limit when a lot of big key reads and writes occur temporarily, you can quickly increase the instance bandwidth to avoid affecting the business.

Billing Details

Increasing the bandwidth is free of charge currently but will be billed in the future.

Concepts

Standard bandwidth: It is the bandwidth per (master or replica) node in the instance.
Read-only replica bandwidth: Each read-only replica has the same bandwidth as that of the master.
Additional bandwidth: If the standard bandwidth cannot meet your needs, you can add additional bandwidth.

Note

Increasing the bandwidth will not affect your business, but reducing the bandwidth may cause throttling of the traffic that exceeds the bandwidth.

Prerequisites

The database instance is on v4.0 or later.
The database instance is in **Running** status.
The bandwidth of the database instance isn't suitable for the current business.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. Open the **Adjust Bandwidth** pop-up window in any of the following ways:

In the **Operation** column of the target instance, select **Configure** > **Adjust Bandwidth**.

Click the instance ID and click **Adjust Bandwidth** after **Max Network Throughput** in the **Network Info** section on the **Instance Details** page.

5. In the **Adjust Bandwidth** pop-up window, select the desired additional bandwidth on the slider bar after **Additional Bandwidth**.

Parameter	Description
Instance Name	The instance name.
Instance Specifications	The instance specification: Shard quantity, memory capacity, and replica quantity.
Read-Only Replica	The read-only replica status.
Standard Bandwidth	It is the bandwidth per (master or replica) node in the instance.
Additional Bandwidth	Select the additional bandwidth on the slider bar.
Total Instance Bandwidth	If read-only replica is enabled, the total instance bandwidth = additional bandwidth *shard quantity + standard bandwidth* shard quantity * Max ([read-only replica quantity, 1]). The shard quantity in the standard architecture is 1. If read-only replica is not enabled, the total instance bandwidth = additional bandwidth *shard quantity + standard bandwidth* shard quantity. The shard quantity in the standard architecture is 1.
Fees	Free of charge currently.

6. After confirming that the total bandwidth meets your expectations, click **Confirm**.
7. **Instance Status** will change to **Processing**. Wait for it to change to **Running**. Then, you can see that the **Max Network Throughput** is the updated total bandwidth in the **Network Info** section on the **Instance Details** page.

Related APIs

API Name	Description
ModifyNetworkConfig	Modifies the network configuration of an instance to change its bandwidth

SSL Encryption

Last updated：2023-05-23 10:39:34

Overview

Secure Sockets Layer (SSL) authentication is a process that authenticates the connection from the user client to the TencentDB server. After SSL encryption is enabled, you can get a CA certificate and upload it to the server. Then, when the client accesses the database, the SSL protocol will be activated to establish an SSL secure channel between the client and the server. This implements encrypted data transfer, prevents data from being intercepted, tampered with, and eavesdropped during transfer, and ultimately ensures the data security for both the client and the server.

Note

The SSL encryption is being gradually released in regions. To try it out, [submit a ticket] (https://console.tencentcloud.com/workorder/category

Billing Details

SSL encryption is free of charge.

Precautions

Enabling SSL encryption ensures the security of data access and transfer but may slightly affect the instance performance. We recommend you enable it only when encryption is required.

When SSL encryption is enabled, password-free access cannot be supported.

After the SSL encryption feature is disabled, clients using encrypted connections will not be able to connect properly.

The SSL certificate is valid for 20 years.

Version and Architecture Requirements

Version Description

New instances: If the compatible version is 4.0, 5.0, or 6.2, SSL encryption can be enabled directly. To use it on v6.0, submit a ticket for application.

Existing instances:

If the compatible version is 2.8, SSL encryption can be enabled after the version is upgraded to version 4.0, 5.0, or 6.2. For more information, see Upgrading Instance Version.

If the compatible version is 4.0, 5.0, or 6.0, the feature can be enabled after the proxy version is upgraded to 5.6.0. For more information, see Upgrading Proxy.

Architecture

Both standard architectures and cluster architecture support SSL encryption.

Prerequisites

The database instance is in **Running** status, with no ongoing tasks.

The operation is performed in off-peak hours, or the client has an automatic reconnection mechanism.

Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. In the **Instance ID**/**Name** column of the target instance, click the instance ID to enter the **Instance Details** page.
5. Click the **SSL Encryption** tab. If the system prompts you to upgrade the version under **SSL Encryption Settings**, click **Upgrade Version**, and wait until the version is successfully upgraded.
6. After **Encryption Status**, click to display **Updating SSL status...**.
7. Wait for **Encryption Status** to become **Enabled** and click **Download Certificate** in the upper right corner.
8. Wait for the **Enable SSL** status to become **Enabled** and click **Download Certificate**.
9. In the bottom-left corner of the page, upload the obtained certificate **-crt.zip** to the server, and then you can access the database over SSL.

For client connection code samples, see Java Connection Sample and Python Connection Sample.

Related APIs

API Name	Description
OpenSSL	Enables SSL
CloseSSL	Disables SSL

Monitoring and Alarms

Update Notes of Monitoring at Five-Second Granularity

Last updated：2023-03-14 15:41:47

Changes

The monitoring feature of TencentDB for Redis has been upgraded as follows:

The monitoring granularity is now narrowed from one minute to five seconds.

The monitoring data delay is now reduced to less than 20 seconds.

Monitoring, data collection, and alarms are now supported for replica nodes.

Monitoring, data collection, and alarms are now supported for proxy nodes.

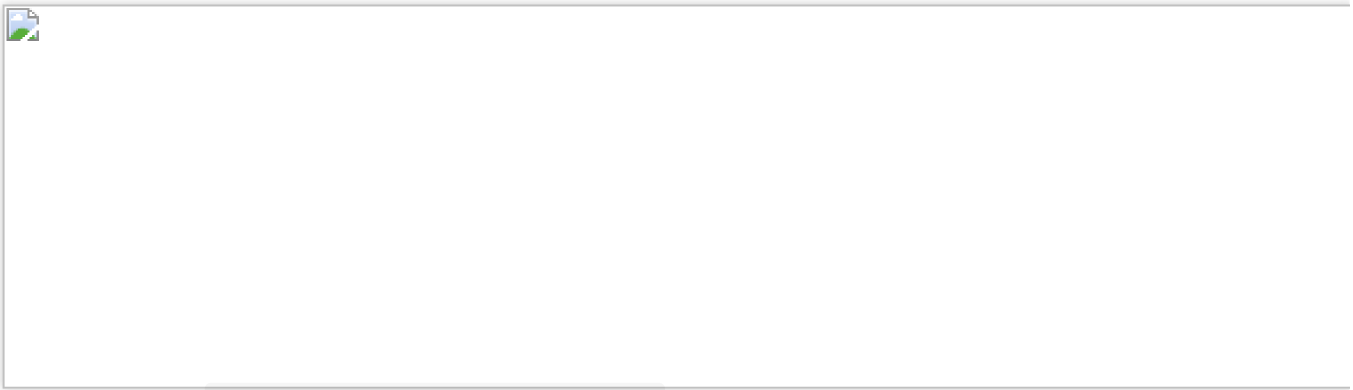
You can now compare monitoring metric values among multiple nodes.

New monitoring metrics are supported.

FAQs

How do I tell whether an instance supports 5-second or 1-minute monitoring granularity?

Log in to the TencentDB for Redis console, click an instance ID to enter the instance management page, select **System Monitoring** > **Monitoring Metrics**, and click the **Period** drop-down list at the top. If you can select **5 seconds** from the drop-down list, this instance supports the monitoring granularity of five seconds, or else it supports only the monitoring granularity of one minute.



Check the value of the `InstanceSet.MonitorVersion` field returned by the DescribeInstances API. If the value is `5s`, this instance supports the monitoring granularity of five seconds; if the value is `1m`, it supports only the monitoring granularity of one minute.

How do I get information of Proxy or Redis nodes?

Use the DescribeInstanceNodeInfo API to get the IDs of Proxy nodes and Redis nodes.

Notes

The IDs of Proxy and Redis nodes will change when node failover, instance capacity expansion/reduction, data migration, etc., occur. Therefore, we recommend that you get the latest node information from the API in a timely manner.

Monitoring at Five-Second Granularity

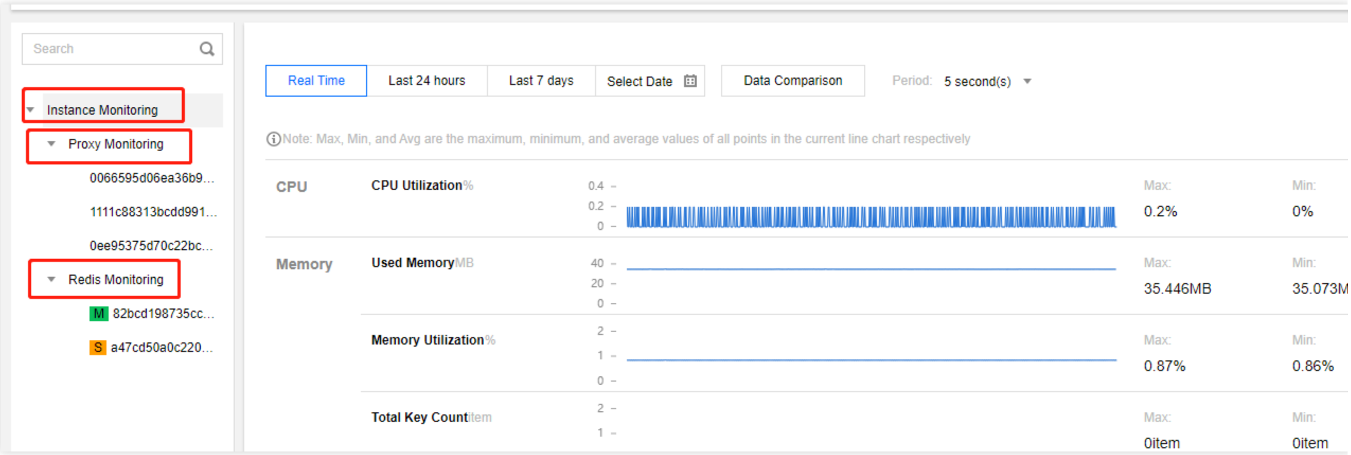
Last updated：2024-03-13 11:23:47

TencentDB for Redis provides comprehensive and transparent monitoring services. The new version, which supports 5-second monitoring granularity, includes Proxy node monitoring, Redis node monitoring, and instance monitoring overview, as detailed below:

Proxy node monitoring: Provides monitoring information of all Proxy nodes in an instance. TencentDB for Redis instances in standard or cluster architecture have Proxy nodes.

Redis node monitoring: Provides monitoring information of master and replica Redis nodes.

Instance monitoring: Instance monitoring aggregates monitoring data for the entire instance, including monitoring data for Proxy nodes and Redis nodes, aggregated using SUM, AVG, MAX, LAST, and other aggregation algorithms.



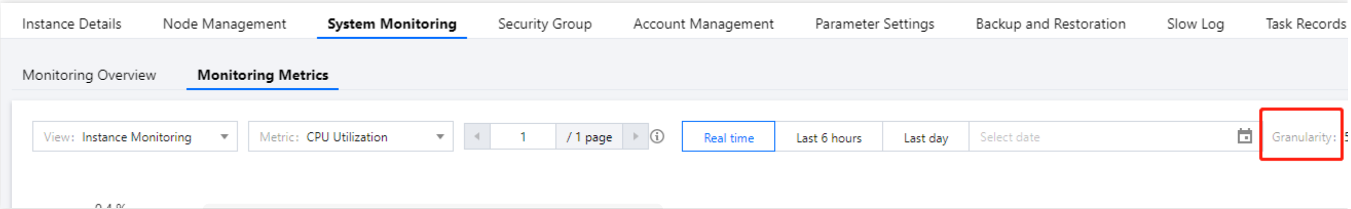
Period: 5 second(s)

Note: Max, Min, and Avg are the maximum, minimum, and average values of all points in the current line chart respectively

		Max:	Min:
CPU	CPU Utilization%	0.2%	0%
Memory	Used MemoryMB	35.446MB	35.073M
	Memory Utilization%	0.87%	0.86%
	Total Key Countitem	0item	0item

Viewing the Instance Monitoring Granularity

Log in to the TencentDB for Redis console, click an **Instance ID** to enter the Instance Management page, then select **System Monitoring** > **Monitoring Metrics**, and click the Period drop-down list at the top. If you can select 5 seconds from the drop-down list, it indicates that this instance supports the monitoring granularity of 5 seconds. TencentDB for Redis currently supports monitoring metrics at the 5-second, 1-minute, 5-minute, 1-hour, or 1-day granularity. For the retention period of monitoring data at each granularity, see Use Limits.



View: Instance Monitoring
Metric: CPU Utilization

Check the value of the `InstanceSet.MonitorVersion` field returned by the DescribeInstances API. If the value is `5s` , this instance supports the monitoring granularity of 5 seconds; if the value is `1m` , it supports the monitoring granularity of 1 minute.

Viewing the Monitoring Data

You can view TencentDB for Redis monitoring information through the Redis instance list, the Redis system monitoring page, and the Cloud Monitor console.

Viewing the Monitoring Data in the Instance List

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. In the target instance row, click 📊 in the **Monitoring**/**Status**/**Task** column to view the monitoring data on the right panel.

Viewing the Monitoring Data on the System Monitoring Page

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the **Instance List**, find the target instance.
4. Click the **Instance ID** in blue font to enter the **Instance Details** page. Then, click the **System Monitoring** tab and select the **Monitoring Metrics** tab to view the monitoring data.

Viewing the Monitoring Data in CM

Log in to the CM console to view the monitoring data overview.

Monitoring Metric Description

Proxy Node Monitoring

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

Group	Metric	Parameter	Unit	Description
CPU	CPU Utilization	cpu_util	%	CPU utilization of the proxy
Request	Total Requests	proxy_commands	Count/s	The number of commands executed by the Proxy per second

	Key Requests	cmd_key_count	Count/s	The number of keys accessed by a command per second
	Mget Requests	cmd_mget	Count/s	The number of MGET commands executed per second
	Execution Errors	cmd_err	Count/s	The number of command execution errors in the proxy, such as unexisting command, incorrect parameters, and so forth.
	Big Value Requests	cmd_big_value	Count/s	The number of executions for requests larger than 32 KB per second
Network monitoring	Connections	connections	Count	The number of TCP connections connected to the instance
	Connection Utilization	connections_util	%	The ratio of the number of TCP connections to the maximum number of connections
	Max Connections Utilization of Node	connections_max_util	%	The maximum value of the connection utilization ratio of all Proxy nodes, where the connection utilization ratio is calculated as the current number of connections on a node divided by the maximum number of connections on that node.
	Inbound Traffic	in_flow	MB/s	Private network inbound traffic
	Inbound Traffic Utilization	in_bandwidth_util	%	The ratio of actual utilization to the maximum traffic of private inbound traffic.
	Inbound Traffic Limit Count	in_flow_limit	Count	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 actual utilization to the maximum traffic of private outbound traffic.

	Outbound Traffic Limit Count	out_flow_limit	Count	The number of times outbound traffic triggers a traffic limit
Latency monitoring	Average Execution Latency	latency_avg	ms	The average execution latency between the Proxy and the Redis server
	Max Execution Latency	latency_max	ms	The maximum execution latency between the Proxy and the Redis server
	Average Read Latency	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For more information on read command types, see Command types.
	Average Write Latency	latency_write	ms	The average execution latency of write commands between the proxy and the Redis server. For more information on write command types, see Command types.
	Average Latency of Other Commands	latency_other	ms	The average execution latency of commands (excluding write and read commands) between the proxy and the Redis server.

Redis Node Monitoring

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

Group	Metric	Parameter	Unit	Description
CPU monitoring	CPU Utilization	cpu_util	%	Average CPU utilization
Network	Connections	connections	Count	The number of connections between the Proxy and a node
	Connection Utilization	connections_util	%	The connection utilization of the node
Memory monitoring	Memory usage	mem_used	MB	The amount of memory actually used by the user, including data and caches

	Memory Utilization	mem_util	%	The ratio of the actually used memory to the requested total memory
	Total Keys	keys	Count	The total number of keys (Primary Keys) stored in the instance
	Expired Keys	expired	Count	The number of keys expired within a time window, which is equal to the value of `expired_keys` output by the `info` command.
	Evicted Keys	evicted	Count	The number of keys evicted within a time window, which is equal to the value of `evicted_keys` output by the `info` command.
	Keys with Expiry Time Set	expires	Count	Number of Keys with Expiry Time Set in the Instance (Primary Key)
	Replication Delay	repl_delay	Byte	The relative command delay length of the secondary node to the primary node.
Request monitoring	Total Requests	commands	Count/s	QPS
	Read Requests	cmd_read	Count/s	The number of read command executions per second. For more information on read command types, see Command types.
	Write Requests	cmd_write	Count/s	The number of write command executions per second. For more information on write command types, see Command types.
	Other Requests	cmd_other	Count/s	The number of command (excluding write and read commands) executions per second
Response monitoring	Slow Logs	cmd_slow	Count	The number of command executions with a latency greater than the `slowlog-log-slower-than` configured
	Read Request Hits	cmd_hits	Count	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	cmd_miss	Count	The number of keys unsuccessfully requested

	Request Misses			by read commands, which is equal to the value of the `keyspace_misses` metric output by the `info` command.
	Read Request Hit Rate	cmd_hits_ratio	%	Key hits/(Key hits + Key misses). This metric can reflect the situation of Cache Miss.

Redis Instance Monitoring

The instance 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.

Group	Metric	Associated View	Parameter	Unit	Description
CPU monitoring	CPU Utilization	Redis node	cpu_util	%	Average CPU utilization
	Max CPU Utilization of Node	Redis node	cpu_max_util	%	The maximum CPU utilization of a node (shard or replica) in an instance
Memory monitoring	Memory usage	Redis node	mem_used	MB	The amount of memory actually used by the user, including data and caches
	Memory Utilization	Redis node	mem_util	%	The ratio of the actually used memory to the requested total memory
	Max Memory Utilization of Node	Redis node	mem_max_util	%	The maximum among all node (shard or replica) memory utilizations in an instance
	Total Keys	Redis node	keys	Count	The total number of keys (Primary Keys) in instance storage
	Expired Keys	Redis node	expired	Count	The number of keys expired within a time window, which is equal to the value of `expired_keys` output by the `info` command.
	Evicted	Redis	evicted	Count	The number of keys evicted

	Keys	node			within a time window, which is equal to the value of `evicted_keys` output by the `info` command.
	Keys with Expiry Time Set	Redis node	expires	Count	Number of Keys with Expiry Time Set in the Instance (Primary Keys)
Network monitoring	Connections	Proxy node	connections	Count	The number of TCP connections connected to an instance
	Connection Utilization	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 actual utilization to the maximum traffic of private inbound traffic
	Inbound Traffic Limit Count	Proxy node	in_flow_limit	Count	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 actual utilization to the maximum traffic of private outbound traffic
	Outbound Traffic Limit Count	Proxy node	out_flow_limit	Count	The number of times outbound traffic triggers a traffic limit
	Average Execution Latency	Proxy node	latency_avg	ms	The average execution latency between the proxy and the Redis server
	Max Execution	Proxy node	latency_max	ms	The maximum execution latency between the proxy

	Latency				and the Redis server
	Average Read Latency	Proxy node	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For more information on read command types, see Command types.
	Average Write Latency	Proxy node	latency_write	ms	The average execution latency of write commands between the proxy and the Redis server. For more information on write command types, see Command types.
	Average Latency of Other Commands	Proxy node	latency_other	ms	The average execution latency of commands (excluding write and read commands) between the proxy and the Redis server.
Request monitoring	Total Requests	Redis node	commands	Count/s	QPS
	Read Requests	Redis node	cmd_read	Count/s	The number of read command executions per second. For more information on read command types, see Command types.
	Write Requests	Redis node	cmd_write	Count/s	The number of write command executions per second. For more information on write command types, see Command types.
	Other Requests	Redis node	cmd_other	Count/s	The number of command (excluding write and read commands) executions per second
	Big Value Requests	Proxy node	cmd_big_value	Count/s	The number of executions for requests larger than 32 KB per second

	Key Requests	Proxy node	cmd_key_count	Count/s	The number of keys accessed by a command per second
	Mget Requests	Proxy node	cmd_mget	Count/s	The number of MGET commands executed per second
	Slow Logs	Redis node	cmd_slow	Count	The number of command executions with a latency greater than the `slowlog-log-slower-than` configured
	Read Request Hits	Redis node	cmd_hits	Count	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	Redis node	cmd_miss	Count	The number of keys unsuccessfully requested by read commands, which is equal to the value of the `keyspace_misses` metric output by the `info` command.
	Execution Errors	Proxy node	cmd_err	Count/s	The number of command execution errors. For example, the command does not exist, parameters are incorrect, etc.
	Read Request Hit Rate	Redis node	cmd_hits_ratio	%	Key hits/(key hits + key misses). This metric can reflect the situation of Cache Miss.

Command Types

Command Type	List

Read commands	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,geodis
Write commands	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,georadiusbymem pfadd,pfmerge,pfdebug

Querying the Node Information

Use the DescribeInstanceNodeInfo API to get the IDs of Proxy nodes and Redis nodes.

Note:

In scenarios such as node failover, instance capacity expansion/reduction, or data migration, the node IDs for Proxy and Redis may change. Therefore, it is necessary to promptly obtain the latest node information from the API interface.

Related APIs

API Name	Description
DescribeInstanceMonitorBigKey	Queries big keys of an instance
DescribeInstanceMonitorBigKeySizeDist	Queries the big key size distribution of an instance
DescribeInstanceMonitorBigKeyTypeDist	Queries the big key type distribution of an instance
DescribeInstanceMonitorHotKey	Queries hot keys of an instance
DescribeInstanceMonitorSIP	Queries the access source information of an instance
DescribeInstanceMonitorTookDist	Queries the distribution of instance access time
DescribeInstanceMonitorTopNCmd	Queries an instance access command
DescribeInstanceMonitorTopNCmdTook	Queries the CPU time consumption of an instance

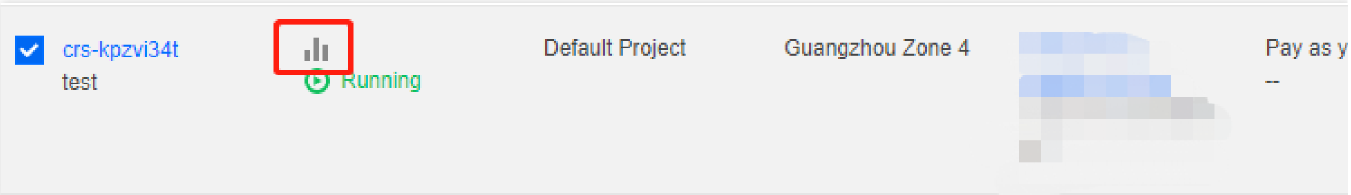
Comparing Monitoring Data Among Instances

Last updated：2023-06-06 15:37:23

This document describes how to compare the monitoring data of multiple TencentDB for Redis instances in the same chart in the console for easier troubleshooting.

Directions

1. Log in to the TencentDB for Redis console and click the icon below in the instance list to enter the monitoring page.



2. Click **Compare Monitoring Data of Instances** on the right to enter the dashboard page in the TCOP console.
3. On the **Dashboard List** page, create a dashboard and a chart to view and compare monitoring data of multiple instances.

Configuring Alarms

Last updated：2023-05-23 10:59:48

Overview

You can configure alarm rules for monitoring metrics to prevent your system operations from being disrupted when these metrics reach a certain value. When monitoring data meets the configured conditions, the system can check it automatically and send alarm notifications to the admin. This allows you to stay on top of business exceptions and solve them quickly.

Billing Details

TCOP allows you to configure alarm policies to monitor the key metrics of instances and offers a free trial. Currently, only **alarm SMS and phone calls** are billed.

