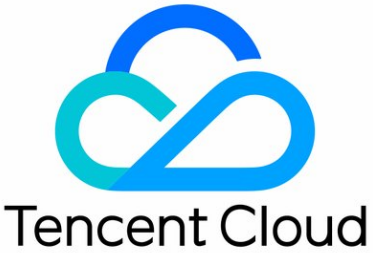
TencentDB for Redis

Operation Guide



Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Operation Guide
Operation Overview

Last updated: 2024-11-01 16:01:27

While using TencentDB for Redis®, you may encounter problems related to accessing, maintaining, backing up, and restoring instances. This document introduces 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 Instance
- Assigning Project to Instance
- 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
- Enabling Password-Free Access
- Managing Account
- Configuring Instance Parameters
- Applying Parameter Templates
- Disabling Commands

Name

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

Backup and restoration

- Data Backup
- Restoring Data
- Cloning Data

Migrating Data

- Migrating with redis-port
- Migrating with DTS

Monitoring and Alarms

Security Group

Security Group is a stateful virtual firewall with filtering capabilities, used for configuring network access control for single or multiple TencentDB instances. It is an important means of network security isolation provided by Tencent Cloud.

Networking

- Configuring Network
- Authorizing Policy Syntax
- Configuring Public Network Address

Monitoring and Alarms

- Monitoring with Five-Second Granularity
- Multi-instance Comparison Monitoring
- Configure Alarms

Global Replication

- Creating Global Replication Group
- Managing Global Replication Group

Cloud Access Management
CAM Overview

Last updated: 2024-02-21 12:47:59

Known 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

You can mitigate these issues by using sub-accounts to allow different users to manage different services. By default, sub-accounts are not permitted use cloud services or access related resources. Therefore, you need to create policies that grant sub-accounts the necessary permissions to use the resources they require.

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.

If you do not need to manage the access permission to TencentDB resources for sub-accounts, you can skip this section. This will not affect your understanding and usage of other parts in the documentation.

Getting Started

A CAM policy must either grant or deny permissions for one or more Redis operations and specify the resources that can be used for these operations (all or partial). Additionally, the policy can include conditions set for these operations.

> **Note:**
> - It is recommended that users utilize CAM policies to manage Redis resources and authorize Redis operations. This approach maintains the existing user experience for users currently using project-based permissions, but it is not advised to continue using project-based permissions for resource management and authorization.
> - Effectiveness conditions cannot be set for Redis for the time being.

Related Content	Document
Basic policy structure	Policy Syntax
Operation definition in a policy	Redis Operations
Resource definition in a policy	Redis Resource Paths
Resource-level Permissions	Resource-level Permissions Supported by Redis

Authorization Policy Syntax

Last updated: 2023-09-15 15:21:04

CAM Policy Syntax

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

- **version** is required. Currently, only the value "2.0" is allowed.
- **statement** describes the details of one or more permissions. It contains a permission or permission set of multiple other elements such as `effect`, `action`, `resource`, and `condition`. One policy has only one `statement`.
- **effect** is required. It describes the result of a statement. The result can be "allow" or an "explicit deny".
- **action** is required. It describes the allowed or denied action (operation). An operation can be an API or a feature set (a set of specific APIs prefixed with "permid").
- **resource** describes the objects the statement covers. A resource is described in a six-segment format. Detailed resource definitions vary by product. This element is required.
- **condition** is required. It describes the constraints for the policy to take effect. Conditions consist of operators, keys, and values. Condition values can include information such as time and IP address, and some services allow specifying other values in the conditions.

Redis Operations

In a CAM policy statement, you can specify any API operation from any service that supports CAM. For Redis, use APIs prefixed with `redis:` such as `redis:CreateRedis` or `redis:DeleteInstance`.
To specify multiple operations in a single statement, separate them with commas as shown below:

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

You can also use wildcards to specify multiple actions. For example, you can specify all actions whose names begin with the word "Describe", as follows:

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

To specify all operations in Redis, use a wildcard (*) as follows:

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

Redis Resource Path

Each CAM policy statement has its own applicable resources.
Resource paths are generally in the following format:

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

- **project_id**: Describes project information, included only for compatibility with early CAM logic, and does not need to be filled in.
- **service_type**: Product abbreviation, such as `redis`.
- **region**: Region information, such as "bj".
- **account**: Describes the root account of the resource owner, such as `uin/12345678`.
- **resource** describes the detailed resource information of each product, such as "instance/instance_id" or "instance/*".