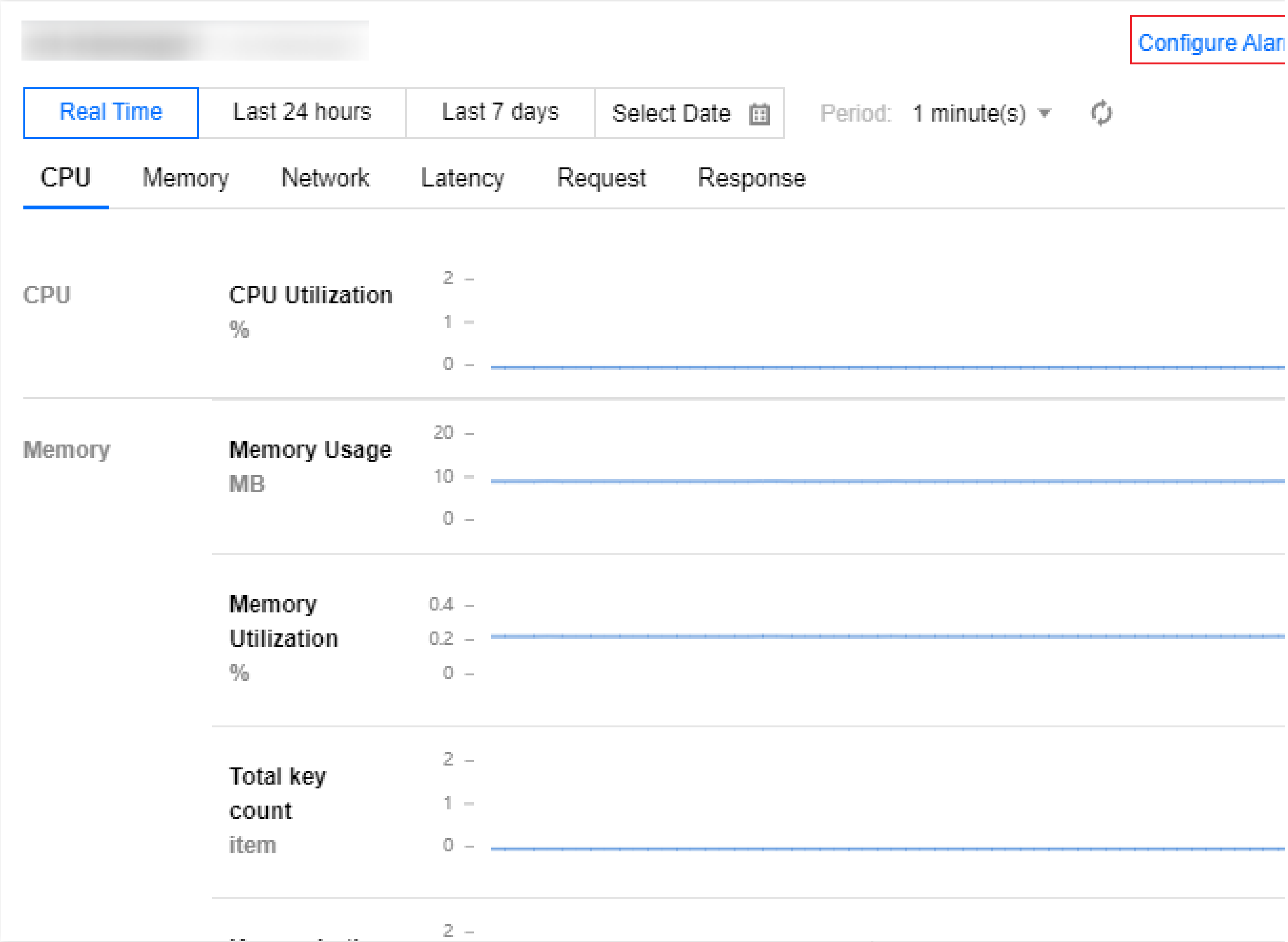
Prerequisites

You have activated Tencent Cloud Observability Platform (TCOP).
The database instance is in **Running** status.
You have collected the information of the recipients of alarm notifications, such as email address and phone number.

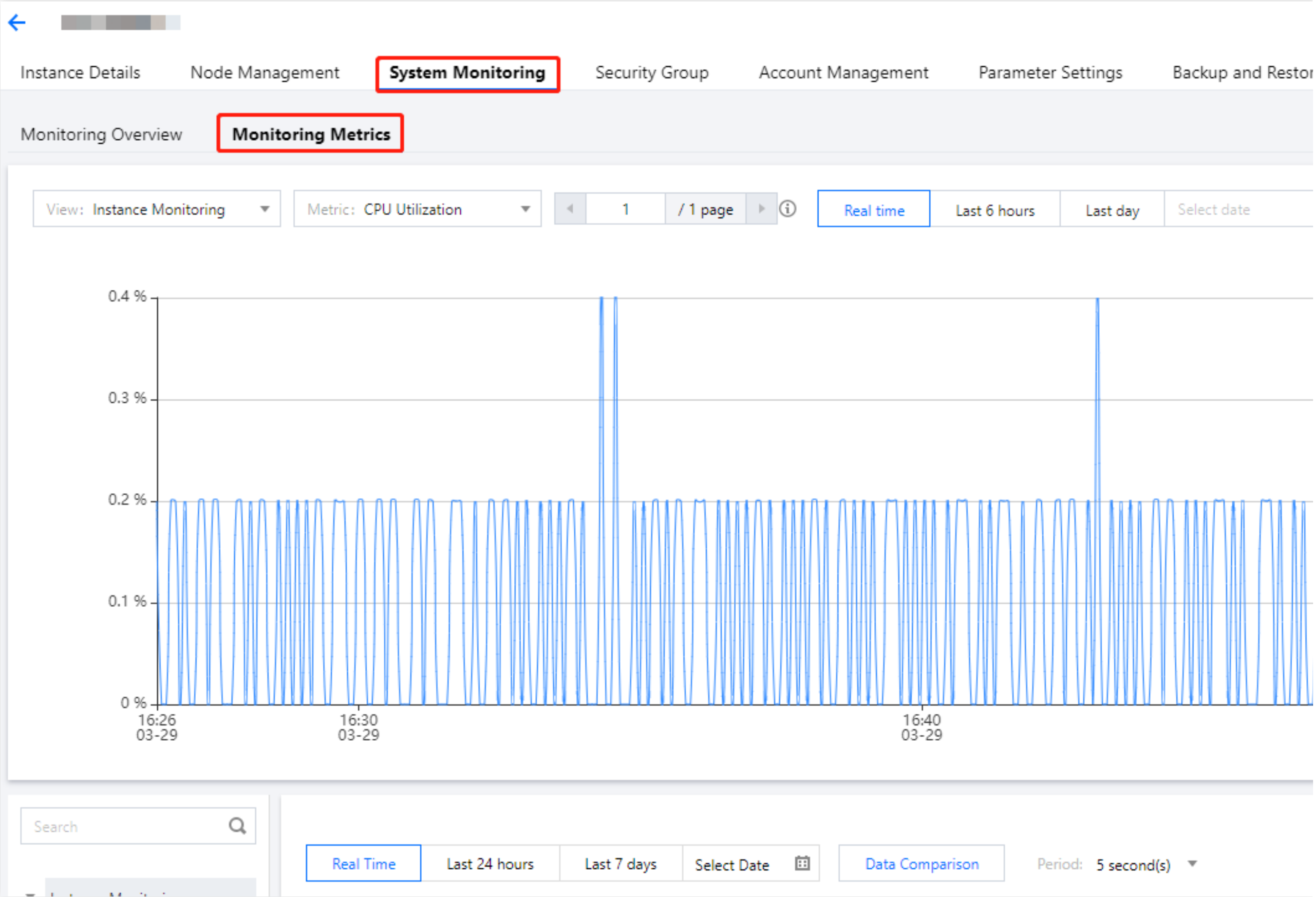
Directions

1. Log in to the TencentDB for Redis console.
2. Above the **Instance List** on the right, select the region.
3. In the instance list, find the target instance.
4. In the row of the target instance, enter the **Create Policy** page of TCOP in any of the following ways:

Click 📊 in the **Monitoring**/**Status**/**Task** column and click **Configure Alarms** in the top-right corner of the instance monitoring dashboard.



Click the **instance ID** in blue to enter the **Instance Details** page. Then, click the **System Monitoring** tab, select the **Monitoring Metrics** tab, and click **Set Alarms**.



5. On the **Create Policy** page, configure the policy as shown below. For more information on the basic concepts of alarm policy, see Creating Alarm Policy.

Parameter	Description
Policy Name	Customize the alarm policy name for easier identification.
Remarks	Briefly describe the alarm policy for easier identification.
Monitoring Type	Select **Cloud Product Monitoring**.
Policy Type	Select **TencentDB/Redis/Memory Edition (5-Second Granularity)/Redis Node** or **TencentDB/Redis/Memory Edition (1-Minute Granularity)/Redis Node** as needed.
Project	Specify a project for the alarm policy. You can quickly locate all alarm policies of a project in the alarm policy list.
Alarm Object	If you select **Instance ID**, the alarm policy will be bound to the specified database instance. If you select **Instance Group**, the alarm policy will be bound to the specified database instance group. For more information on how to create an instance group, see Instance Group. If you select **All Objects**, the alarm policy will be bound to all instances the current account has permission on.

	If you select **Tag**, the alarm policy will be bound to all instances associated with the current tag key and tag value.
Trigger Condition	If you select **Select template**, you can select a template file in the drop-down list, and alarms will be reported based on the trigger conditions preset in the template. For specific configurations, see Configuring Trigger Condition Template. If you select **Configure manually**, you need to configure the threshold for triggering an alarm for each metric in the **Metric Alarm** section below. **Threshold Type** in the **Metric Alarm** section: If you select **Static**, you can manually set a fixed threshold, and alarms will be triggered when the threshold is reached. If you select **Dynamic**, exceptions will be determined based on the dynamic threshold boundaries calculated by machine learning algorithms. For more information, see Creating Alarm Policy.
Alarm Notification	Select a preset or custom notification template. Each alarm policy can be bound to three notification templates at most. For more information, see Alarm Notification.

6. After confirming that the configuration is correct, click **Complete**. For more information on alarms, see Overview.

Related APIs

API Name	Description
CreateAlarmPolicy	Creates a TCOP alarm policy

Creating Event Rule

Last updated：2023-03-21 16:17:39

Overview

TencentDB for Redis has been connected to CM to report cloud monitoring events. All events will be automatically delivered to the Event Bus in EventBridge.

CM event

Currently, the following **events** can be reported:

Event	Name	Event Type	Dimension	Recoverable	Description
Master/Replica switch	MasterSlaveSwitched	Status change	TencentDB for Redis instance	No	A failover occurs in TencentDB for Redis.
Unavailable service	ServiceNotAvailable	Exception	TencentDB for Redis instance	Yes	The TencentDB for Redis service is unavailable due to a failure.

Read-only replica failover	ReadonlyReplicaSwitched	Status change	TencentDB for Redis instance	Yes	A failover occurs in the TencentDB for Redis read-only replica.
Unavailable read-only replica	ReadonlyReplicaNotAvailable	Exception	TencentDB for Redis instance	Yes	A failure occurs in the TencentDB for Redis read-only replica.

Instance migration caused by server failure	ServerfailureInstanceMigration	Status change	TencentDB for Redis instance	Yes	A server error occurs in TencentDB for Redis.

Event target

An event rule can have multiple event targets. Before creating an event rule, plan the event target types first. Currently, EventBridge supports the following event targets.

Message Push Target (supported by rules only in the Tencent Cloud service event bus)

CLS

SCF

CKafka

Billing Overview

EventBridge is **pay-as-you-go**. For more information, see Product Pricing.

Type	**Pay-as-You-Go**
Payment mode	Fees are charged hourly by the number of events actually delivered to the event bus.
Billing unit	USD/million events

Use cases	Low or fluctuating message volumes

Directions

1. Log in to the EventBridge console and select Event rule on the left sidebar.
2. At the top of the page on the right, select **Guangzhou** for **Region** and select **default** in the **Event Bus** drop-down list.
The Tencent Cloud service event bus collects monitoring and audit events generated by Tencent Cloud services in all regions. It is created in Guangzhou region by default and cannot be deleted.
Select **Event Bus** on the left sidebar. In the event bus list, click **default**, and you can see that the **default** event bus already contains TencentDB for Redis. For detailed directions, see Tencent Cloud Service Event Source.
3. Click **Create**. On the **Event Pattern** page, configure parameters by referring to the following descriptions:

Section	Parameter	Description
Basic information	Region	The region where to create an event rule.
	Event bus	Information of the event bus to which the event rule belongs.
	Rule name	Event rule name, which can contain 2–60 letters, digits, underscores, and hyphens. It must start with a letter and end with a digit or letter.
	Rule description	Brief description of the event rule.
	Tag	Tag key and value of the event.
	Data conversion	Choose whether data conversion is required.
Sample event	Sample event	In the drop-down list, search for Redis to view sample events of Redis.
Event matching	Write mode	Form mode: In this mode, you can select **Tencent Cloud service** and **Event type** to see the event matching rule. Custom events: Customize the event matching rule.
	Tencent Cloud service	This parameter will be displayed if **Form mode** is selected for **Write mode**. In the drop-down list, select **TencentDB for Redis**.
	Event type	This parameter will be displayed if **Form mode** is selected for **Write mode**. In the drop-down list, select the supported event type.
	Event	Preview the generated event matching rule.

	matching rule preview	

4. Click **Test matching rule** to test the defined event matching rule.

5. (Optional) To convert the data format, enter the event transformation page. Configure the data conversion formats and fields according to the parameter descriptions in the following table and click **OK** to start parsing the data. After data parsing is completed, set the filter rule and data processing method. For detailed directions, see Configuring Data Conversion.

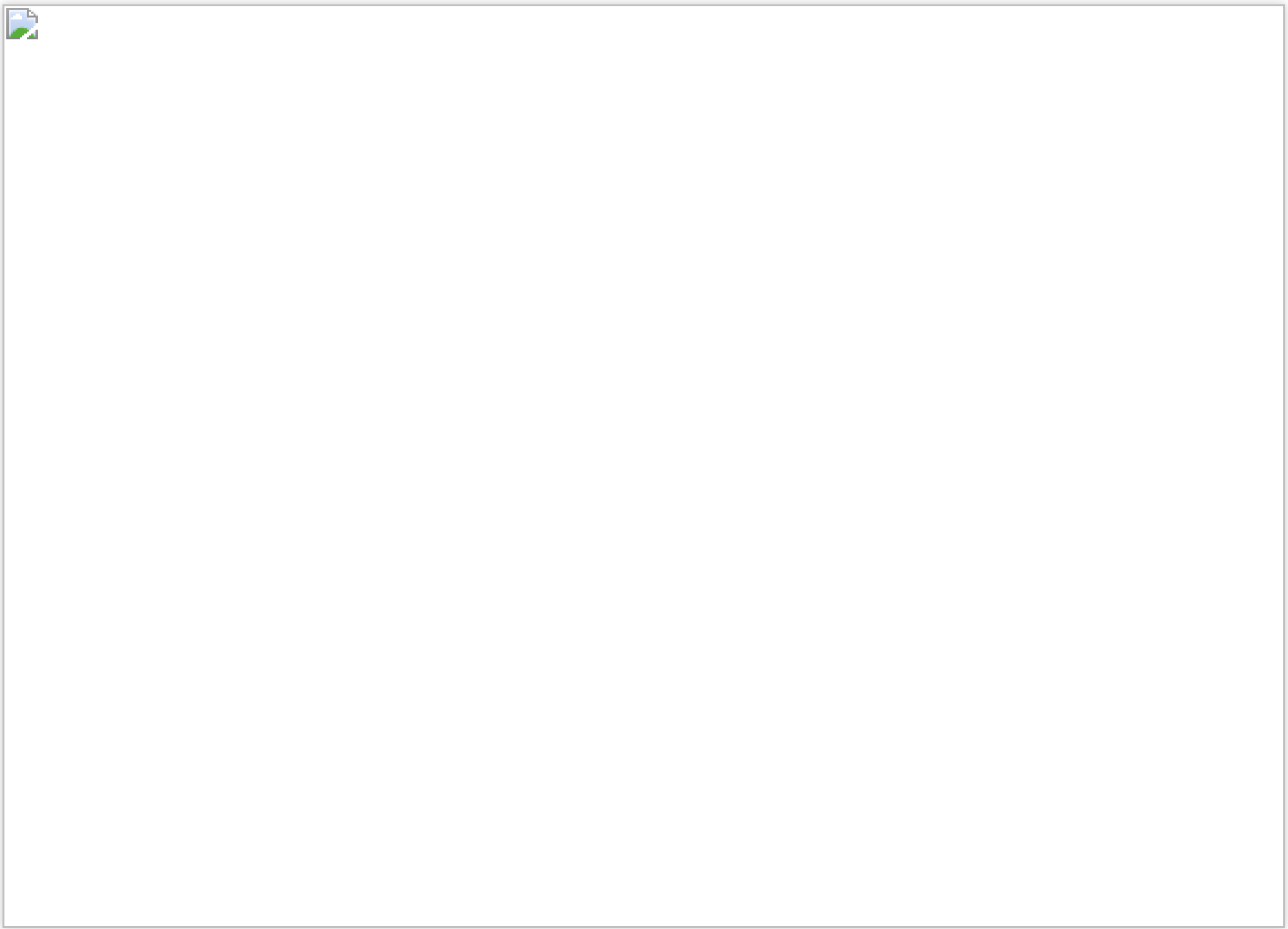
Notes

EventBridge provides simple data processing capabilities. After you pass in data and configuration items to EventBridge, EventBridge formats the data and distributes the structured data obtained after processing to downstream targets, creating a bridge between data sources and data processing systems.

Parameter	Description
Rule pattern preview	If you select **Sample event**, you can use an event template. If you select **Manually input**, you can customize the event fields in the input box below.
Event template	This parameter will be displayed if **Sample event** is selected for **Rule pattern preview**. You can search for "Redis" in the drop-down list and select the TencentDB for Redis event template. Then, the specific field information of the event template will be displayed in the input box below.
Target	**All events**: The complete structure of event fields will be routed to the delivery target.

	Specified events: EventBridge will extract event parameters according to the event fields configured in `JSONPath` to route the specified event fields to the delivery target.
JSONPath	This parameter will be displayed if **Specified events** is selected for **Target**. Enter the event fields to be converted in the input box.
Parsing mode	Select a parsing mode, which can be **JSON**, **Separator**, or **Extract with regex**.

6. Click **Next** to select the delivery target bound to the rule. You can deliver collected events to the specified target to process and consume them. In the figure below, **Notification message** is selected for **Trigger method** as an example.



7. To make the event rule take effect immediately, select **Enable event rules now** and click **Complete**.

Event Rule APIs

API	Description
CheckRule	Checks a rule

CreateRule	Creates an event rule
DeleteRule	Deletes an event rule
GetRule	Gets the details of an event rule
ListRules	Gets the list of event rules
UpdateRule	Updates an event rule

More Operations

You can view, modify, or delete an event rule. For more information, see Managing Event Rule.

FAQs

For questions about event rule concepts and billing, see Concept.

Viewing Alarm Information

Last updated：2023-05-23 11:03:17

Overview

You can view the following three types of alarms triggered in a certain period of time in the TCOP console. Information of alarms with custom messages cannot be viewed in the console for the time being.

Basic alarms: alarms sent by preset alarming metrics.

Customized alarms: Alarms sent by the customized monitoring feature.

Directions

1. Log in to the TCOP console.
2. Select **Alarm Management** on the left sidebar.
3. On the **Alarm Records** tab on the right, you can filter TencentDB alarms by time period or alarm information. For more information, see Viewing Alarm Records.

(Optional) You can filter alarms generated today, yesterday, and in the last 7 days or 30 days, and you can also select a custom time period. You can view the alarm records in the last six months at most.

(Optional) You can enter the information of an alarm object (such as instance name, public IP, and private IP) in the search box at the top-right corner to search for corresponding records.

(Optional) You can also click **Advanced Filter** to search for alarm records by policy name, alarm content, user information, monitor type, and policy type.

More operations

For more information about TCOP alarm service, see Creating Alarm Policy.

For more information on how to manage alarm policies, see Creating Alarm Policy.

For more information on how to manage alarm notifications, see Creating Notification Template.

Event Management

Review Pending Events

Last updated：2024-03-13 11:12:59

In addition to notifications through SMS, emails or site messages, Ops events of TencentDB for Redis (such as instance migration, etc.) will also be notified in the console. To ensure the continuity of cloud services and efficiently manage Ops events, TencentDB for Redis supports viewing event types, event statuses, event levels, related instance lists, service impacts, etc. in event management for specific regions. You can also arrange the event execution plan reasonably based on the severity and urgency of Ops events in line with actual service conditions.

Version Description

TencentDB for Redis versions **4.0**, **5.0**, and **6.2**, both standard and cluster architecture instances **support** event management.

Prerequisites

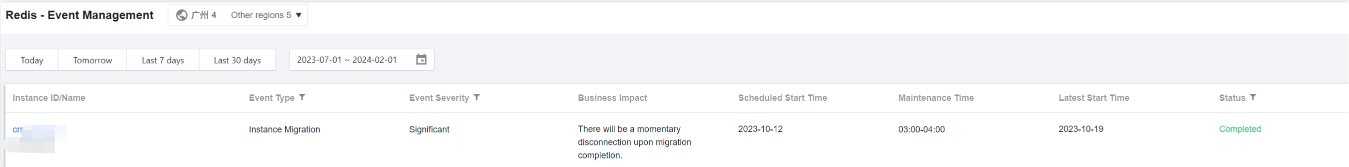
Instance operation status: Running.

Directions

1. Log in to the TencentDB for Redis console.
2. On the left navigation, select **Event Management**.
3. On the right side, select the region from the drop-down list at the top of the **Redis > Event Management** page.
4. Filter the event's planned start time range. You can directly select: **Today**, **Tomorrow**, **Last 7 days**, **Last 30 days**, or click 📅 in the time box to select a specific time period. The event list defaults to displaying events from the **Last 7 days**.
5. In the event list below, peek at all pending Ops events in the selected time period for this region.

Note:

The event list is arranged in reverse order according to the planned execution date of the Ops event.



Instance ID/Name	Event Type	Event Severity	Business Impact	Scheduled Start Time	Maintenance Time	Latest Start Time	Status
[redacted]	Instance Migration	Significant	There will be a momentary disconnection upon migration completion.	2023-10-12	03:00-04:00	2023-10-19	Completed

Interface Parameter	**Parameter Meaning**
Instance ID / Name	Instance ID: The unique identification ID of the instance. Name: The name set when creating the instance. Hover over the instance name and click to edit it for easy recognition and management.
Event type	The type of Ops event that the current instance needs to be dealt with, currently only supports **Instance Migration**, which refers to resource migration and the closure of a computer room.
Event Severity	The event level is graded according to its severity and urgency, from heavy to light they are **Severe**, **Significant**, **Moderate**, **Low**.
Service Impact	Describes the impact of this event execution on current service.
Scheduled Start Time	The date to start executing the event as planned.
Maintenance Time	The maintenance window of the current event is individually set **HH:MM - HH:MM**. When the **Planned Start Date** is reached, maintenance operations will be performed within the **Planned Maintenance Window**. Changes in the instance maintenance window will not affect the event's maintenance window.
Latest Start Time	The latest time for the execution of the Ops event, i.e., the event must be completed before this date, otherwise it may affect the service.
Status	Displays the current event running status, including: **Pending Execution**, **Running**, **Completed**, **Cancelled**. **Pending Execution**: Events that have not reached the execution date or are not within the maintenance window. **Running**: Events that have reached the date, are within the maintenance window, and are being executed. **Completed**: Events that have been completely executed. **Cancelled**: Events that have been cancelled.
Completion Time	If the current event has been completed, the completion time will be displayed.

Operation	Manage the event, modify the event execution time, or cancel the execution of this event. Click **Modify Execution Plan**, in the **Modify Execution Plan** window, modify the event execution time. For detailed operations, see Modify the Event Execution Plan. Click **Cancel Execution** to cancel the execution of this Ops event. **Note:** If the event is currently **Running** or **Completed**, the execution plan cannot be modified and the execution cannot be cancelled. For events with a level of **Critical** or **Important**, cancellation is not supported. That is, serious events must be executed and cannot be cancelled.

Related APIs

API Name	Description
DescribeEventInfo	Query event information
ModifyEventSwitchTime	Modify event maintenance time window

Modify Event Execution Plan

Last updated：2024-03-13 11:10:52

In Event Management, you can flexibly arrange the execution time of operations events based on your actual service conditions and the severity and urgency of the Ops events. This ensures the efficiency and accuracy of Ops work while minimizing the impact on services.

Version Description

TencentDB for Redis instances with **4.0**, **5.0**, or **6.2** standard or cluster architecture, whether deployed in a single availability zone or multiple availability zones, all **support** Event Management.

Prerequisites

Instance operation status: Running.

Directions

1. Log in to the TencentDB for Redis console.
2. On the left navigation, select **Event Management**.
3. On the right side, select the region from the drop-down list at the top of the **Redis > Event Management** page.
4. In the **Event Management** event list, select the event you need to modify.
5. Click the **Modify Execution Plan** in the **Operation** column.
6. In the **Modify Execution Plan** window, understand the impact of this event, confirm the instance information, modify the start date of the event in **Modify Start Date**, and in **Modify Time Window**, choose the **Start time** and **End time** to reschedule the execution plan.

Note:

For events classified as **Critical**, and the start date cannot be modified; if the event level is **Critical** or **Important**, the execution cannot be cancelled.

The latest start date refers to the latest allowable start time for an event. If it exceeds this time, it may adversely affect the whole schedule. The rescheduling start date cannot be later than the latest start date.

After selecting the start time in the **Modify Time Window**, the end time can only be scheduled within 30 minutes, 1 hour, 1.5 hours, 2 hours, and 3 hours after the start time.

7. click **OK**.

Related APIs

API Name	Description
DescribeEventInfo	Query event information
ModifyEventSwitchTime	Modify event maintenance time window

Global Replication
Overview

Last updated：2024-04-22 16:24:37

TencentDB for Redis supports global replication for consistent data sync across regions.

Why Global Replication

When the native code of Redis performs cross-region replication, if the local instance is written excessively or is disconnected from replication for a long time, the `backlog` of the master node may not be able to resume the replication, resulting in mandatory full replication by the remote instance. This will block access to and seriously affect the normal business of the remote instance. For more information, see Replication.

In addition, Redis' native replication scheme does not mark written nodes in the `backlog` , which tends to cause replication loops and data inconsistency when two-way replication is required between the local and remote instances.

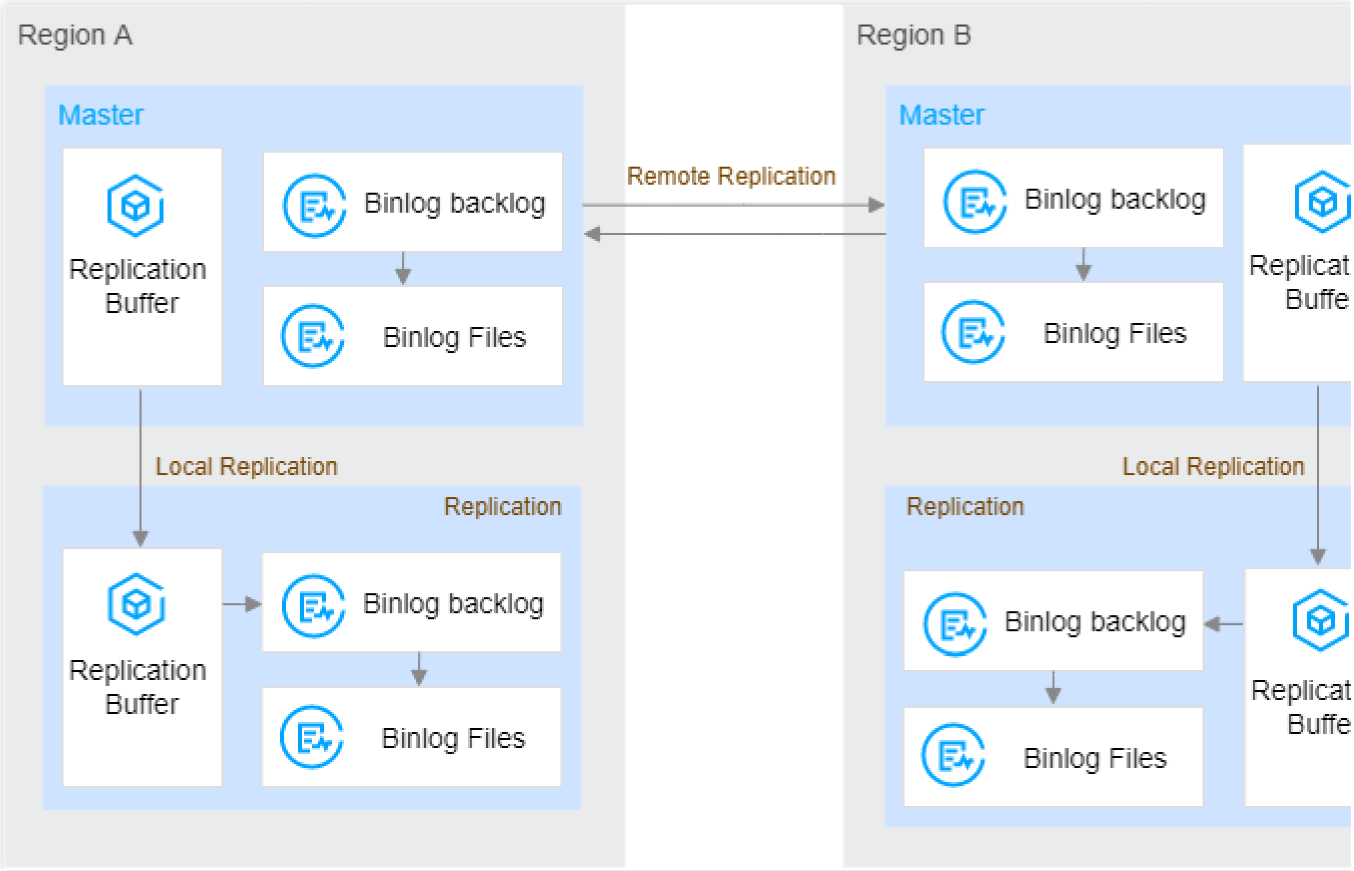
Implementation Mechanism

Global replication is a new feature added to the open-source version of the kernel by Tencent Cloud. It is fully compatible with Redis 4.0 and Redis 5.0 commands. Based on the original leader-follower replication scheme, a new log file is added for remote replication to ensure the eventual data consistency for instances in any region in the replication group.

Before the log file is replicated by the remote node, the local node will keep it to ensure the continuity of remote replication.

The log file contains the `ServerID` to identify the ID of the node to which the log file is written and supports two-way replication to avoid replication loops.

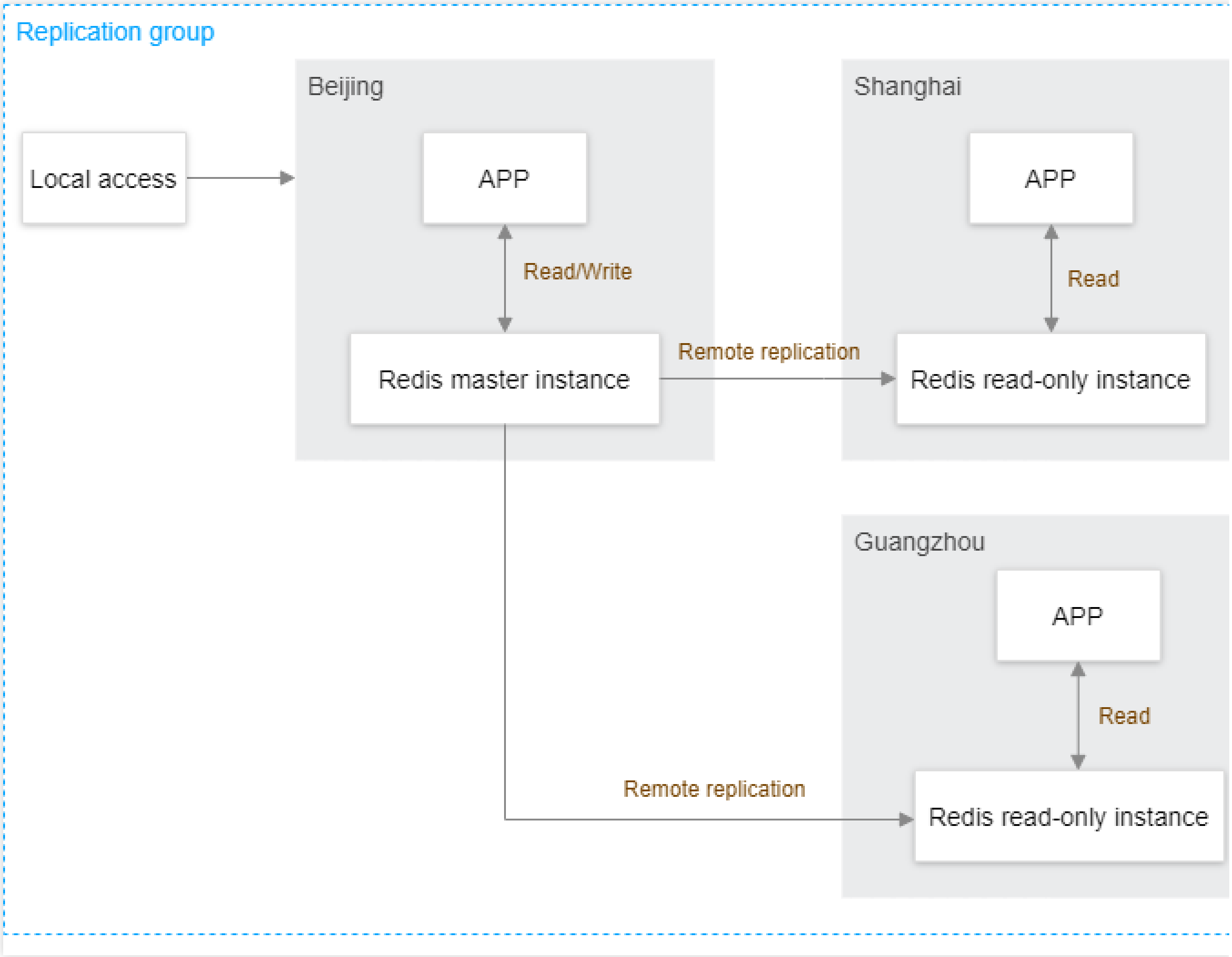
The log file contains the command execution timestamp and the version number of the operation `KEY` to resolve command conflicts.



Use Cases

Read-only instance and disaster recovery

In the global replication scheme of TencentDB for Redis, a master instance is configured in a replication group, while read-only instances are deployed in multiple regions and replicate data from the master instance. The data version is based on the master instance, and the data consistency level is eventual consistency. This allows you to access data locally with a quicker response, better experience, higher data availability, and greater data security.



Note：

Shard Quantity: For disaster recovery scenarios, it is recommended that the number of shards in the primary instance and the disaster recovery instance of the global replication group be set to a power of 2, with a maximum value of 64, such as 8, 16, 32, 64. Otherwise, it may lead to uneven shard storage capacity.

Transaction Commands: The transaction operations of global replication, compared to ordinary Redis transactions, will result in a certain degree of database performance degradation.

Multi-Key Commands: In global replication instances, multi-key commands are split by Slot. Compared to non-global replication group database instances, performance decreases, and the decrease ratio is related to the number of splits. The more splits, the greater the performance degradation.

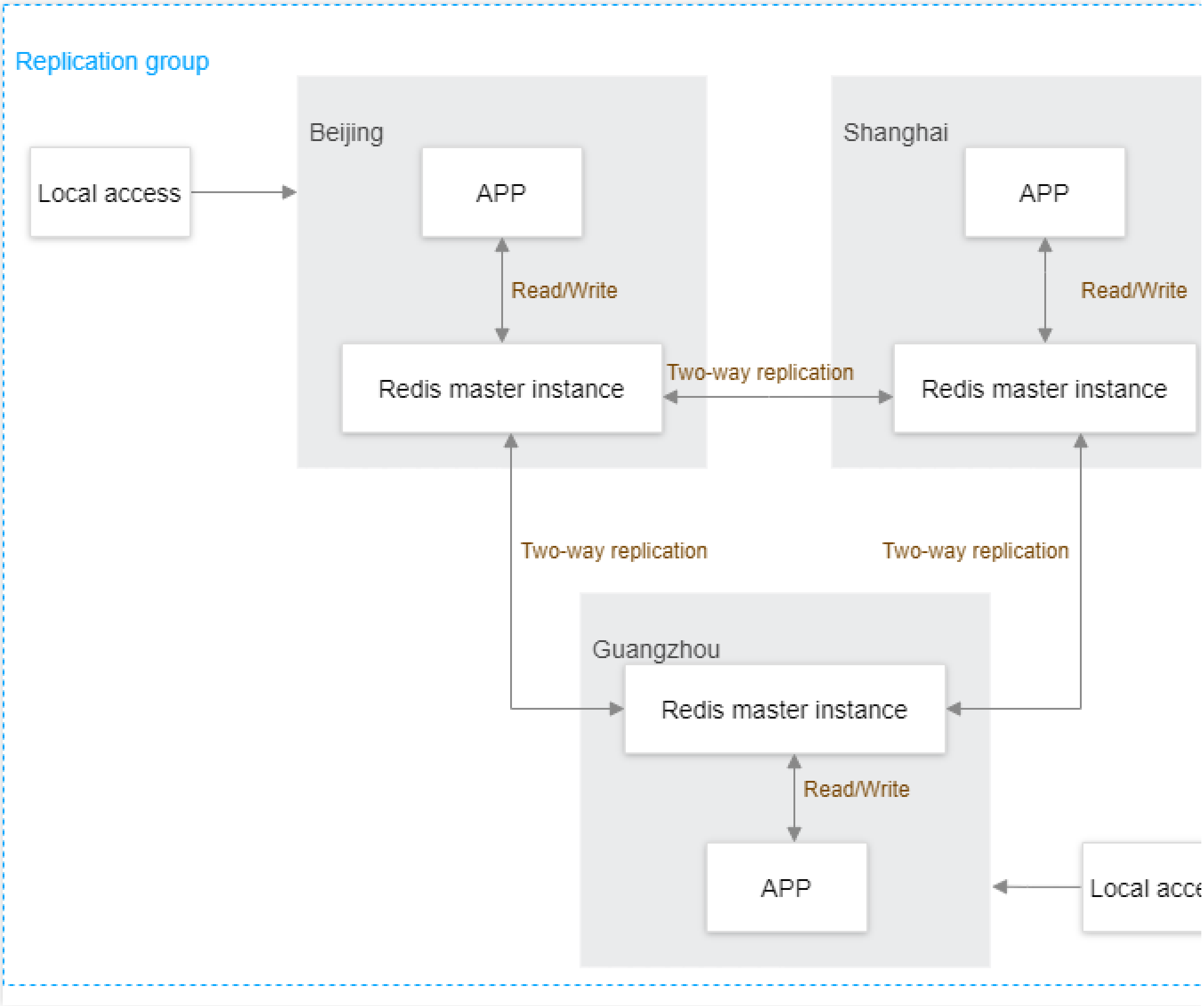
Data Consistency: The version of the data is based on the primary instance, and the level of data consistency is the final data consistency. When the master-slave node link of the primary instance fails, the remote replication of the disaster recovery instance will also be affected by delay. For business scenarios with high database consistency requirements, it is strongly recommended not to perform read operations on the disaster recovery instance.

Eviction Policy: Global replication does not support setting the eviction policy `maxmemory-policy`. If eviction is required, please use the `del` delete command to clear the Key.

Latency Description: In global replication, the synchronization delay between the master node and the replica node of the primary instance will affect the remote synchronization delay between the primary instance and the disaster recovery instance. The delay duration theoretically does not exceed 500ms.
Monitoring Metrics: Supports monitoring of command quantity deviation during the remote synchronization process between the primary instance and the disaster recovery instance, as well as synchronization delay monitoring. For specific information, please refer to Monitoring Metrics.

Multi-master architecture

If you need to roam and merge data across regions, you need to distribute the same copy of data in multiple regions, read and update data in any region, or merge data from multiple regions. In this case, the database should have the ability to write data in multiple regions. In a global replication group, you can configure a multi-master architecture, so data written to one master instance will be synced to other master instances in other regions.



Note:

TencentDB for Redis master instances do not perform version detection and write time check for data written by the application and by other master instances in the replication group; instead, they execute the commands chronologically in the order they are received. If the same data is updated in different master instances at the same time, the globally replicated data may be misaligned and fail to be consistent.
Therefore, in multi-master scenarios, avoid updating the same data in different master instances at the same time. As the multi-instance architecture may cause data inconsistency, carefully evaluate whether it is suitable for your business.

Use Limits

Last updated：2023-05-23 10:38:07

Limits on Region and AZ

The global replication feature can replicate data in the same AZ or across AZs between any Tencent Cloud regions no matter where instances in the replication group are deployed. Currently, only the following regions and AZs support global replication, and AZs of instances in a global replication group cannot be adjusted.

Region	AZ
Hong Kong (China)	Hong Kong Zones 2 and 3
Chengdu	Chengdu Zone 1
Virginia	Virginia Zone 2
Shanghai	Shanghai Zones 4 and 5
Beijing	Beijing Zones 5 and 7
Guangzhou	Guangzhou Zones 4, 5, and 6
Tianjin	Tianjin Zone 2
Nanjing	Nanjing Zones 2 and 3
Singapore	Singapore Zone 2
Shenzhen	Shenzhen Zone 4

Limits on the version and architecture of global replication group instances

Global replication supports instances running on 4.0 Standard Architecture, 4.0 Cluster Architecture, 5.0 Standard Architecture, and 5.0 Cluster Architecture.

The architectures of instances in a replication group cannot be changed; for example, you cannot change an instance from Cluster Architecture to Standard Architecture.

The version and architecture of instances to be added to a replication group must be the same as those of the master instance specified during group creation.

Limits on the specifications of global replication group instances

We recommend that you set the number of shards for instances in the replication group to the nth power of 2, such as 8, 16, 32, and 64 (maximum value).

When creating a replication group, you must specify a master instance with at least two replicas for the group.

Currently, you can add up to four instances in a global replication group in the following deployment schemes: one master and three read-only instances, four master instances, or two master and two read-only instances.

The specifications of instances to be added to a replication group must be the same as those of existing instances in the group, and their memory capacity must be greater than or equal to the used capacity of the master instance specified during group creation.

When you change the specifications, all instances in a replication group must have the same specifications; otherwise, performance or capacity problems may occur.

Limits on the parameter settings

The `maxmemory-policy` parameter of instances in a global replication group must be set to `noeviction` .

Limits on Command Sync

The **FLUSHDB** or **FLUSHALL** command will be synced to all instances in the replication group. Therefore, run them with caution.

The **Pub** and **Sub** command groups will not be synced. We recommend that you use the `Stream` data structure to replicate notification messages across regions.

When the **RESTORE** command is synced, if the target instance has the same key, it will not be executed.

Limits on the sync granularity

Currently, syncing is performed at the instance granularity, that is, all instance data will be synced. You cannot choose to sync partial instance data.

Creating Global Replication Group

Last updated：2023-05-23 10:25:58

TencentDB for Redis allows you to create a replication group in the console and add master or read-only instances to it, so as to implement consistent data sync in a one-master or multi-master architecture within the replication group.

Concepts

Instance role: You need to assign different roles to instances in a replication group, including **master instance** and **read-only instance**.

Master instance: It provides data read/write access and is used to write the business data.

Read-only instance: It provides the data read-only access and is used for read-only data operations or disaster recovery.

IP address: Each instance in a replication group has a separate IP address, which can be accessed independently.

Master/Replica switch: Automatic failover can be performed between master and replica nodes in each instance. However, it is not supported between master and read-only instances.

Version Description

Global replication supports instances running on 4.0 Standard Architecture, 4.0 Cluster Architecture, 5.0 Standard Architecture, and 5.0 Cluster Architecture.

The current version of the global replication feature supports only instances deployed in the same AZ. Multi-AZ instances will be supported in the future.

Billing Details

The instance syncs data from other instances in the replication group, no additional fees will be incurred.

Creating Global Replication Group

Prerequisites

You have created a TencentDB for Redis instance. For more information, see Creating TencentDB for Redis Instance.

The instance is in the **Running** status.

Directions

1. Log in to the TencentDB for Redis console.
2. Select **Global Replication** on the left sidebar.
3. On the **Redis - Global Replication** page on the right, click **Create Replication Group**.
4. In the **Create Replication Group** pop-up window, configure the following parameters and click **OK**.

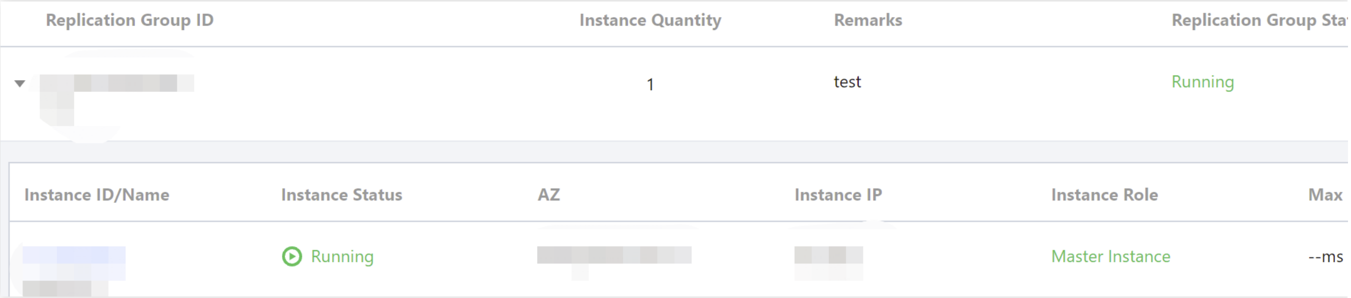
Parameter	Description	Required	Example
Error Code	Name of the replication group to be created. Enter a name as prompted.	Yes	test
Remarks	Brief description of the replication group. You can enter any characters to distinguish the group from others.	No	Replication group creation test
Master instance region	Select the region of the master instance in the replication group.	Yes	Guangzhou
Select master instance	Select the master instance in the replication group. The version, architecture, and memory capacity of the selected instance will be displayed, and you need to confirm whether the specification meets your requirements.	Yes	test-XXX

Note

The Redis kernel of the master instance specified during replication group creation must be upgraded to the Global Replication Edition. After the upgrade is completed, one or multiple momentary disconnections lasting 5 seconds will occur.

5. Return to the **Redis - Global Replication** page, and you can see the newly created replication group in the replication group list.

Click ▶ before the name of the replication group to show its instance list, where you can view the status of the master instance. You can use the master instance after the system upgrades its kernel to the Global Replication edition.



Replication Group ID	Instance Quantity	Remarks	Replication Group Sta
[redacted]	1	test	Running

Instance ID/Name	Instance Status	AZ	Instance IP	Instance Role	Max
[redacted]	Running	[redacted]	[redacted]	Master Instance	--ms

Adding Instance to Replication Group

After creating a replication group, you can add instances in the same or different regions and assign master and read-only instance roles to the added instances as needed to implement data sync.

Note

An instance newly added to a replication group will sync data from the master instance node, and it cannot be manipulated or accessed before the full data is synced.

Once an instance is added to a replication group, its kernel version will be upgraded, and momentary disconnections will occur after the upgrade.

Prerequisites

You have created a global replication group, and it is in **Running** status.

You have created an instance to be added to the replication group. Its compatible Redis version and architecture must be the same as those of the master instance specified during replication group creation, its memory capacity must be greater than or equal to the used capacity of the master instance, and it must be in **Running** status.

If you want to specify the instance to be added as a master instance, it must have at least two replica nodes.

When you add the first instance to a replication group, you don't need to clear the data. When adding subsequent instances, be sure to clear the data.

Note

The first instance added in the replication group has the following two scenarios:

The master instance added to a newly created replication group

The first instance added to an existing replication group after all instances in it are removed.

Directions

1. In the instance list on the **Redis - Global Replication** page, select the target replication group.
2. In the **Operation** column of the replication group, click **Add Instance**.
3. In the **Add Instance** pop-up window, read the notes, configure the following parameters, and click **OK**.

Region: Select the region of the target instance.

Select Instance: Select the target instance.

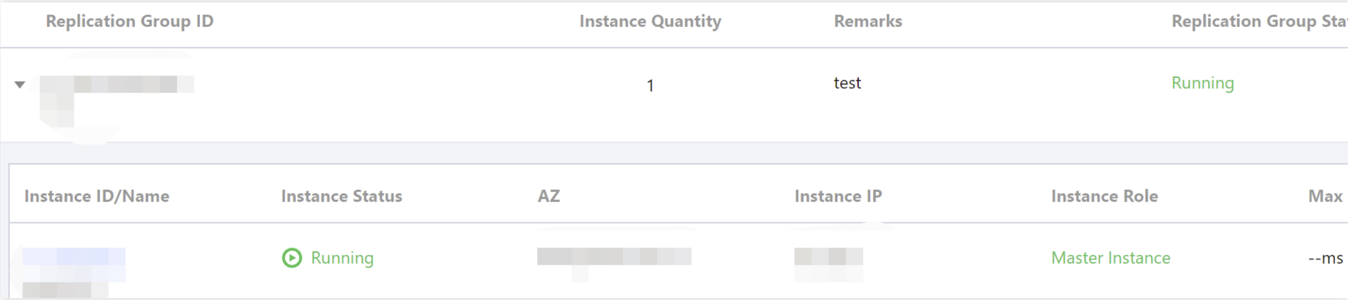
Instance Role: Assign a role (**master instance** or **read-only instance**) to the target instance.