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 it for 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, you can use the wildcard (*) in the `resource` element as shown below:

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

To specify multiple resources in one policy, separate them with commas. In the following example, two resources are specified:

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

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

Resources	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 Type

Last updated: 2024-11-01 16:18:08

Resource-level permissions see the capability to specify which resources a user can execute operations on. Redis partially supports resource-level permissions, which allow control over user execution of certain operations or access to specific resources. The types of resources that can be authorized within the CAM (Cloud Access Management) system are as follows:

ResourceType	Resource Description Method in Authorization Policy
TencentDB for Redis® instance-related	`qcs::redis:$region::instance/*` `qcs::redis:$region:$account:instance/$instance`

The table below presents the currently supported Redis API operations with resource-level permissions, as well as the resources supported by each operation. When specifying a resource path, you can use the wildcard "*" within the path.

List of APIs supporting resource-level authorization

Configuring using API	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
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 TencentDB API operations that don't support resource-level authorization, you can still authorize a user to perform them, but you must specify `*` as the resource element in the policy statement.

Configuring using API	API Description
CreateInstances	Creating a TencentDB for Redis® instance
CreateParamTemplate	Creates a parameter template.
DeleteParamTemplate	Deletes a parameter template.
DescribeInstanceDealDetail	This example shows you how to query the order information.
DescribeParamTemplateInfo	Retrieving Parameter Template Details
DescribeParamTemplates	Retrieving Parameter Template List
DescribeTaskInfo	DescribeTasks
DescribeTasks	Querying job list
ModifyParamTemplate	Modifies a parameter template.
ListUsers	Querying Sub-account Name
ListCollaborators	Querying Collaborator Account Name
ListWeChatWorkSubAccounts	Querying WeChat usernames

SDK Connection
PHP Connection Sample

Last updated: 2023-09-15 15:21:16

Prerequisites:
Download the client phpredis.
Sample Code:

```
<?php
  /*Enter your Redis instance's private IP, port number, instance ID, and password in the following
parameters/
  $host = "192.xx.xx.2";
  $port = 6379;
  $pwd = "123tj6na";

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

  /*You can start manipulating the Redis instance. For more information, visit
https://github.com/phpredis/phpredis./

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

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

Execution results:

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

Java Connection Sample

Last updated: 2024-02-21 12:58:18

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

Preparation

- In the 'Network Information' section of the 'Instance Details' page on the 'Redis Console', obtain the 'Internal IPv4 Address' and port for connecting to the database. For more information, please refer to 'View Instance Details'.
- Obtain the account and password for database access. For detailed instructions, see Managing Accounts.
- 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, and password in
the following parameters in case of private network access.
                        Configure the instance's public network address, port number, and password
in case of public network access. No need to set the instance ID.*/
            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, visit
https://github.com/xetorthio/jedis./
            // 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:

```
[root@VM_0_194_centos bin]# ./java -cp jedis-2.4.2.jar:. HelloRedis
set key redis suc, value is: tencent
get key redis is: tencent
```

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.getInstance("PKIX");
        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 GenericObjectPoolConfig();

        //with ssl config jedis pool
        // vip is the private IPv4 address for database connection, 6379 is the default port
number, and pwd is the password of the default account. You need to replace them as needed.
        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: 2023-09-15 15:21:27

Prerequisites:

Run the following command to install node-redis:

```
npm install hiredis redis
```

Sample Code:

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

/*Enter your Redis instance private IP, port number, instance ID, and password in the following
parameters/
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 now start operating 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 results:

```
[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: 2024-02-21 12:51:29

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

Preparation

- In the Redis Console, navigate to the **Instance Details** page and locate the **Network Information** section to obtain the **Internal IPv4 Address** and port for connecting to the database. For more details, please refer to Viewing Instance Details.
- Obtain 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 you connect
through the default account, it is pwd. If you connect through a custom account, it is
user+'@'+pwd
r = redis.StrictRedis(host=host, port=port, password=user+'@'+pwd)

#Database operations can be performed after the connection is established. For more information,
visit https://github.com/andymccurdy/redis-py
r.set('name', 'python_test');
print r.get('name')
```

Execution results:

```
[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 port number, pwd is
the password of the default account, and ca.pem is the obtained SSL certificate file. You need to
replace them as needed.
    client = redis3.Redis(host="vip", port=6379, password="pwd", ssl=True,
ssl_cert_reqs="required",
                          ssl_ca_certs="ca.pem")
    print(client.ping())
```

C Connection sample

Last updated: 2023-09-15 15:21:38

Prerequisites:
Download 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 [redacted] 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: 2024-02-21 12:50:15

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

   if err != nil { // Check for connection errors

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

      return

   }

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

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

   if err != nil { // Error setting value

      log.Println(err)

      return

   }

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

   if err != nil {

      log.Println(err)
```

```
        return

    }

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

}
```

Execution results:

```
[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: 2024-02-21 12:53:32

Prerequisites:
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"; //Instance access host address
            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 + ":" + pass);
            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(key));
//Retrieve value
            System.Console.WriteLine("value:" + getValue);
            System.Console.Read();
          }
     }
}
```