Note

When there is no instance in the replication group, only the master instance can be added, and the read-only instance cannot be added.

If there are instances in the replication group, you can choose to add a read-only instance. If you need to add a master instance, submit a ticket.

4. Return to the **Redis - Global Replication** page. In the replication group list, click ▶ before the name of the target replication group to show its instance list, where you can see the newly added instance.



Replication Group ID	Instance Quantity	Remarks	Replication Group Sta
[redacted]	1	test	Running

Instance ID/Name	Instance Status	AZ	Instance IP	Instance Role	Max
[redacted]	Running	[redacted]	[redacted]	Master Instance	--ms

You can add multiple instances to a replication group as needed and then sync data between them.

Notes on Availability

Cross-Region Disaster Recovery

A master instance and a read-only instance can be added to a replication group to set up a cross-region disaster recovery system. However, the system will not automatically perform failover, which can only be manually performed in the console or through TencentCloud API. For detailed directions, see Switching Instance Role.

Replication exceptions

No matter whether a replication group has one or multiple master instances, when replication is interrupted, the system will not set them as read-only instances or perform other operations; instead, it will automatically resume the replay of incremental logs after instance recovery. To ensure data consistency, we recommend you configure alarms for replication exceptions and set master instances as read-only instances when a replication exception (such as replication interruption) occurs.

Related APIs

API	Description
DescribeReplicationGroup	Queries a replication group
ChangeMasterInstance	Switches with master instance in a replication group
ChangeInstanceRole	Modifies the role of an instance in a replication group

Managing Global Replication Group

Last updated：2023-03-21 16:18:29

You can manage a created replication group as needed in your actual Ops use cases. Specifically, you can set read-only instances as master instances, switch the roles of master and read-only instances, remove instances, or delete the replication group.

Prerequisites

You have created a global replication group and it is in **Running** status. For more information, see Creating Global Replication Group.

You have added an instance to the replication group and the instance is in **Running** status. For more information, see Creating Global Replication Group.

Setting a read-only instance as the master instance

You can set a read-only instance in a replication group as a master instance for data writes. This change only updates the instance configuration without causing data migration. The entire process takes approximately three minutes.

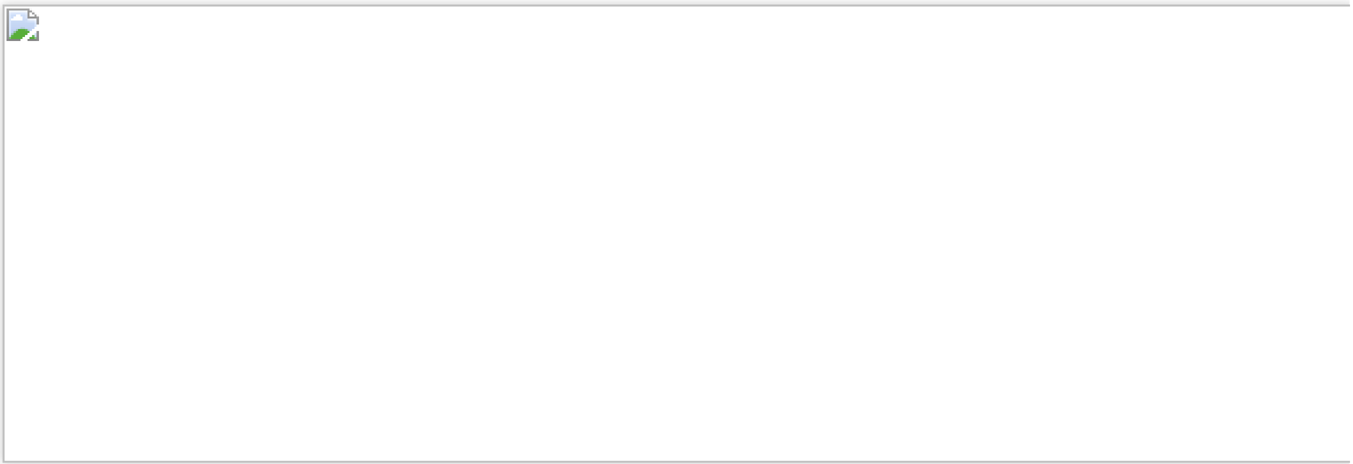
For detailed directions, see the following steps:

1. In the replication group list, click

before the name of the target replication group to show its instance list.
2. Find the target read-only instance and click **Set to Master** in its **Operation** column.

Notes

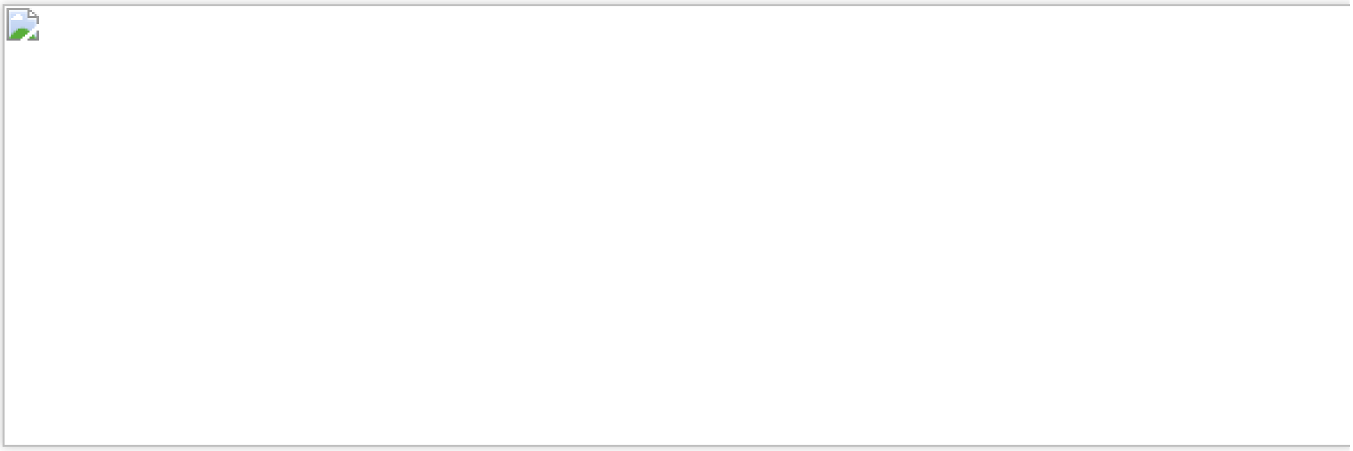
You can set a ready-only instance as a master instance only if there are at least two read-only instance replicas in the replication group.



3. Read the prompt carefully in the **Set to Master** pop-up window and click **OK**.
In the instance list of the replication group, the **instance status** changes to **switching instance role**, and the **instance role** will change to **master instance** after the switch.

Setting the master instance as a read-only instance

You can also set a master instance added to a replication group to a read-only instance as detailed below.



Switching the instance role

You can switch the master instance only in the disaster recovery use case; that is, there is only one master instance and one read-only instance in the replication group. Usually, the replication group will become inaccessible for one minute during the switch.

Notes

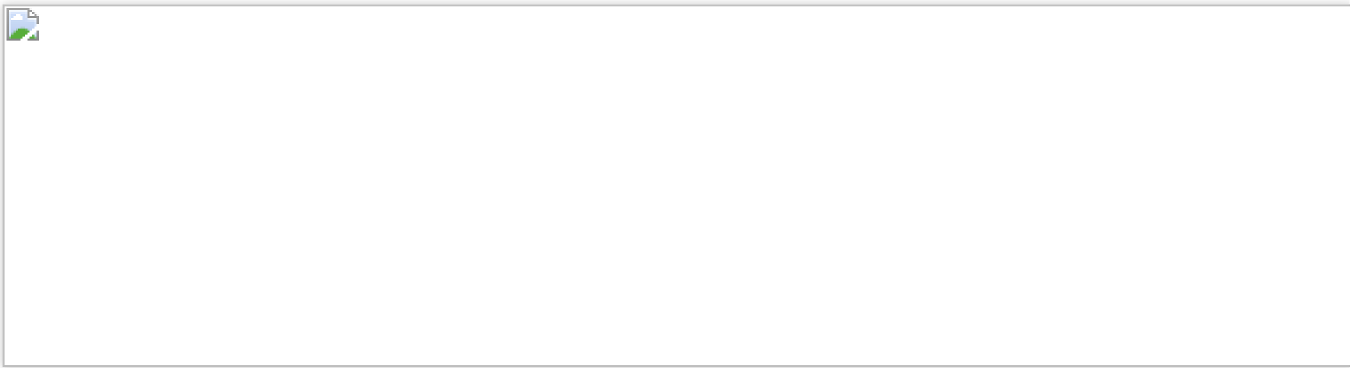
When data is synced to the new master instance (the original read-only instance), it is still available for read access.

See the following steps for how to switch the master instance:

1. In the replication group list, click

before the name of the target replication group to show its instance list.

2. Find the target read-only instance and click **Switch with Master Instance** in its **Operation** column.



3. Read the prompt carefully in the **Switch with Master Instance** pop-up window and click **OK**.

4. Both the master and read-only instances in the replication group enter the **switching with master instance** status. Wait for the switch to complete and then you can see that the original master instance is read-only, while the original read-only instance is now the master instance.

Removing an instance from a replication group

You can remove an instance from the replication group. The removed instance will stop syncing data from other instances in the replication group.

See the following steps for how to remove an instance from a replication group:

1. In the replication group list, click

before the name of the target replication group to show its instance list.

2. Find the target instance and click **Remove from Replication Group** in its **Operation** column.

Notes

You cannot remove an instance if it is the only master instance in the replication group where read-only instances exist.

3. In the **Remove from Replication Group** pop-up window, confirm the instance information and click **OK**.

If the instance to be removed is a master instance, you need to select a **removal mode**.

Remove now: In this mode, the system disconnects data sync within the replication group immediately, without waiting for other nodes to complete data sync from this master instance. Some data may be lost.

Upon data sync completion: In this mode, the system sets the master instance as read-only, waits for other nodes in the replication group to complete data sync from it, disconnects the replication between it and all other nodes in the replication group, removes it from the replication group, cancels its read-only status, and then enables its write access.

4. In the instance list of the replication group, the **instance status** changes to **Disassociating replication group**. Wait for the removal to complete.

Deleting a replication group

You need to remove all instances in a replication group before you can delete it.

1. In the replication group list, select the target replication group and click **Delete Replication Group** in its **Operation** column.

2. In the **Delete Replication Group** pop-up window, confirm the replication group information and click **OK**.

Related APIs

API	Description
DescribeReplicationGroup	Queries a replication group
ChangeMasterInstance	Switches with master instance in a replication group
ChangeInstanceRole	Modifies the role of an instance in a replication group

Monitoring Metrics

Last updated：2024-04-22 15:27:29

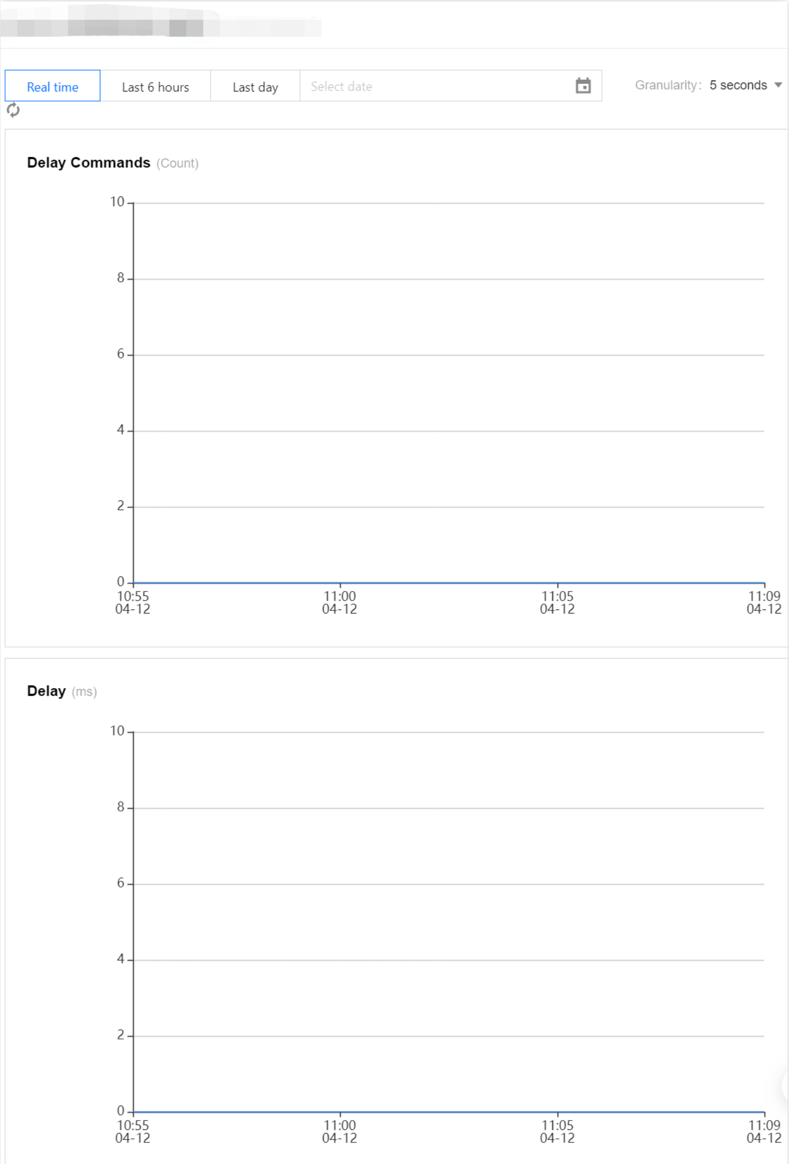
encentDB for Redis supports graphical monitoring of the command count deviation during the global replication process between the primary instance and disaster recovery instances, as well as the synchronization latency in remote synchronization.

Monitoring Metrics

Metric	Definition	Unit
Delay Commands	During the global replication process, the command count deviation between the primary instance and disaster recovery instances in remote synchronization.	Count
Latency	During the global replication process, the synchronization latency between the primary instance and disaster recovery instances in remote synchronization.	ms

Viewing the Monitoring View

1. Sign in to the TencentDB for Redis console.
2. In the left side bar, choose **Global Replication**.
3. In the right global replication list, find the row where the replication group you want to view is located.
4. In the **Replication Group Status** column, click 📊 to slide out the monitoring view of monitoring metrics on the right, as shown below.


Real time
Last 6 hours
Last day
Select date
Granularity: 5 seconds
Delay Commands (Count)
10
8
6
4
2
0
10:55 04-12
11:00 04-12
11:05 04-12
11:09 04-12
Delay (ms)
10
8
6
4
2
0
10:55 04-12
11:00 04-12
11:05 04-12
11:09 04-12

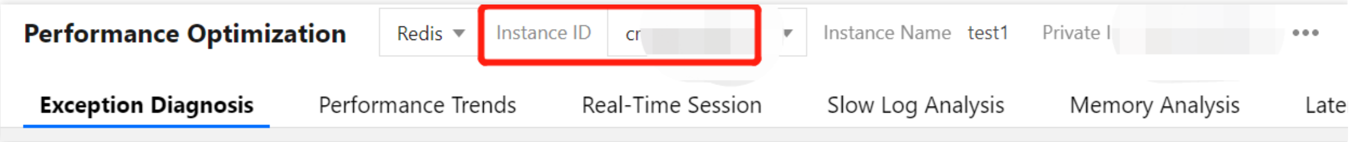
Performance Optimization Overview

Last updated：2022-06-24 16:34:16

TencentDB for Redis is connected to the performance optimization feature of DBbrain. The feature monitors and diagnoses database instance exceptions in real time, automatically generates health reports, and gives expert optimization suggestions. This helps you stay on top of the running status of the current database, quickly locate and troubleshoot issues, and promptly optimize the database performance.

Viewing Performance Optimization

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



Performance Optimization

Redis ▾ Instance ID cr[redacted] ▾ Instance Name test1 Private I[redacted] •••

4. View and analyze the diagnosis data of the instance.

Monitoring Type	Description
Exception Diagnosis	Performs real-time performance monitoring and health inspections on the database and gives diagnosis prompts and optimization suggestions for failures.
Performance Trends	Monitors the performance metrics by instance, Redis node, and proxy node.
Real-Time Session	Collects the information of database client sessions in real time, such as the sources and number of sessions as well as the number of active sessions.
Slow Log Analysis	Monitors the number and duration of slow queries by instance and proxy in real time.
Memory Analysis	Monitors and analyzes the memory utilization of big keys in the database.
Latency Analysis	Performs statistical analysis on the latency of all request commands to the database, accurate down to the millisecond.

Command Word Analysis	Performs statistical analysis on the number and latency of access requests for database command words.
Hot Key Analysis	Performs statistical analysis on frequently accessed hot keys.

Exception Diagnosis

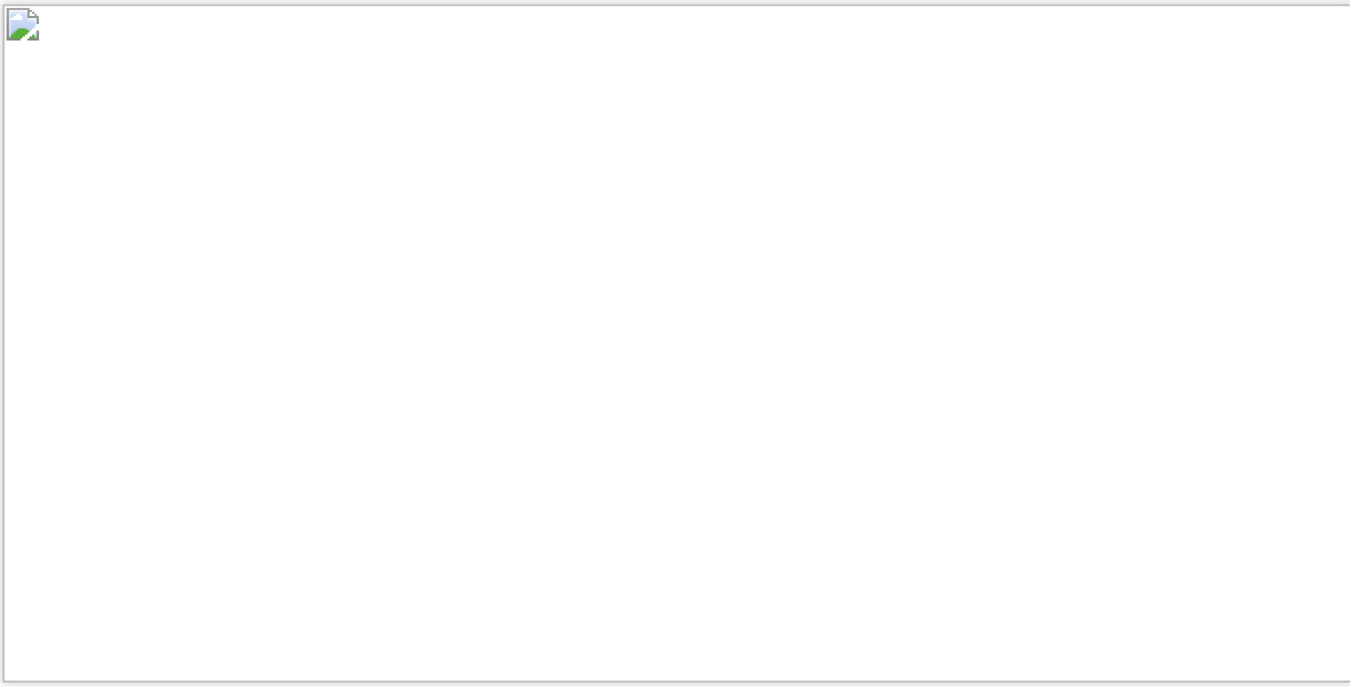
Last updated：2023-03-14 15:46:10

Description

Based on the metric data collected by smart monitoring, the exception diagnosis feature diagnoses and analyzes database exceptions around the clock. For database performance problems, it leverages the analysis results of the SQL optimization, performance analysis, and rule engines to present the current and historical instance health scores and risk levels, with regard to specification configuration, SQL analysis, business logic, and usage rationality. This gives you a visual, clear, and quick picture of the running status of your database instances.

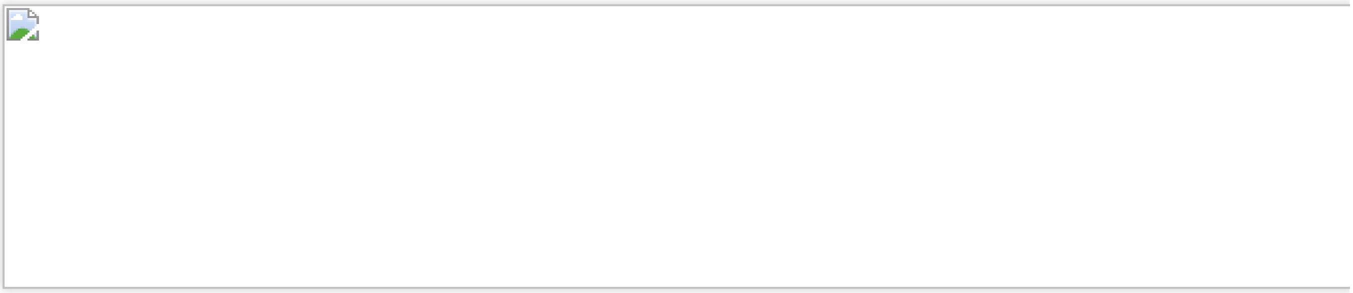
Viewing performance optimization

1. Log in to the TencentDB for Redis console and select **Performance Optimization** on the left sidebar.
2. On the **Performance Optimization** page of **DBbrain**, select **Exception Diagnosis** to view the exception diagnosis details.



Overview

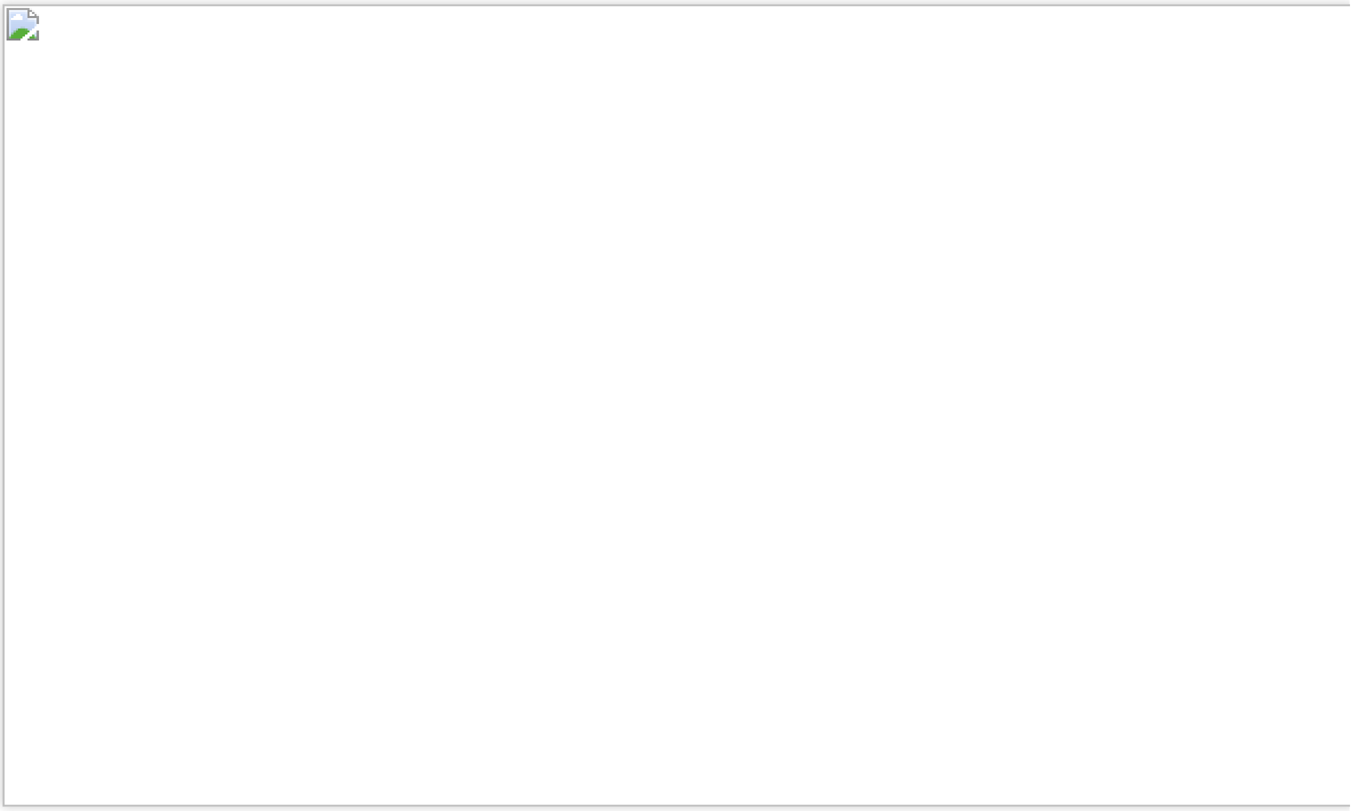
In the **Overview** section, the TencentDB for Redis instance's health score trend and risk level as diagnosed through health inspection in the last hour are displayed by default. Click **Historical** in the top-right corner and select a past time period to view the specific health score and inspection diagnosis result.



Below the **Overview** section, your instance's health score and node alarms and exceptions at the current time point are displayed.

Your database instance's current health score and the statistics of the current resource monitoring metrics are displayed, including the utilizations of CPU, memory, connection, inbound traffic, and outbound traffic, as well as read request hit rate, so you can quickly check the database resource usage. Click **Details** below **Health Score** to enter the **Health Report** tab, where you can view the health score details. For more information, see Health Reports.

In addition, your database's system distribution architecture diagram is displayed. You can view the proxy, master, and replica nodes and the numbers of their respective alarms and exceptions. You can also hover over a node to view the monitoring data of its key metrics. For more information on monitoring metrics, see Performance Trends.



Diagnosis prompt

In the **Diagnosis Prompt** section, the diagnosis data of the database health inspection in the last 3 hours is displayed.

Risk Distribution: The proportions of the risk levels as diagnosed through the database health inspection in the last 3 hours are displayed from high to low, including **Critical**, **Serious**, **Alarm**, and **Note**. If the proportions of all risk levels are 0%, the database is **Healthy**.

Diagnosis Details: The details of the database risk level are collected once every 10 minutes. You can hover over any exception alarm to view, ignore, or unignore alarms of the same type.

Parameter	Description
Level	Diagnosed risk level, including **Critical**, **Serious**, **Alarm**, **Note**, and **Healthy**, from high to low.
Start Time	Start time of the health inspection. A health inspection is initiated once every 10 minutes.
Diagnosis items	Diagnosis type of the risk level, such as health inspection and incorrect command.
Duration	Duration of the risk level.
View	Click **View** to view the details of exceptions identified during the health inspection:
Ignore	Click **Ignore** and then click **OK** in the **confirmation** window, and exception alarms triggered by the same cause will be ignored.
Unignore	Click **Display Ignored Items** in the top-right corner of the **Diagnosis Details** section. In the diagnosis details list, hover over an ignored item and click **Unignore**, and all exception alarms triggered by the same cause will be unignored.

Performance Trends

Last updated：2023-05-23 10:41:50

Feature Overview

The performance trends feature monitors the key performance metrics of database instances, Redis nodes, and proxy nodes in real time, including CPU, memory, key information, network usage, network utilization, requests, and responses. It collects monitoring data at the second granularity, dynamically displays the change trends of metrics as well as their maximum, minimum, and average values in tables graphically. You can compare the metrics of multiple nodes and different time periods and zoom in on and drag the monitoring view as needed.
The performance trends feature can meet your needs in various routine Ops and troubleshooting scenarios of database instances thanks to its strong statistical analysis capabilities, diverse visual options, and extremely high real-timeness. These capacities let you quickly get a holistic picture of database performance to minimize risks.

Monitoring Metrics

Supported monitoring metrics are displayed in three dimensions: instance, Redis node, and proxy node.

Instance

Category	Description	Metric	Unit	Meaning
CPU	CPU Utilization	cpu_util	%	Average CPU utilization
	Max CPU Utilization of Node	cpu_max_util	%	The maximum CPU utilization of a node (shard or replica) in an instance
Memory information	Memory Usage	mem_used	MB	Actually used memory capacity, including the capacity for data and cache
	Memory utilization	mem_util	%	The ratio of the actually used memory to the requested total memory
	Max Memory Utilization of Node	mem_max_util	%	The maximum among all node (shard or replica) memory utilizations in an instance

Key information	Total Keys	keys	Pcs	The total number of keys (level-1 keys) in instance storage
	Expired Keys	expired	Pcs	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	Pcs	The number of keys evicted in a time window, which is equal to the value of `evicted_keys` output by the `info` command.
Network usage	Number of connections	connections	Pcs	The number of TCP connections to an instance
	Inbound Traffic	in_flow	MB/s	Private network inbound traffic
	Outbound traffic	out_flow	MB/s	Private network outbound traffic
Network usage rates	Connection Utilization	connections_util	%	The ratio of the number of TCP connections to the maximum number of connections
	Inbound Traffic Utilization	in_bandwidth_util	%	The ratio of the actually used private inbound traffic to the maximum traffic
	Outbound Traffic Utilization	out_bandwidth_util	%	The ratio of the actually used private outbound traffic to the maximum traffic
Request	Total Requests	commands	Times/sec	Queries per second
	Read request	cmd_read	Times/sec	The number of read command executions per second
	Write request	cmd_write	Times/sec	The number of write command executions per second
	Other Requests	cmd_other	Times/sec	The number of command (excluding write and read commands) executions per second
	Big Value Requests	cmd_big_value	Times/sec	The number of executions of requests larger than 32 KB per second
	Key Requests	cmd_key_count	Times/sec	The number of keys requested per

				second
	Mget Executions	cmd_cmget	Times/sec	The number of requests made through MGET per second
Response	Checking slow queries	cmd_slow	times	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read Request Hits	cmd_hits	times	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	times	The number of keys unsuccessfully requested by read commands, which is equal to the value of the `keyspace_misses` metric output by the `info` command.
	Read Request Hit Rate	cmd_hits_ratio	%	Key hits/(key hits + key misses). This metric can reflect the situation of cache miss. When the access request quantity is 0, the value of this metric will be null.
Execution Error	Execution Error	cmd_err	times	The number of command execution errors. For example, the command does not exist, parameters are incorrect, etc.
Latency	Average Execution Latency	latency_avg	ms	The average execution latency between the proxy and the Redis server
	Max Execution Latency	latency_max	ms	The maximum execution latency between the proxy and the Redis server
	P99 execution latency	latency_p99	ms	The P99 execution latency between the proxy and the Redis server
	Average Read Latency	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For more

				information on read command types, see Command types.
	Average Write Latency	latency_write	ms	The average execution latency of write command between the proxy and the Redis server. For more information on write command types, see Command types.
	Average Latency of Other Commands	latency_other	ms	The average execution latency of commands (excluding write and read commands) between the proxy and the Redis server

Redis node

Category	Description	Metric	Unit	Meaning
CPU	CPU Utilization	cpu_util	%	Average CPU utilization
Network usage	Number of connections	connections	Pcs	The number of connections between the proxy and a node
	Connection Utilization	connections_util	%	The connection utilization of the node
Memory information	Memory Usage	mem_used	MB	Actually used memory capacity, including the capacity for data and cache
	Memory utilization	mem_util	%	The ratio of the actually used memory to the requested total memory
Key information	Total Keys	keys	Pcs	The total number of keys (level-1 keys) in instance storage
	Expired Keys	expired	Pcs	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	Pcs	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	Replication Delay	repl_delay	Byte	The command delay between the replica node and the master node
Request	Total Requests	commands	Times/sec	Queries per second
	Read request	cmd_read	Times/sec	Number of read command executions. For more information on read command types, see Command types.
	Write request	cmd_write	Times/sec	Number of write command executions. For more information on write command types, see Command types.
	Other Requests	cmd_other	Times/sec	The number of command (excluding write and read commands) executions per second
Response	Checking slow queries	cmd_slow	times	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read Request Hits	cmd_hits	times	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	times	The number of keys unsuccessfully requested by read commands, which is equal to the value of the `keyspace_misses` metric output by the `info` command.
	Read Request Hit Rate	cmd_hits_ratio	%	Key hits/(key hits + key misses). This metric can reflect the situation of cache miss.

Proxy node

Category	Description	Metric	Unit	Meaning
CPU	CPU Utilization	cpu_util	%	CPU utilization

				of the proxy
Request	Total Requests	proxy_commands	Times/sec	The number of commands executed by the proxy
	Key Requests	cmd_key_count	Counts/sec	The number of keys accessed by a command per second
	Mget Executions	cmd_mget	Times/sec	The number of MGET commands executed per second
	Execution Error	cmd_err	Times/sec	The number of command execution errors in the proxy. For example, the command does not exist, parameters are incorrect, etc.
	Big Value Requests	cmd_big_value	Times/sec	The number of executions of requests larger than 32 KB per second
Traffic	Inbound Traffic	in_flow	MB/s	Private network inbound traffic
	Outbound traffic	out_flow	MB/s	Private network outbound traffic
Network usage	Number of connections	connections	Pcs	The number of TCP connections to an instance
	Connections per sec	client_connections_received_per_second	Pcs	The number of TCP

				connections established per second
	Disconnections per sec	client_connections_closed_per_second	Pcs	The number of TCP connections closed per second
	Abnormal connections per second	client_connections_aborted_per_second	Pcs	The number of TCP connections aborted per second
Network usage rates	Connection Utilization	connections_util	%	The ratio of the number of TCP connections to the maximum number of connections
	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	times	The number of times inbound traffic triggers a traffic limit
	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	times	The number of times outbound traffic triggers a traffic limit
Latency	Average	latency_avg	ms	The average

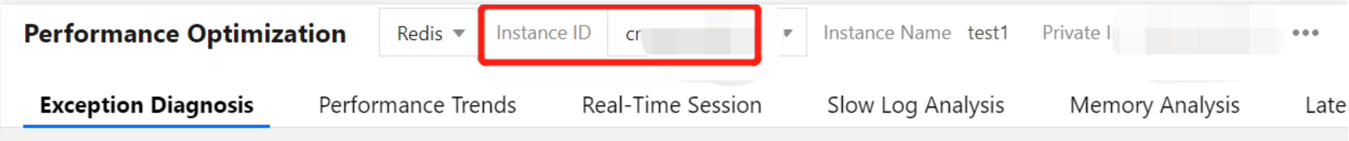
	Execution Latency			execution latency between the proxy and the Redis server
	Max Execution Latency	latency_max	ms	The maximum execution latency between the proxy and the Redis server
	P99 execution latency	latency_p99	ms	The P99 execution latency between the proxy and the Redis server
	Average Read Latency	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For more information on read command types, see Command types.
	Average Write Latency	latency_write	ms	The average execution latency of write command between the proxy and the Redis server. For more information on write command types, see

				Command types.
	Average Latency of Other Commands	latency_other	ms	The average execution latency of commands (excluding write and read commands) between the proxy and the Redis server

Viewing Monitoring Data

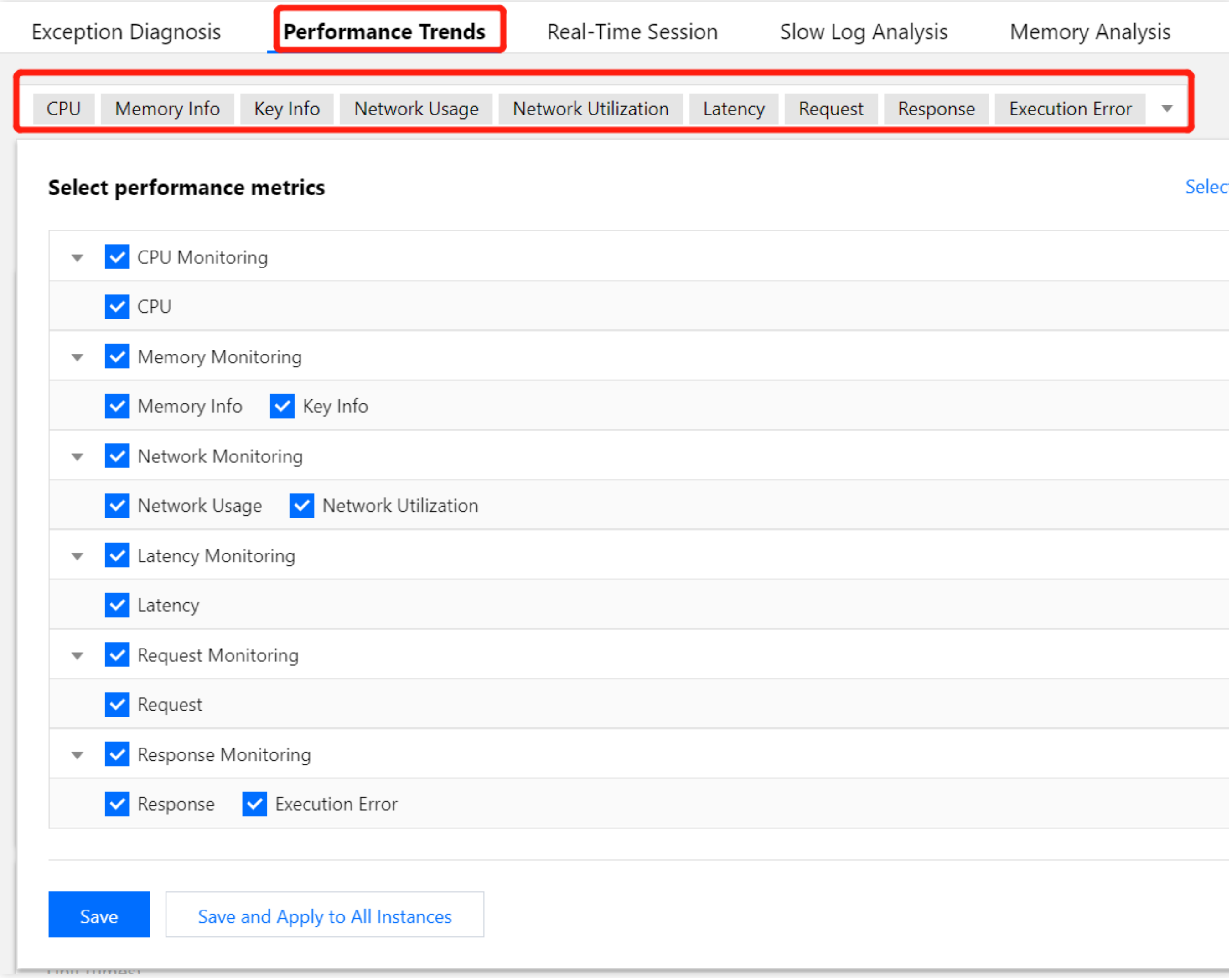
Step 1. Select monitoring metrics

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.


Performance Optimization Redis Instance ID cr[redacted] Instance Name test1 Private I[redacted]

4. Click the **Performance Trends** tab, select the target performance metrics in the metric category drop-down list, and click **Save**.

To apply the selected performance metrics to all the TencentDB for Redis instances under your account, click **Save and Apply to All Instances** as shown below:

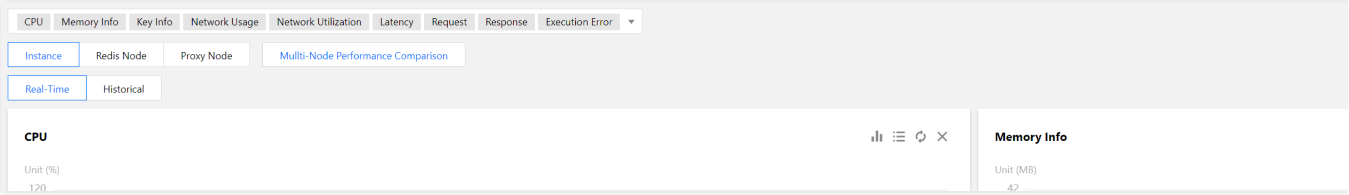


Select performance metrics

☑ CPU Monitoring
☑ CPU
☑ Memory Monitoring
☑ Memory Info ☑ Key Info
☑ Network Monitoring
☑ Network Usage ☑ Network Utilization
☑ Latency Monitoring
☑ Latency
☑ Request Monitoring
☑ Request
☑ Response Monitoring
☑ Response ☑ Execution Error

Step 2. Set the collection granularity

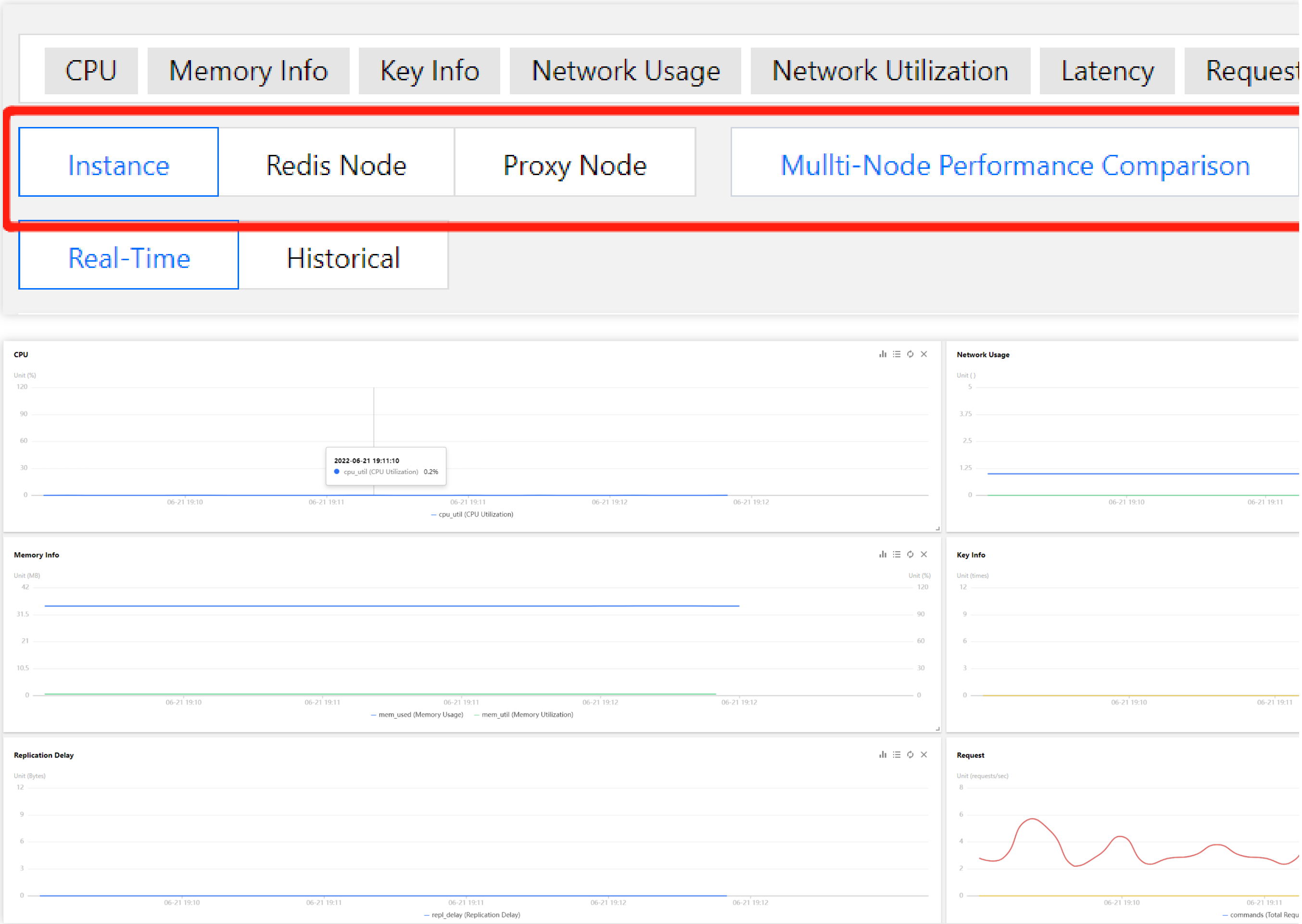
In the top-right corner of the **Performance Trends** tab, set the collection granularity of the monitoring data in the drop-down list on the right of **Auto-Refresh** to **5s**, **15s**, or **30s** as shown below:



Step 3. View the change trends of the monitoring metrics

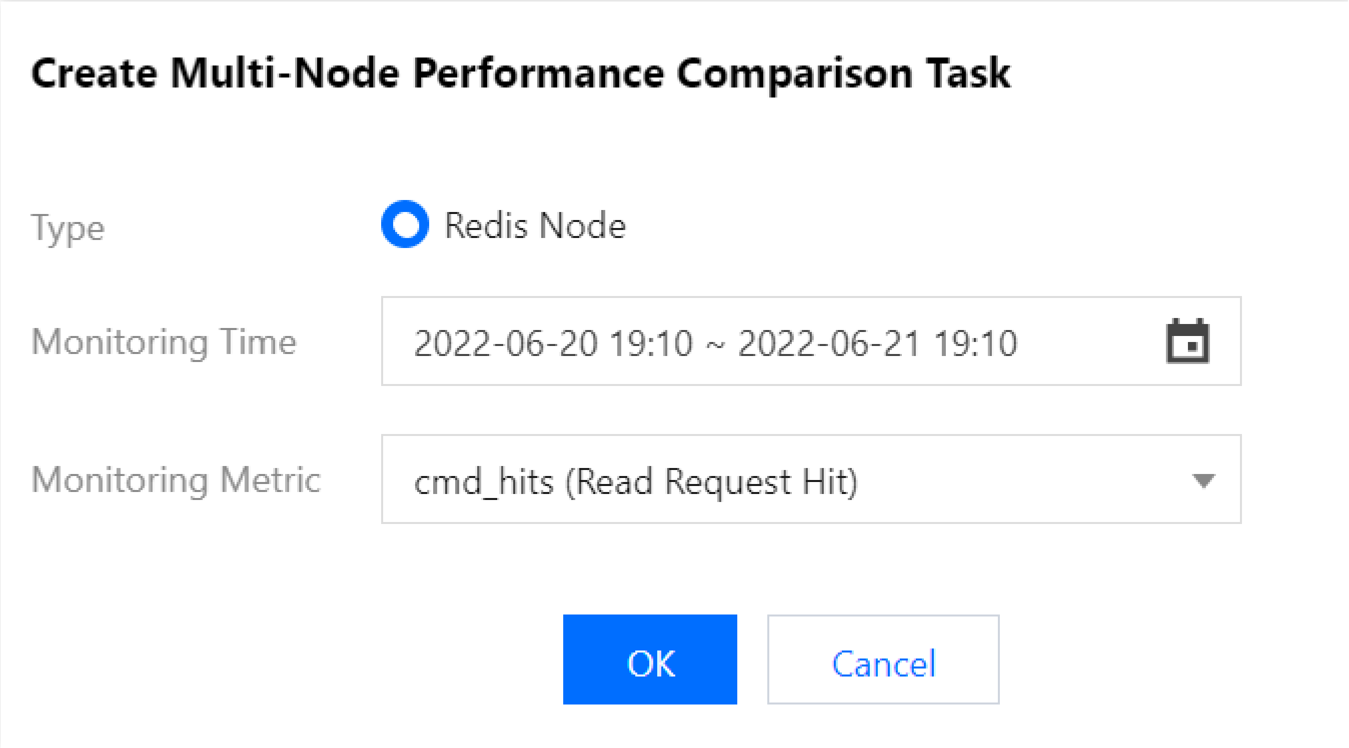
Viewing monitoring metrics in different dimensions

Below the metric categories on the **Performance Trends** tab, you can view the monitoring metric data by instance, Redis node, and proxy node as needed.



Comparing the performance metrics of multiple nodes

1. On the **Performance Trends** tab, click **Multi-Node Performance Comparison**.
2. In the **Multi-Node Performance Comparison** panel, click **Create Multi-Node Performance Comparison Task**.
3. In the **Create Multi-Node Performance Comparison Task** window, click 📅 in the selection box next to **Monitoring Time** to select the monitoring time period, select the target monitoring metric in the **Monitoring Metric** drop-down list, and click **OK**.