- Using 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://:InstanceID:Password@AccessAddress:Port/
            };
            RedisConfig.VerifyMasterConnections = false;//Required setting
            PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10/Number
of connection pools/,
            10/Connection pool timeout/, testReadWriteHosts);
            for (int i = 0; i < 100; i++)
            {
                IRedisClient redisClient = redisPoolManager.GetClient();//Get connection
                RedisNativeClient redisNativeClient = (RedisNativeClient)redisClient;
                redisNativeClient.Client = null;//Needs 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();
        }
    }
}
```

- Using 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@Access Address:
Port/
               };
               PooledRedisClientManager redisPoolManager = new
PooledRedisClientManager(10/Number of connection pools/
               count/, 10/Connection pool timeout*/, testReadWriteHosts);
               for (int i = 0; i < 100; i++)
              {
               IRedisClient redisClient = redisPoolManager.GetClient();//Get connection
               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:

Daily Instance Operation
Viewing Instance Information

Last updated: 2024-11-01 16:18:49

Scenario

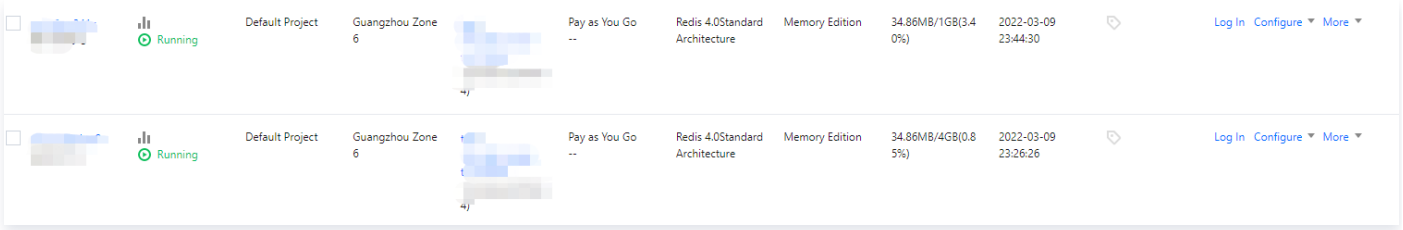
After purchasing a TencentDB for Redis® instance, you can quickly view its details in the console, such as the status, capacity usage, master/replica nodes in the cluster, and network status. You can also perform Ops and management operations efficiently.

Preparations

- You have created a 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 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.
   - 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 cannot find the target instance in the instance list, please check the **Recycle Bin** on the left sidebar to confirm whether the instance has been isolated in the recycle bin due to an expired payment. For more information, see Restoring Isolated Instance.
4. View the target instance information, such as the status, specification, and storage engine.

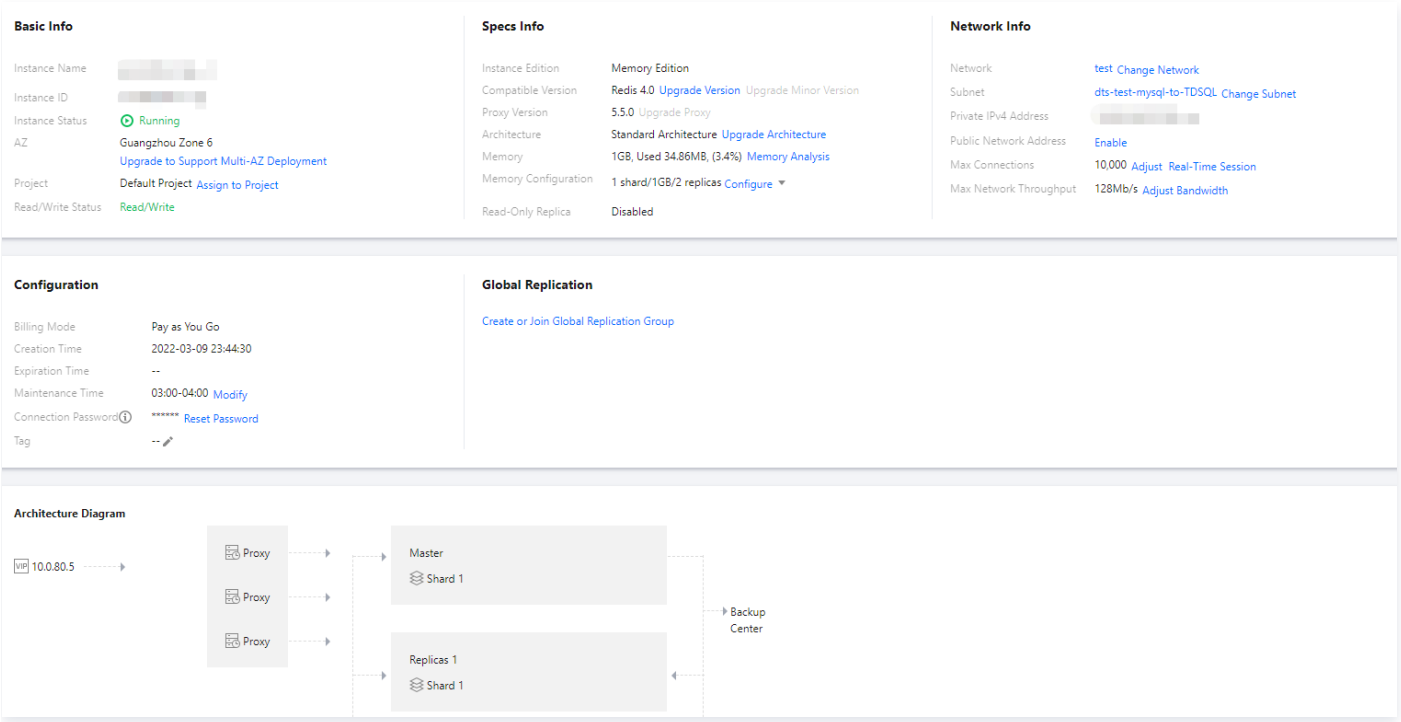


Instance Information List	Description
Instance ID/Name	• Instance ID: The instance's unique ID. • Name: The name set when creating the instance. Mouse over the instance name, click to edit the instance name for easy identification and management.
Monitoring/Status/Task	• Monitoring: Click to view the monitoring metrics data of the instance on the monitoring panel. For more information, see Monitoring Features. • Status: The running status of the instance. The normal status is **Running**. • Task: When a task is being executed, the name of the current task for the instance will be displayed here. For example, "Configuration Change in Progress."
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 of 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 within the same region.
Network	Displays the name of the private network and subnet to which the instance belongs, as well as the private IPv4 address. Click on the blue text of the private network name to view detailed information about the network. The private IPv4 address is required when accessing the database, and this IP address needs to be configured. For more information, see Connecting to a Redis Instance.
Billing Mode	Billing Mode: Pay-as-you-go or monthly subscription. The two billing methods differ; for more information, see Billing Overview.
Auto-renewal	Specify whether the auto-renewal feature is enabled for the current instance.
Architecture Version	For the version information and architecture of the database, see Product Series.
Product Version	Currently, only **Memory Edition** is supported.
Used/Total Capacity	The instance's currently used memory and total memory.
Creation Time	Specific date and time of instance creation.
Tags	Displays tag information for an instance. Instances can be managed by tags. For how to edit tags, see Editing Instance Tags.