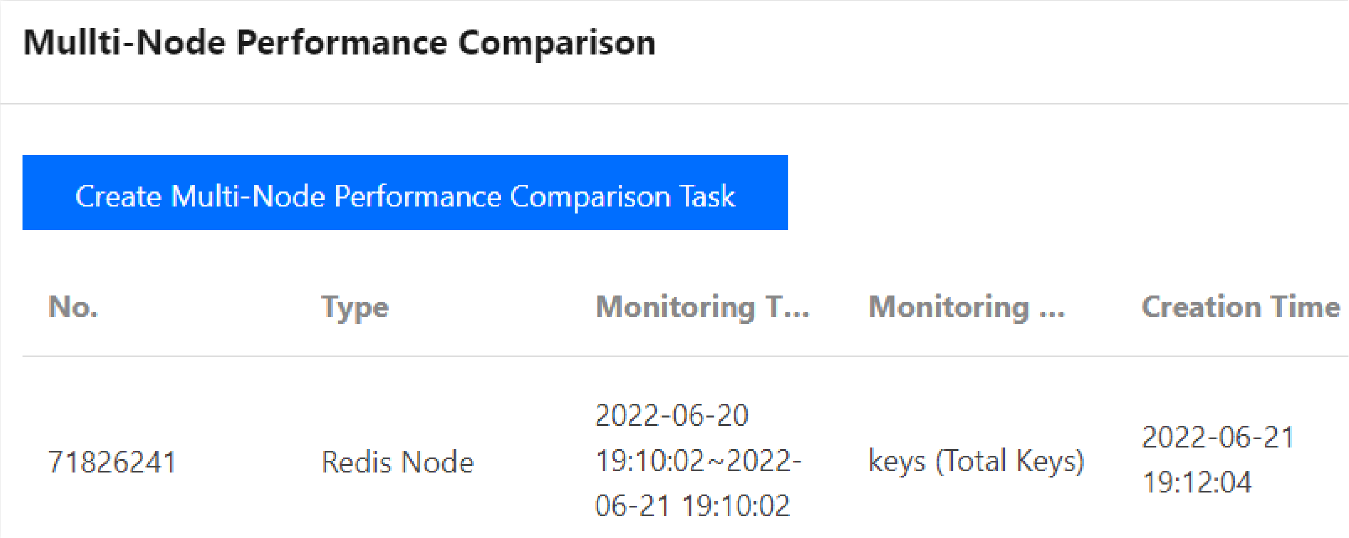
Create Multi-Node Performance Comparison Task

Type: ● Redis Node

Monitoring Time: 2022-06-20 19:10 ~ 2022-06-21 19:10

Monitoring Metric: cmd_hits (Read Request Hit)

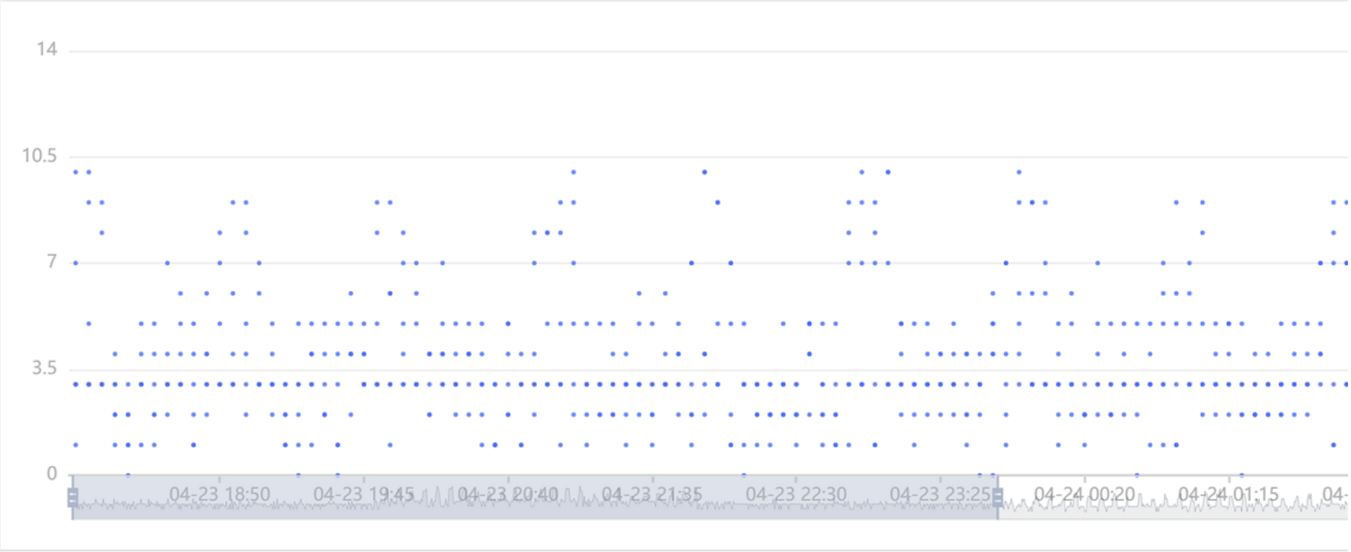
4. Wait for the **Status** to become **Successful** in the task list in the **Multi-Node Performance Comparison** panel.



Mullti-Node Performance Comparison

No.	Type	Monitoring T...	Monitoring ...	Creation Time
71826241	Redis Node	2022-06-20 19:10:02~2022-06-21 19:10:02	keys (Total Keys)	2022-06-21 19:12:04

5. Click **View** in the **Operation** column to view the comparison data of all Redis nodes. The connections metric is used as an example as shown below:



Switching between real-time and historical views

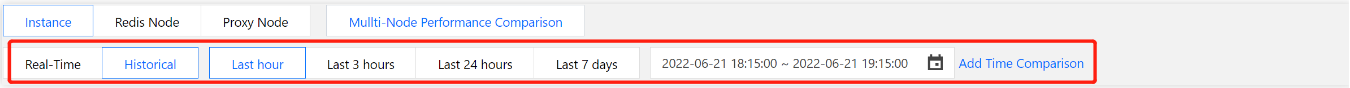
On the **Performance Trends** tab, the real-time monitoring data is displayed by default.

In routine Ops monitoring, database instance metrics can be monitored in real time.

When you need to locate exceptions, you can click **Historical** to analyze the monitoring data in a past time period.

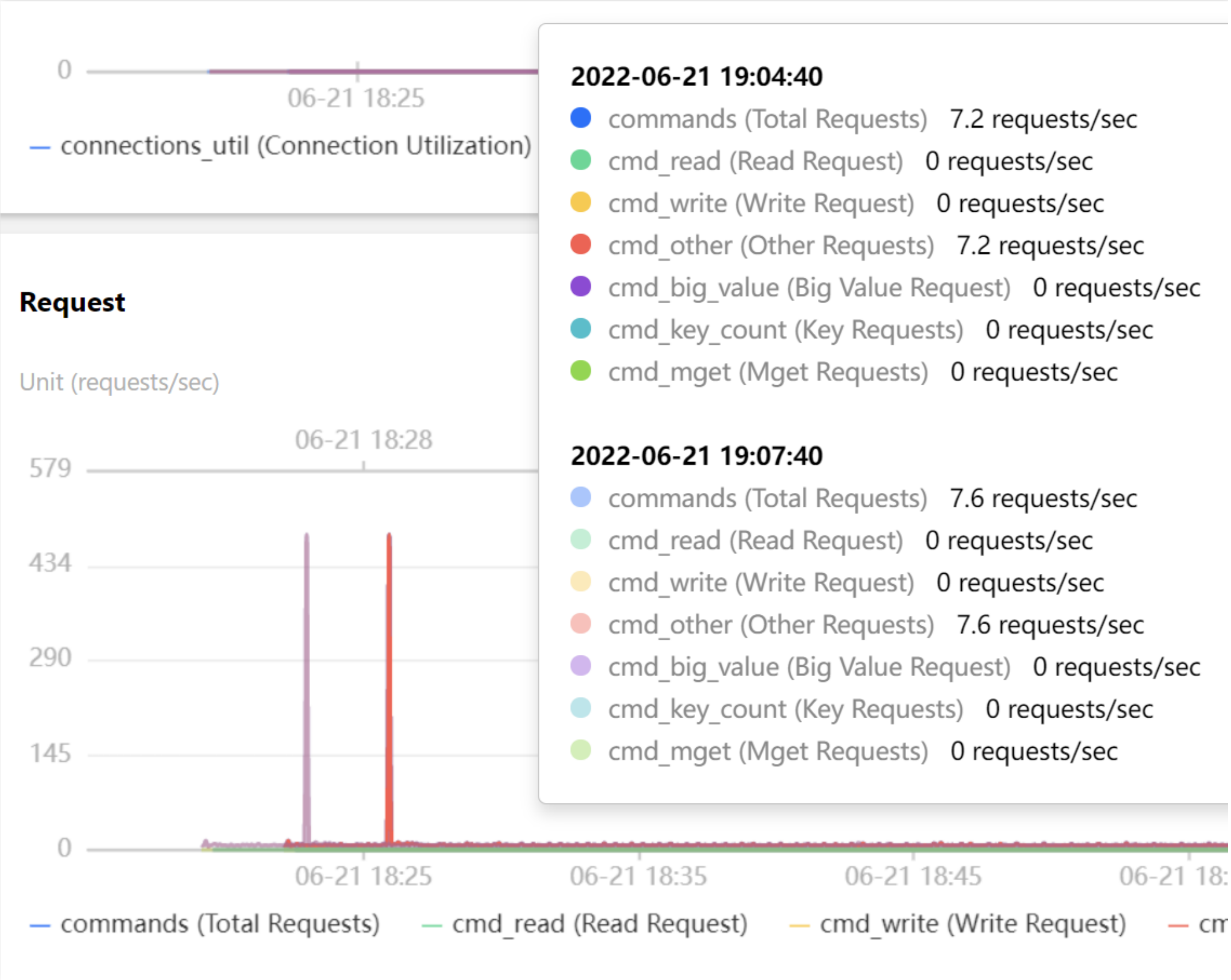
The monitoring data in the last 1 hour, 3 hours, and 7 days can be viewed.

Click 📅 to view the monitoring data in any time period in the last 30 days.



Comparing performance metrics in different time periods

1. On the **Performance Trends** tab, click **Historical** and then click **Add Time Comparison**.
2. In the time selection box, select two time periods for comparison.
3. Select the target monitoring metrics and hover over the change trend in the monitoring view to compare the monitoring data in the two time periods.



2022-06-21 19:04:40

- ● commands (Total Requests) 7.2 requests/sec
- ● cmd_read (Read Request) 0 requests/sec
- ● cmd_write (Write Request) 0 requests/sec
- ● cmd_other (Other Requests) 7.2 requests/sec
- ● cmd_big_value (Big Value Request) 0 requests/sec
- ● cmd_key_count (Key Requests) 0 requests/sec
- ● cmd_mget (Mget Requests) 0 requests/sec

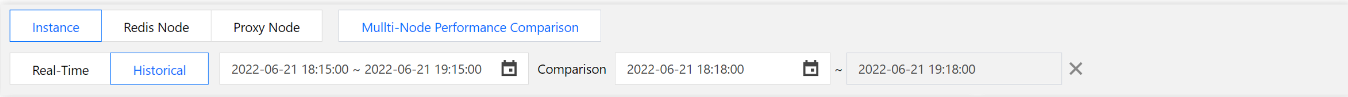
2022-06-21 19:07:40

- ● commands (Total Requests) 7.6 requests/sec
- ● cmd_read (Read Request) 0 requests/sec
- ● cmd_write (Write Request) 0 requests/sec
- ● cmd_other (Other Requests) 7.6 requests/sec
- ● cmd_big_value (Big Value Request) 0 requests/sec
- ● cmd_key_count (Key Requests) 0 requests/sec
- ● cmd_mget (Mget Requests) 0 requests/sec

Displaying monitoring metric data in chart

Click

next to **Show Statistics** as shown below to display the max, min, and average values of each monitoring metric in a table.



2022-06-21 18:15:00 ~ 2022-06-21 19:15:00

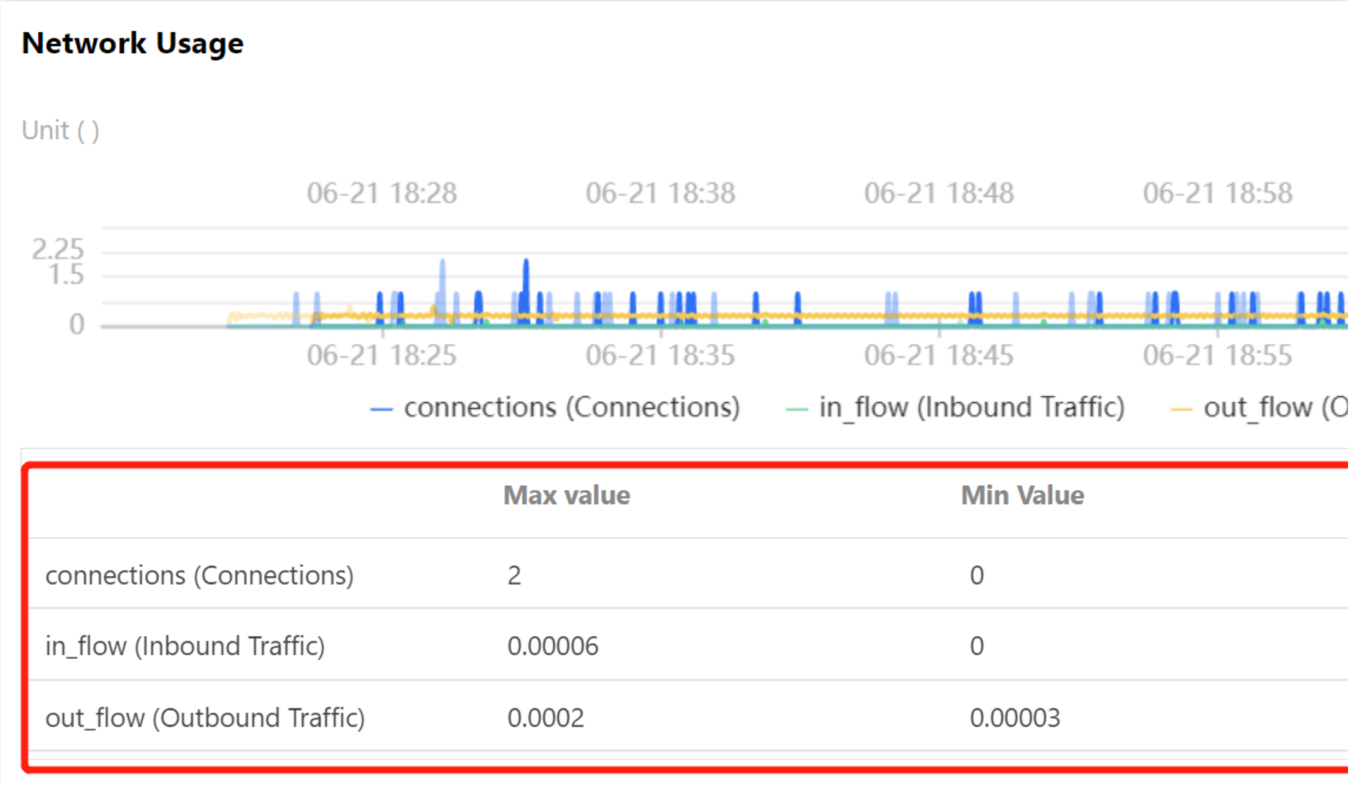
Comparison 2022-06-21 18:18:00 ~ 2022-06-21 19:18:00

Click

☰

in the top-right corner of any monitoring view to display the max, min, and average values of the monitoring metric in a table.

The **Network Usage** metric is used as an example as shown below:

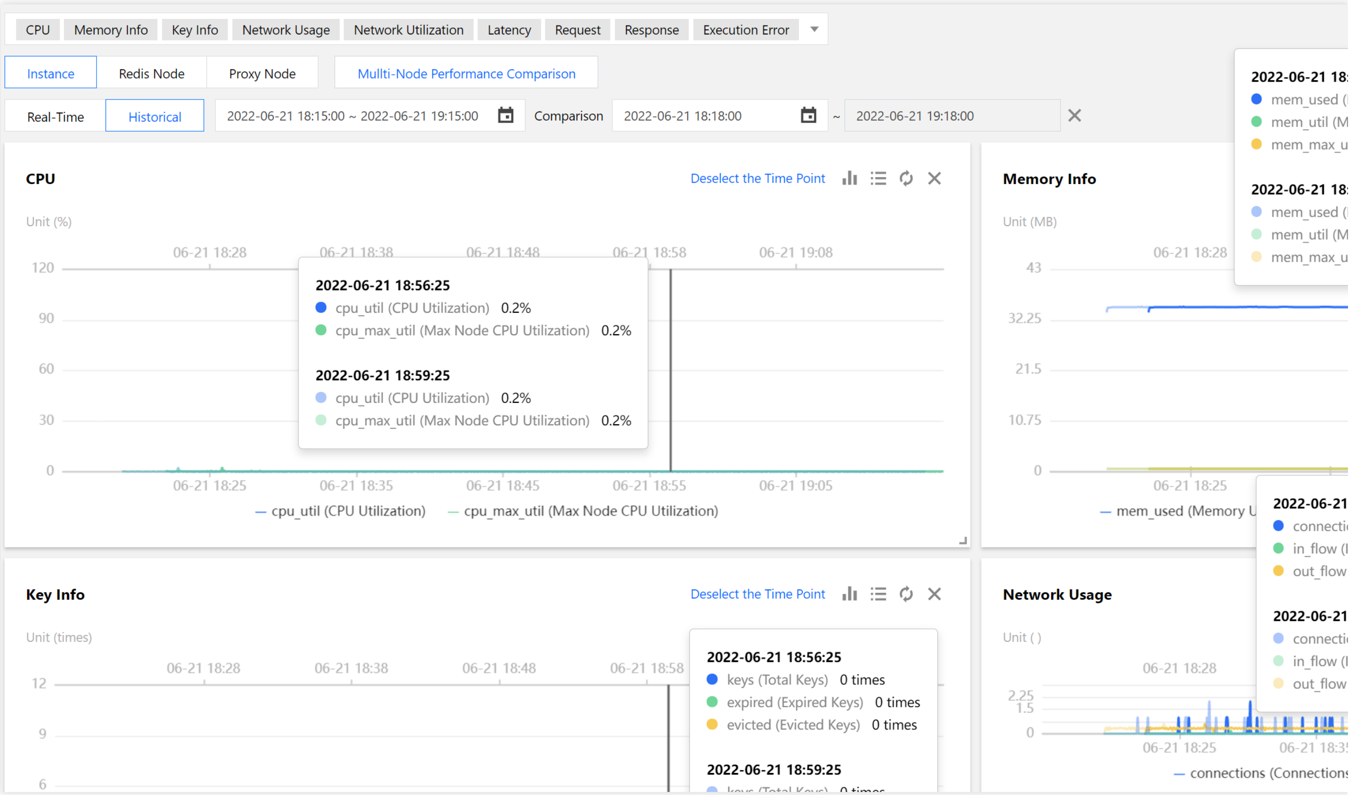


	Max value	Min Value
connections (Connections)	2	0
in_flow (Inbound Traffic)	0.00006	0
out_flow (Outbound Traffic)	0.0002	0.00003

Viewing monitoring data through chart interaction

The **Chart Interaction** feature is suitable for analyzing the data of a monitoring view and its associated monitoring views.

1. In the top-right corner of the **Performance Trends** tab, click next to **Chart Interaction**.
2. In any of the monitoring views to be analyzed, select a time point and click it, and the data at the same time point will be fixed for display in other monitoring views.
3. You can click **Deselect the Time Point** in the top-right corner of the monitoring view to cancel the fixed display.

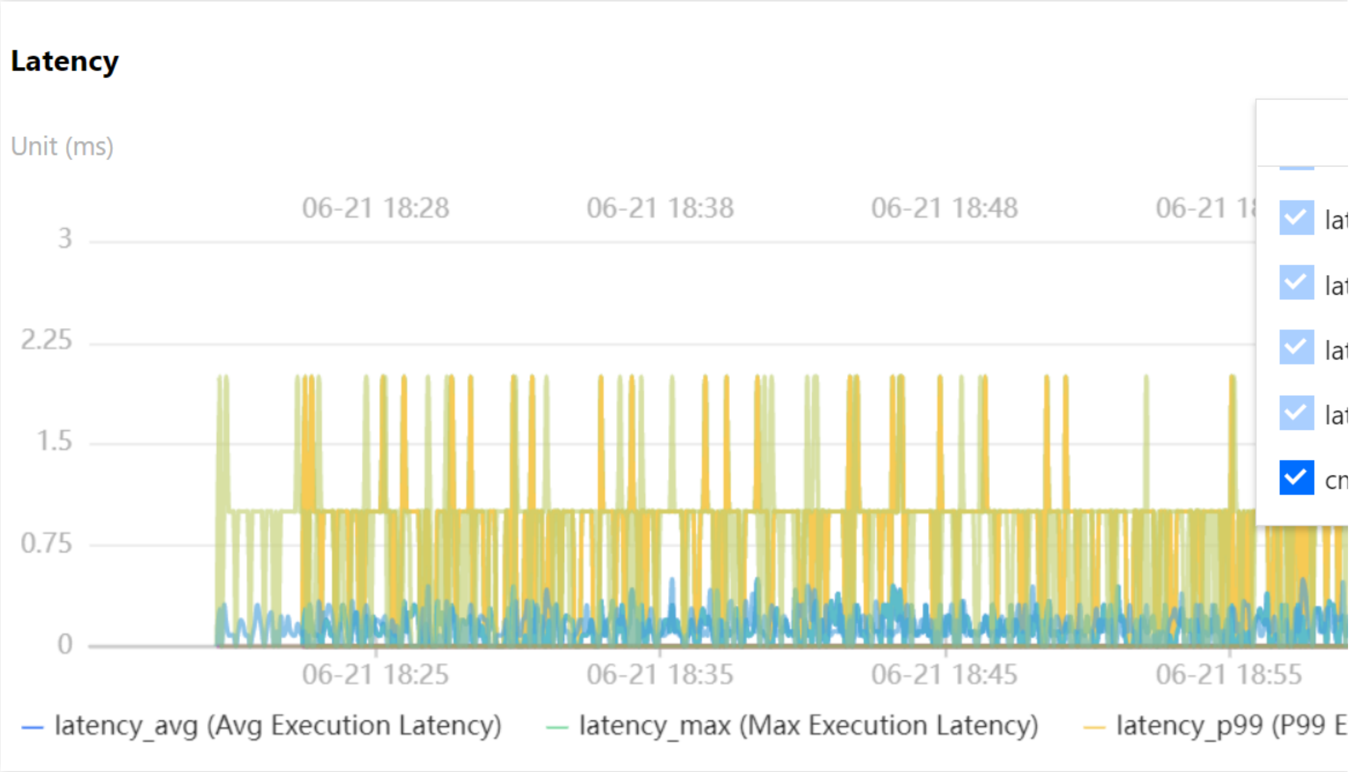


Customizing monitoring metric for comparative analysis

Click

📊

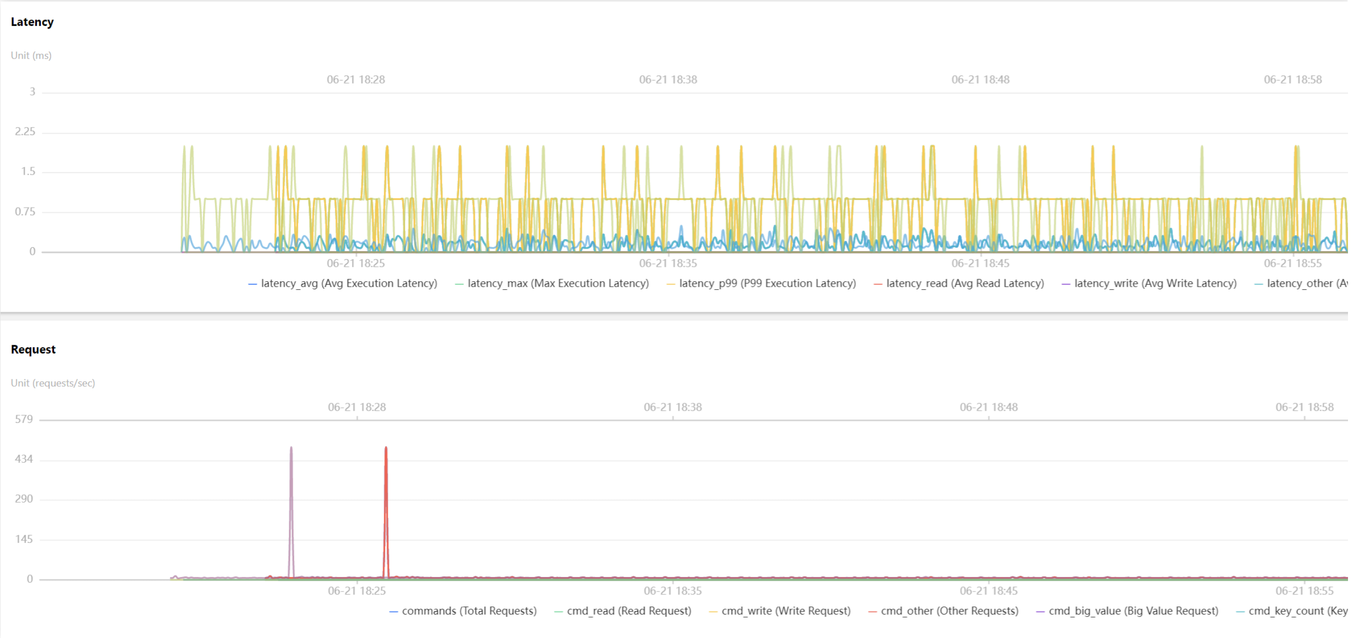
in the top-right corner of any monitoring view to add monitoring metrics of other types for comparative display and analysis.



Switching between one-column and two-column mode of monitoring view

Click

⊞ on the right of **Chart Interaction** in the top-right corner to switch between the one-column and two-column modes. The former is as shown below:



Dragging monitoring view

The monitoring views can be freely dragged to flexible adjust their order for efficient display and analysis.

Zooming in on monitoring view

Drag the icon in the bottom-right corner of any monitoring view to zoom in on the image for clearer display of the metric trends.

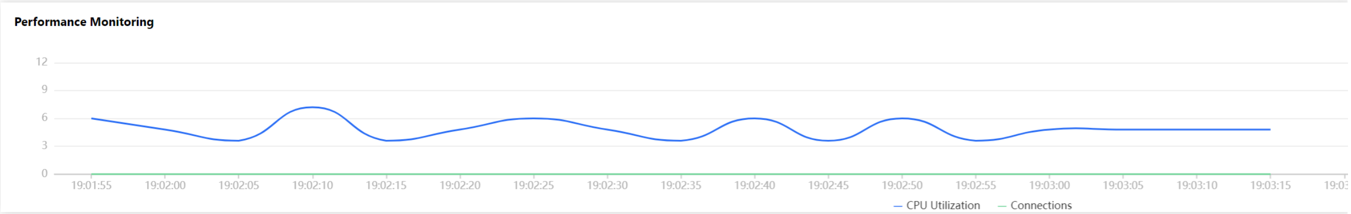
Real-Time Session

Last updated：2023-05-23 10:41:31

Feature Overview

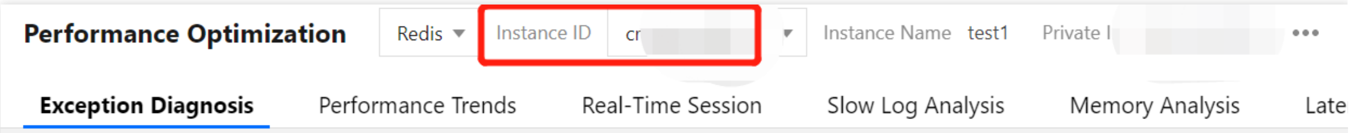
The real-time session feature focuses on the two key metrics of database proxy node CPU utilization and client connection quantity. It dynamically displays their change trends and continuously collects the data of database sessions, access sources, and active connection quantity.

Real-time session allows you to quickly identify the CPU utilization of current sessions and locate challenging logic issues about database session connections.



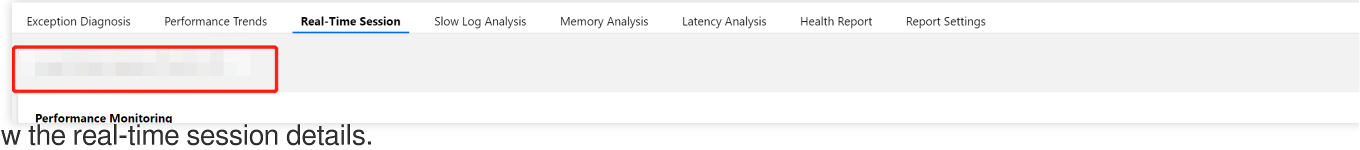
Viewing Real-Time Session Statistics

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



4. Click the **Real-Time Session** tab and select the target **proxy ID** in the drop-down list in the top-left corner of the **Performance Monitoring** view based on the trend of **CPU Utilization** or **Connections**.

In the top-right corner of the **Performance Monitoring** view, select the collection granularity of the monitoring data in the drop-down list next to **Auto-Refresh**, which is **5s** by default and can also be set to **15s** or **30s**.

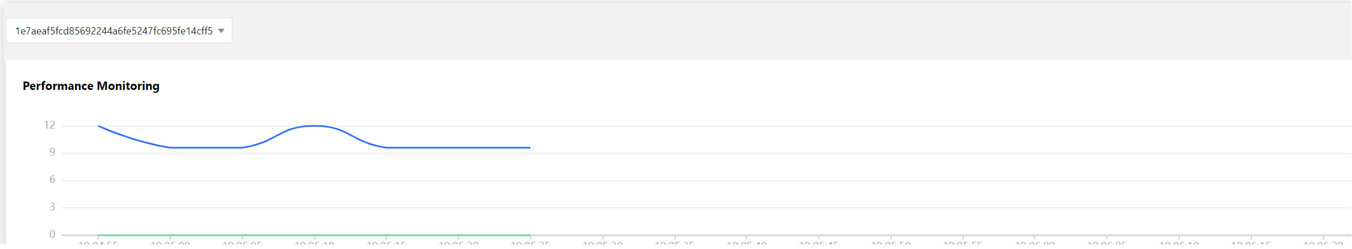


5. View the real-time session details.

- In the **Performance Monitoring** section, you can view the change trends of the current proxy node connections and CPU utilization.
In the **Session Statistics** section, you can view the database's statistics of current access sources, total connections, and active connections.

One-Click Kill

You can click **One-Click Kill** to quickly kill all sessions.



Slow Log Analysis

Last updated：2023-03-14 15:44:00

Overview

A slow query is defined as a query statement that takes more time than the specified value, and the statement is called a slow query statement. The slow log analysis feature performs statistical analysis on the number of slow queries by **instance** and **proxy** and gives expert optimization suggestions to help improve the database performance.

In the instance (Redis database instance) dimension, you can view the CPU utilization, number of slow logs, consumed time statistics by log segment, and information of the entire slow log list.

In the proxy (middleware cluster node) dimension, you can view the proxy's slow log statistics, consumed time statistics by segment, and details of the slow log list.

Viewing the slow log analysis data

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.

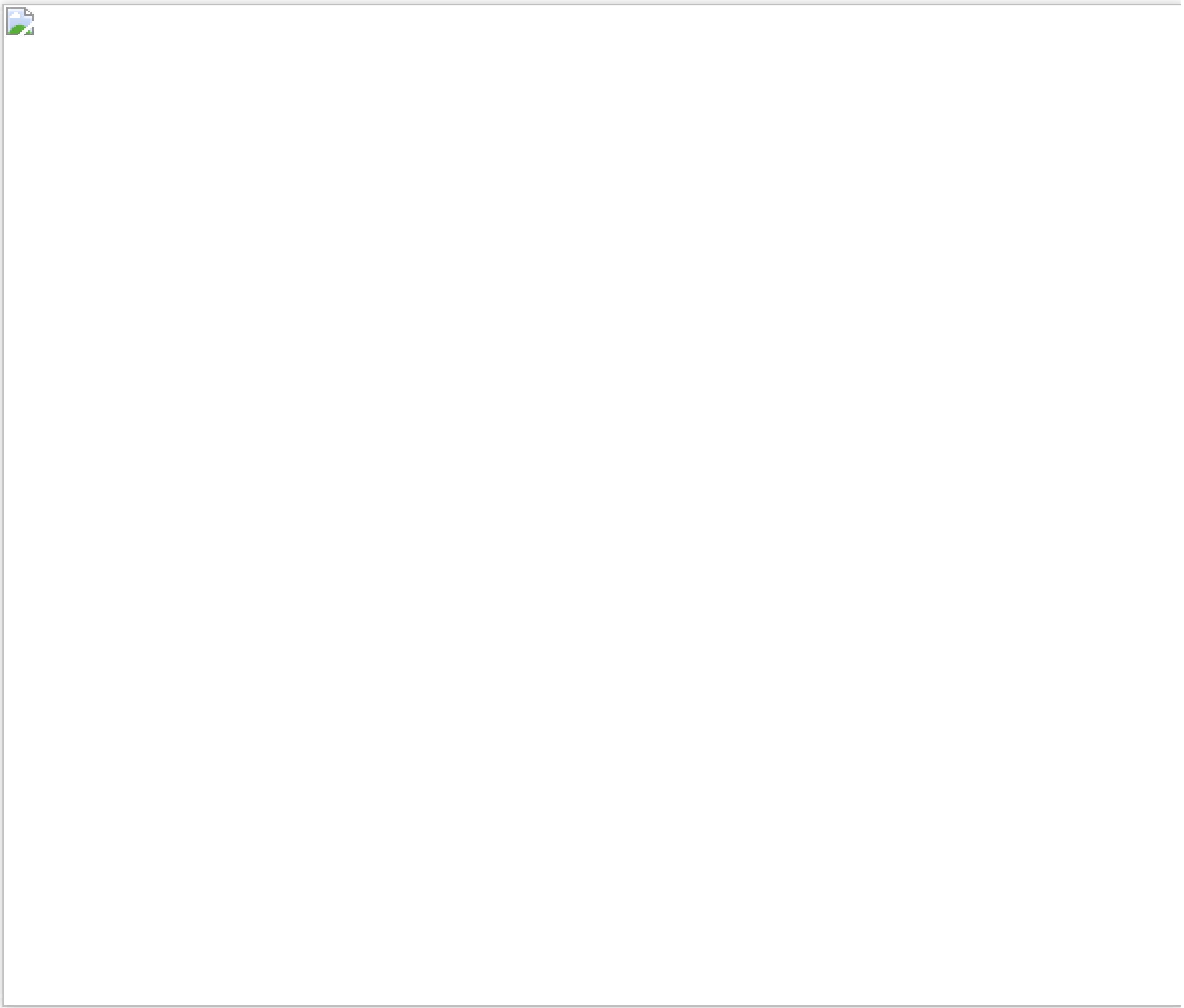
4. Click the **Slow Log Analysis** tab. Then, select the dimension for viewing slow logs and set the query time period in the **Statistics** section.

Click **Instance** to view the instance's slow log statistics.

Click **Proxy Node** and select the target proxy ID in the drop-down list based on the change trend of the **CPU utilization** or the number of **slow queries**.

In the time box, click

to select a time period of up to 4 days to view slow logs.



5. View the slow log statistics change trend, slow query statistics, and slow log list.

Slow log statistics change trend

Slow Log Statistics displays the number of slow queries and CPU utilization. It enables you to quickly identify the CPU utilization value when the number of slow queries stays high in the selected time period. This helps you avoid computer lags or response failures caused by high CPU utilization.

Slow query statistics: **Slow Query Statistics** displays the proportion of the proxy node's slow queries by duration range, where the vertical axis represents the duration range, and the horizontal axis represents the proportion of slow queries. As shown below, 100% of slow queries fall within the duration range of 10–15s.

Slow log list

Slow Log List displays the number and duration of slow query command executions. You can click **Export** to export the data to view and analyze it locally.

Click a command template to display the specific analysis, optimization suggestion, and statistics on the right.

The **Analysis** tab displays the command template, command sample, optimization suggestion, and description.

The **Statistics** tab displays the execution duration distribution of the aggregated commands, as well as the distribution and proportion of access source IPs (for proxies only and unavailable for Redis).

Chart interaction

In the monitoring view of **Slow Log Statistics**, click the target time point. Then, you can view the slow query statistics by duration on the right. You can also view the execution counts and durations of slow queries in the slow log list below.

Monitoring details

On the **Slow Log Statistics** page, click **Monitoring Details** in the top-right corner. In the drop-down list, select the associated monitoring metrics and set the time period to compare the statistics of the maximum and average values of multiple metrics in the time period. Then, click **Add Time Comparison** to compare the statistics of two time periods.

Memory Analysis

Last updated：2023-03-14 15:45:47

Overview

If slots are sharded unevenly in the cluster mode of TencentDB for Redis, a data or query skew may occur. In this case, some Redis nodes with big keys may use more memory and ENI resources, causing Redis blockage.
The memory analysis feature mainly analyzes big keys stored in the database. It dynamically displays the change trend of the instance's memory utilization and collects the statistics of memory utilization, element quantity and length, and expiration time of top 100 big keys in real time. This helps you quickly identify big keys for splitting or clearing after expiration, so you can promptly optimize the database performance and avoid business blockage caused by high memory usage of big keys.

Big key analysis data interpretation

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.
4. Select the **Memory Analysis** tab to view the analysis data of big keys.

Memory utilization

On the **Big Key Analysis** page, the change trend of the instance's memory utilization in the last 30 days is displayed by default. You can select a time period on the timeline to view the specific change trend of memory utilization.

Top 100 big keys

Select a data type in the **Data Type** drop-down list to view the information of the top 100 big keys by memory usage, element quantity, max element length, average element length, and expiration time.

Top 100 Big Keys (by MEM Usage): Top 100 big keys by memory usage from high to low.

Top 100 Big Keys (by Element Quantity): Top 100 big keys by element quantity from high to low.

Top 100 Key Prefixes: Top 100 big keys by prefix.

Quickly find big keys

In the monitoring view of memory utilization, the change trend of the instance's memory utilization in the last 30 days is displayed by default. If you find that the memory utilization on a certain date is high, you can click the date on the date axis, then the time column will be fixed, and the information of big keys of that date will be displayed in the list of the top 100 big keys at the bottom, so you can quickly identify big keys with a high memory utilization.

Creating ad hoc analysis of big keys

1. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list. Then, select the **Memory Analysis** > **Ad Hoc Analysis of Big Key** tab.
2. Click **Create Task**, and DBbrain will fetch the last backup of the database for automatic analysis. You can view the analysis progress on the progress bar in the task list.
3. After the analysis is completed, click **View** in the **Operation** column to view the result of big key analysis. If the big key needs to be deleted, click **Delete** in the **Operation** column in the task list.
4. View the analysis result in the **Analysis Result** panel on the right.

Latency Analysis

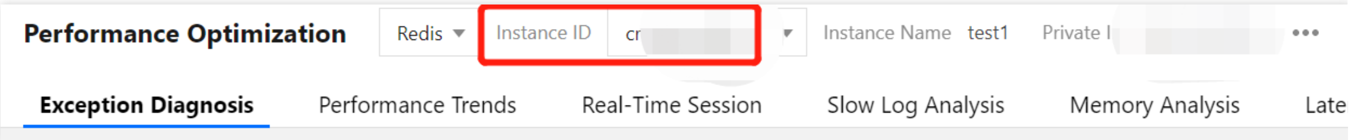
Last updated：2023-05-23 10:52:35

Feature Overview

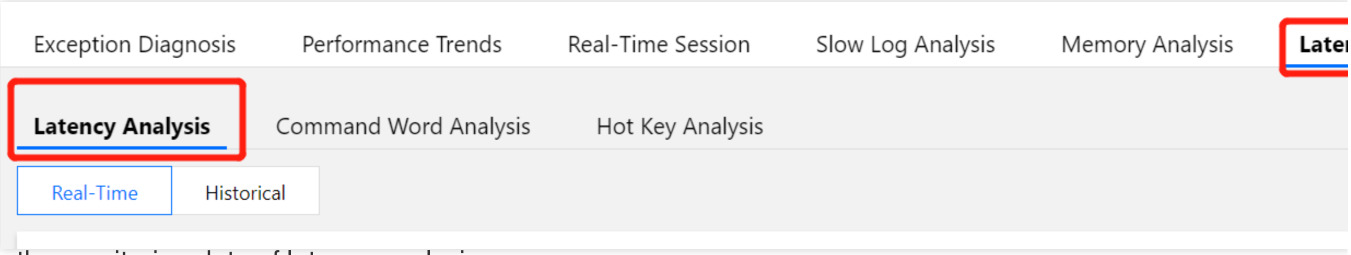
DBbrain's latency analysis feature collects the execution latency statistics of all database requests and monitors the latency accurate down to the millisecond. This helps you troubleshoot TencentDB for Redis instance failures and performance problems.

Viewing Latency Analysis

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



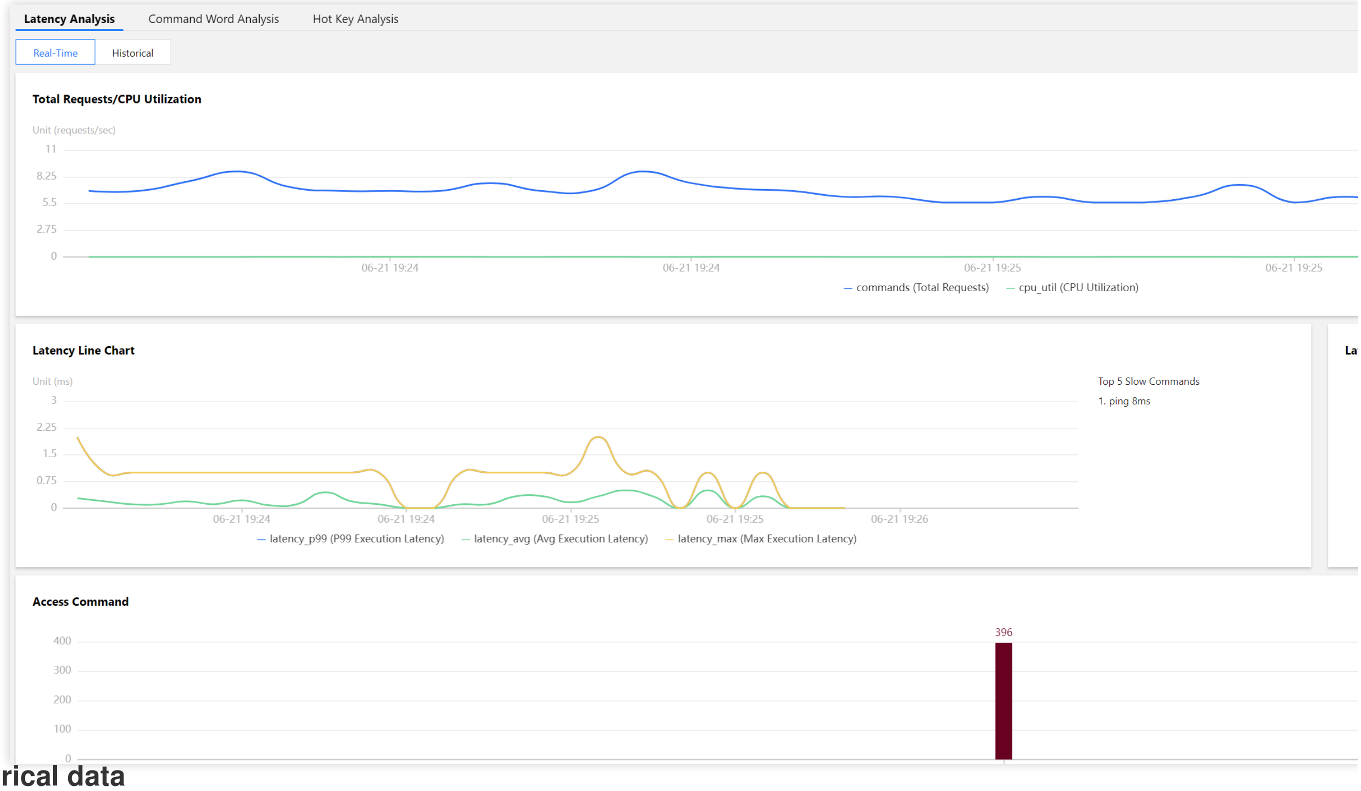
4. Select the **Latency Analysis** tab and set the collection granularity to **5s**, **15s**, or **30s** as needed in the drop-down list next to **Auto-Refresh** in the top-right corner.



5. View the monitoring data of latency analysis.

Real-time monitoring

By default, the change in the real-time monitoring data is displayed as a curve.

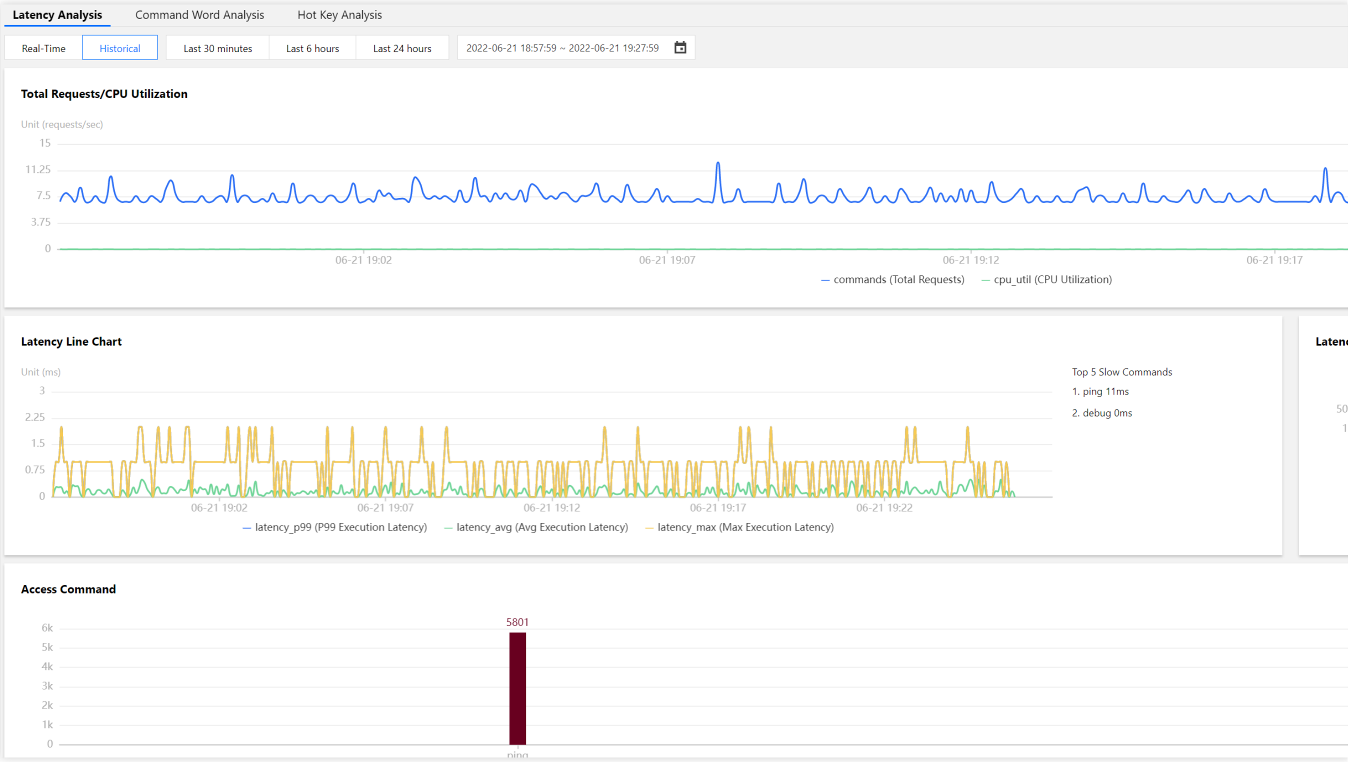


Historical data

Click **Historical** to view the monitoring data in a past time period.

You can select **Last 30 minutes**, **Last 6 hours**, or **Last 24 hours** to view the specific monitoring data.

Click 📅 in the time selection box to view the monitoring data in the **last 2 days**.



Latency Analysis Statistics Interpretation

Total requests/CPU utilization

The change trend of the total number of requests to the database instance per second and the corresponding CPU utilization are displayed for you to quickly identify the CPU utilization value when the number of requests is high.

Latency line chart

The change trends of three key metrics for database request execution latency and the top five slow commands are displayed.

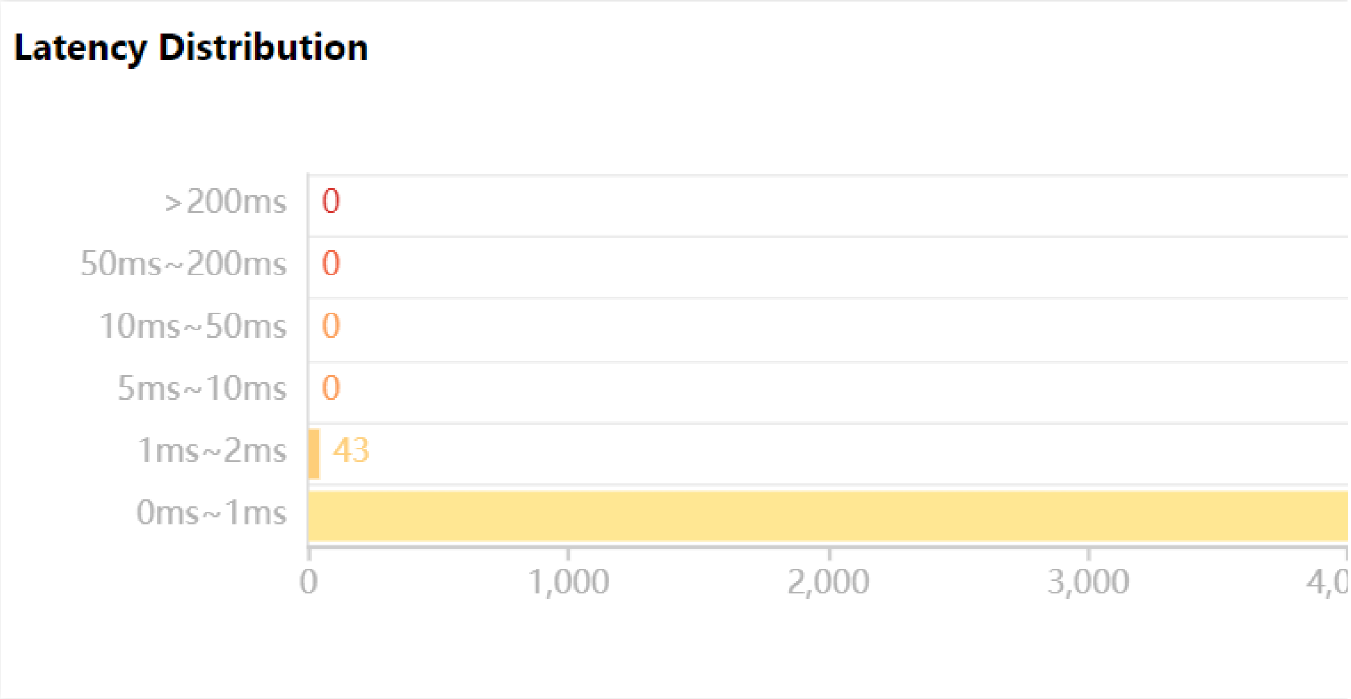
P99 Execution Latency: The change trend of the 99th percentile request execution latency.

Average Execution Latency: The change trend of the average request execution latency.

Max Execution Latency: The change trend of the maximum request execution latency.

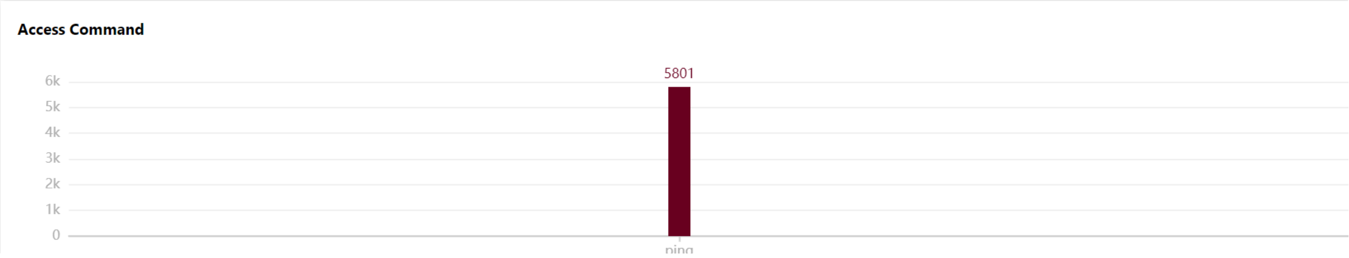
Latency distribution

The numbers of commands with an execution latency in different ranges are displayed in a bar chart, including 0–1 ms, 1–2 ms, 5–10 ms, 10–50 ms, 50–200 ms, and >200 ms. As shown below, the number of commands with an execution latency of 0–1 ms is 2,703.



Access command

The number of hits of database access commands is displayed in a column chart. As shown below, the numbers of access requests made by using the AUTH and GET commands are displayed respectively. Click the **Command Word Analysis** tab to view the statistics of various commands. For more information, see Command Word Analysis.



Command Word Analysis

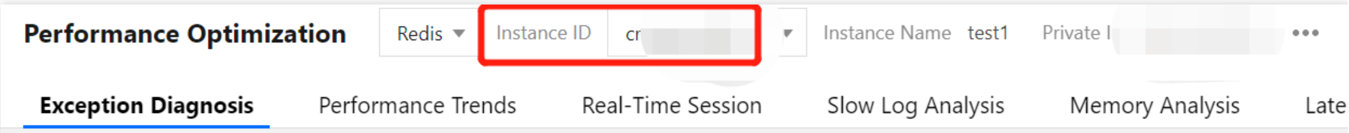
Last updated：2023-05-23 10:44:00

Feature Overview

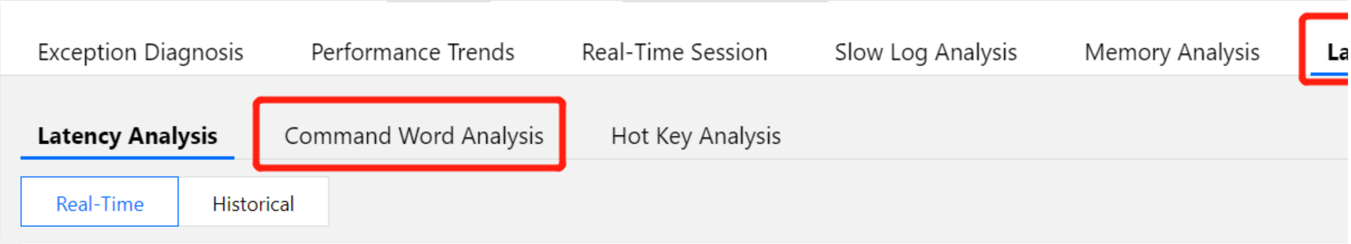
The command word analysis feature analyzes the number and latency of database access commands statistically. Latency analysis makes it simple to identify the command that is used the most frequently, and command word analysis helps you further identify when it is most frequently executed as well as its execution latency, which facilitates troubleshooting and performance optimization.

Viewing Command Word Analysis

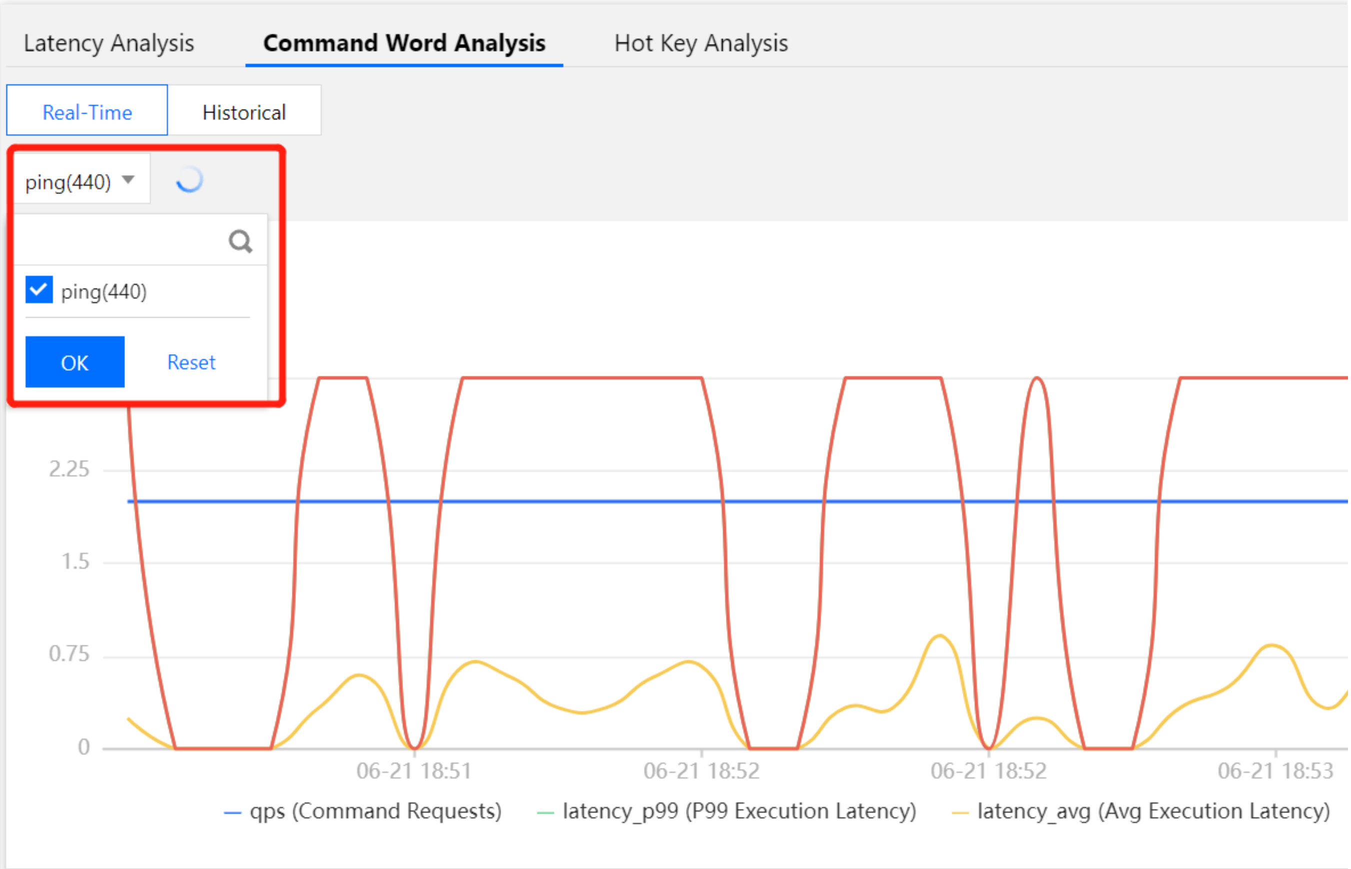
1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



4. Select the **Latency Analysis** > **Command Word Analysis** tab and set the collection granularity to **5s**, **15s**, or **30s** as needed in the drop-down list next to **Auto-Refresh** in the top-right corner.



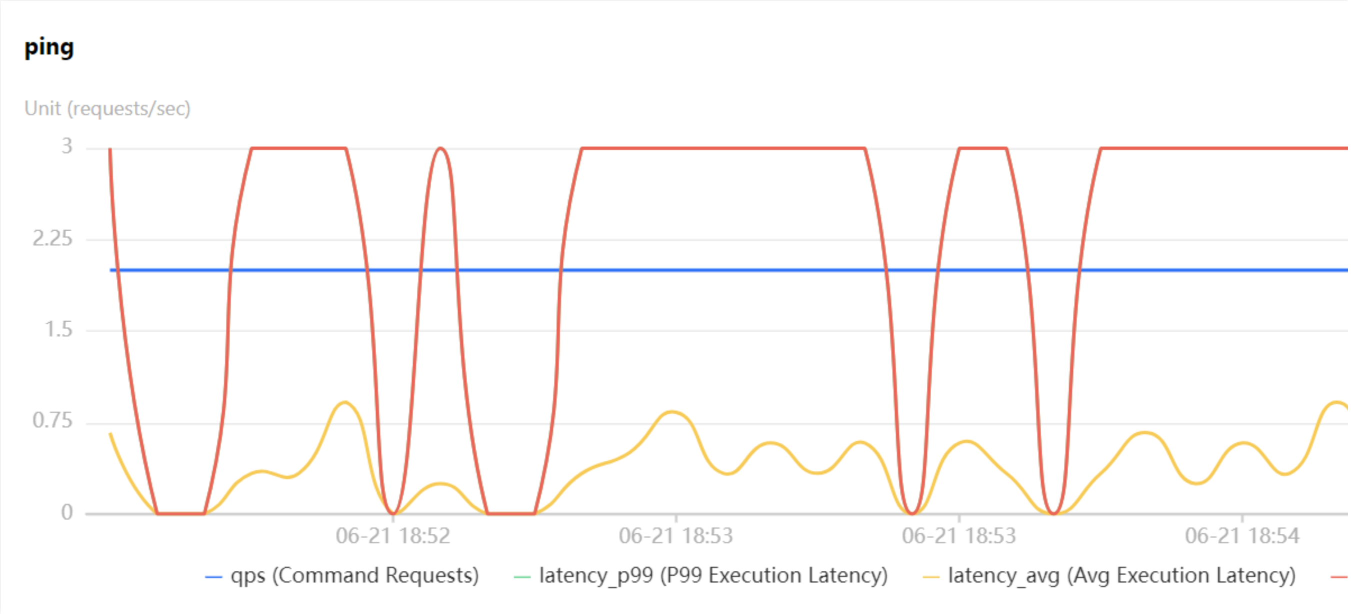
5. (Optional) Quickly filter target command words in the drop-down list in the top-left corner.



6. Analyze the trend data of the command words.

Real-time statistics

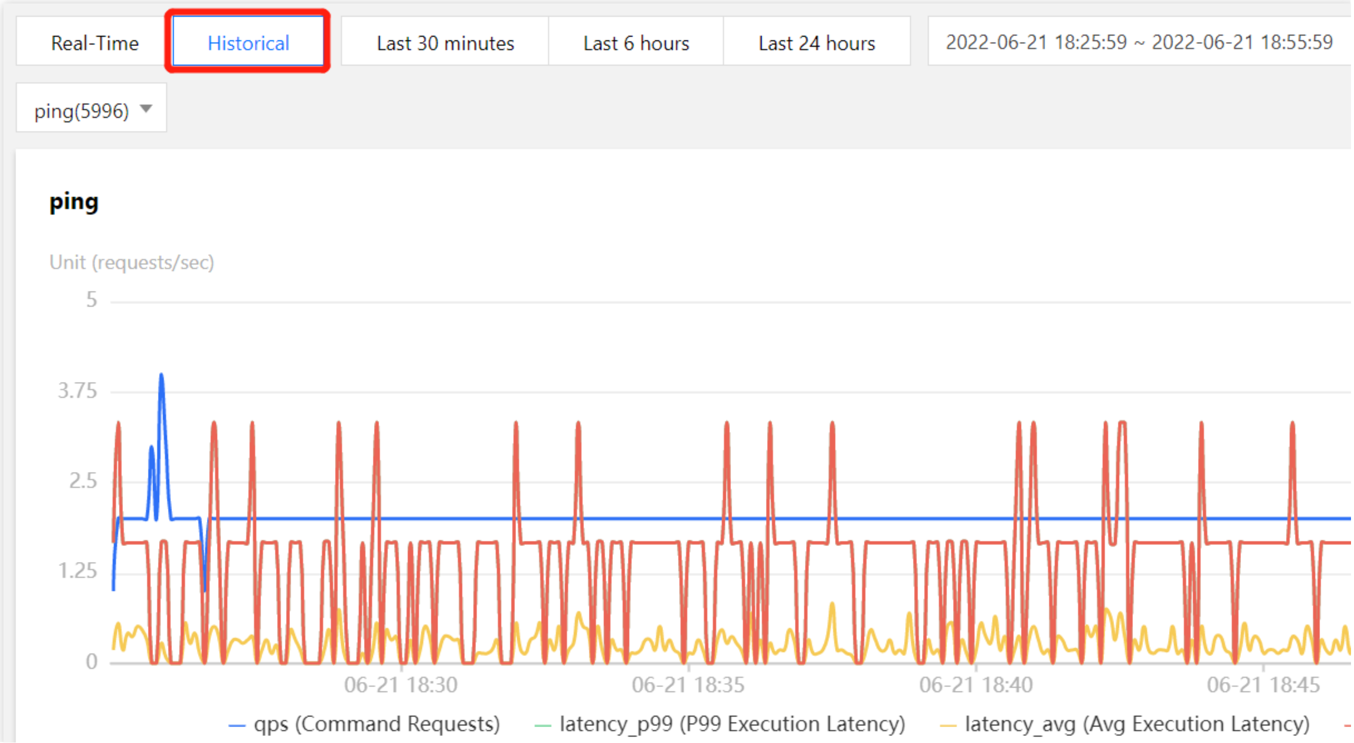
Real-time statistics are displayed by default, including the change trends of command requests, P99 execution latency, average execution latency, and max execution latency. For more information on metrics, see Performance Trends.



Historical data

Click **Historical** to view the statistics in the last 30 minutes, 6 hours, or 24 hours. You can also click

📅 in the time selection box to view the statistics in the last 2 days.



Hot Key Analysis

Last updated：2023-03-14 15:39:31

Overview

In Redis, frequently accessed keys are called hot keys. When a Redis database receives a lot of requests to access a hot key, the traffic gets too concentrated and reaches the upper limit of the physical ENI, which will cause problems or even downtime of the Redis service.

With DBbrain's hot key analysis feature, you can find frequently accessed hot keys quickly to optimize the database performance accordingly.

Viewing the hot key analysis data

1. Log in to the TencentDB for Redis console.
2. On the left sidebar, select **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.

4. Select the **Latency Analysis** > **Hot Key Analysis** tab and set the collection granularity to **5s**, **15s**, or **30s** as needed in the drop-down list next to **Auto-Refresh** in the top-right corner.

5. View the statistics of hot keys. You can switch between the real-time and historical views.

Real-time view

By default, the access frequency of the current database's hot keys is displayed in real time.

Historical view

Click **Historical** to view the statistics in the last 1, 3, or 24 hours or last 7 days. You can also click in the time selection box to query the statistics in a period of up to 7 days in the last month.

Sentinel Mode

Last updated：2023-05-23 10:42:40

Overview

Sentinel is a standalone process that monitors the master and replica nodes in a Redis cluster. When the master node fails, Sentinel can elect a new master from the replica nodes to replace it automatically. This high-availability solution ensures that business operations run smoothly.

Sentinel Commands

TencentDB for Redis 4.0 and later support the Sentinel mode by default. You can use the following Sentinel commands.

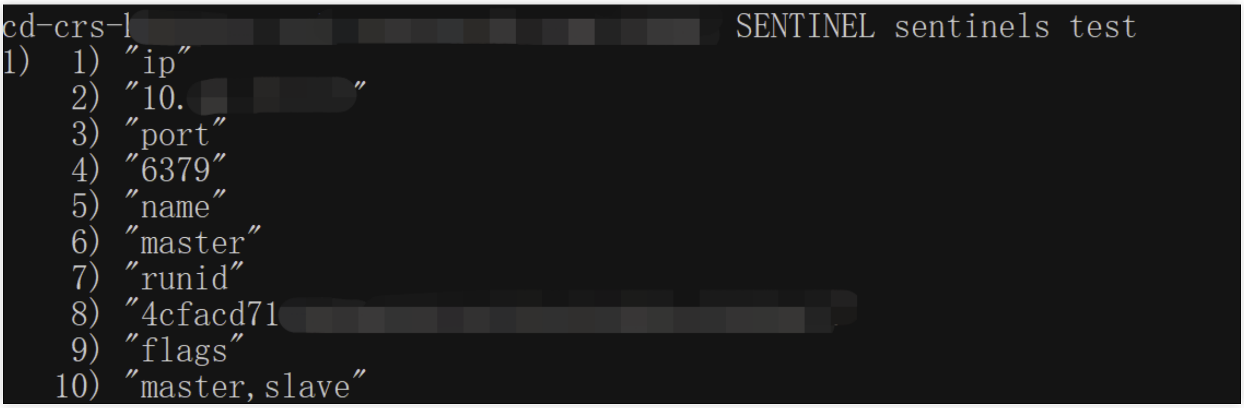
SENTINEL sentinels

This command lists the sentinels information of the monitored master.

Command format

```
SENTINEL sentinels <any name>
```

Sample code



```
cd-crs-[redacted] SENTINEL sentinels test
1)  1) "ip"
    2) "10.[redacted]"
    3) "port"
    4) "6379"
    5) "name"
    6) "master"
    7) "runid"
    8) "4cfacd71[redacted]
    9) "flags"
   10) "master,slave"
```

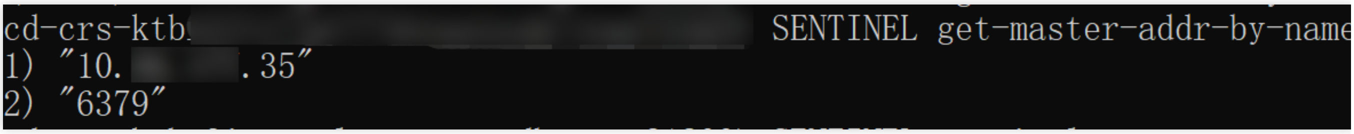
SENTINEL get-master-addr-by-name

This command gets the IP address information of the `master-name` .

Command format

```
SENTINEL get-master-addr-by-name <any name>
```

Sample code



```
cd-crs-ktb[redacted] SENTINEL get-master-addr-by-name
1) "10.[redacted].35"
2) "6379"
```

Connection Sample for the Sentinel Mode

Prerequisites

The Redis instance version is 4.0 or 5.0.

The database instance is in the **Running** status.

Get the **private IPv4 address** and **port** information for database connection in the **Network Info** section on the **Instance Details** page in the TencentDB for Redis console. For detailed directions, see Viewing Instance Information.

Get the account and password for database access. For detailed directions, see Managing Account.

Download and install Jedis. The latest version is recommended.

Connection sample

The following sample code takes Jedis 3.6.0 as an example. The latest version is recommended.

The Jedis version is 3.6.0 or later.

The Lettuce version is 5.3.0.RELEASE or later.

The Spring Data Redis version is 2.5.1 or later, for which the `spring.redis.sentinel.password` parameter should be configured.

You need to modify the parameters based on the comments, including IP, port, account, and password for database access.

Connection via Java

```
package com.example.demo;

import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.JedisSentinelPool;

import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        String masterName = "test";
        Set<String> sentinels = new HashSet<>();
        // You need to configure the database instance's private IPv4 address and p
        sentinels.add("XX.XX.XX.XX:6379");
        GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
        String dbPassword = "root:xxx";// Replace this with your database access pa
        String sentinelPassword = "root:xxx";// Replace this with your database acc

        JedisSentinelPool jedisSentinelPool =
            new JedisSentinelPool(masterName, sentinels, poolConfig,
                2000, 2000, dbPassword,
                0, null, 2000, 2000,
                sentinelPassword, null);
        System.out.println("jedisSentinelPool.getResource().ping() = " + jedisSenti
        jedisSentinelPool.close();
    }
}
```

Connection via the Spring Data framework

```
package com.example.demo;

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.condition.ConditionalOnBean;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.RedisPassword;
import org.springframework.data.redis.connection.RedisSentinelConfiguration;
import org.springframework.data.redis.connection.jedis.JedisConnectionFactory;
import org.springframework.data.redis.core.RedisTemplate;
import redis.clients.jedis.JedisPoolConfig;

@SpringBootApplication

public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}

@Configuration
class RedisConfig {
    @Bean
    @Qualifier("jedisConnectionFactory")
    public JedisConnectionFactory connectionFactory() {
        RedisSentinelConfiguration sentinelConfig = new RedisSentinelConfiguration(
            .master("test")
            .sentinel("XX.XX.XX.XX", 6379);// Replace this with the private IPv4 ad
        sentinelConfig.setPassword(RedisPassword.of("xxx"));// Replace this with yo
        sentinelConfig.setSentinelPassword(RedisPassword.of("xxx"));// Replace this
        JedisPoolConfig poolConfig = new JedisPoolConfig();
        JedisConnectionFactory connectionFactory = new JedisConnectionFactory(senti
        connectionFactory.afterPropertiesSet();
        return connectionFactory;
    }

    @Bean
    @ConditionalOnBean(JedisConnectionFactory.class)
    public RedisTemplate<String, String> redisTemplate(@Qualifier("jedisConnectionF
        RedisTemplate<String, String> template = new RedisTemplate<>();
        template.setConnectionFactory(factory);
        template.afterPropertiesSet();
        //test
        template.opsForValue().set("test", "test1");
        System.out.println("template.opsForValue().get(\\"test\\") = " + template.o
        return template;
    }

}
```