Operation	• Click **Log In** to access the database through DMC. For directions, see Connecting to TencentDB for Redis® Instance. • Click **Configure** and select Expand Node, Reduce Node, Add Replica, Delete Replica, Add Shard, or Delete Shard to expand or reduce the memory capacity of instance nodes, add or delete instance replicas, and increase or decrease the number of shards in the cluster architecture, respectively. For detailed instructions, 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 Monthly Subscription instances, select **More** > **Return and Refund**; for pay-as-you-go instances, select **More** > **Terminate** to return the instance and isolate it in the recycle bin. For detailed steps, 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 to navigate to the **Instance Details** page.



Section	Parameter	Description
Basic Information	Instance Name	Name: The name set when creating the instance. Mouse over the instance name, click to edit the instance name for easy identification and management.
	Unique instance ID	The instance's unique ID.
	Instance Status	The instance's current status. The normal status is **Running**.
	Availability Zones	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 the instance is associated with. Click **Assign to Project** to reassign the instance to another project. For detailed instructions, see Assigning Instance to Projects.
	Read/Write Status	The current read/write status of the database.
Specification Information	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	Information about the version of the Redis proxy. If the "Upgrade Proxy" text is gray, it means the current proxy version is the latest. If the text is blue, you can click **Upgrade Proxy** to upgrade to a higher version. For detailed operations, please refer to Upgrading Proxy.

	Architecture Version	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.
	Storage Capacity	The total memory capacity, used capacity, and the percentage of used capacity for the current instance. Click **Memory Analysis** to navigate to the Memory Analysis page under Performance Optimization. Here, you can view the memory overhead of large keys in the database, quickly identify large keys, analyze their information, and perform splitting or cleaning operations. For more information, see Memory Analysis.
	Memory Configuration	The memory configuration specification of the purchased instance includes shard quantity, memory capacity of each shard node, and 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 Information	Network	The instance's VPC name can be changed by clicking **Switch Network**, allowing you to switch the VPC and subnet. For more information, see Configuring Network. 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 directly copy the IP address and port number.
	Public Network Address	Access the public network address of the database. By default, public network access is not enabled. Click **Enable** to enable public network access for convenient daily testing and management. For detailed steps, 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 Sessions** to view the current real-time session sources, active connection count, and other statistics for the instance. For more information, see Real-Time Sessions.
	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 Information	Billing Mode	Instance billing modes: pay-as-you-go and monthly subscription.
	Creation time	Creation time of the instance.
	Expiration Time	For the Yearly/Monthly Subscription billing mode, the instance expiration date is displayed.
	Maintenance Time	The maintenance time for the instance. To ensure database stability, the backend system periodically performs maintenance operations on the

		instance during the maintenance time. Click **Modify** to adjust the maintenance time, preferably during off-peak business hours. For detailed instructions, see Setting Maintenance Time.
	Connection Password	When connecting to the database, you need to configure the password. Click **Reset Password** to reset the password or enable password-free authentication. For detailed steps, see Resetting Password.
	Tag	The tag associated with the instance, which can be modified. For directions, see Editing Instance Tag.
Data Synchronization	Sync Mode	The data sync mode used by the instance, such as DTS.
	Sync jobs	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 the instance in data synchronization, indicating whether it is a source or target instance.
	Synchronize Instance	The peer instance ID of the sync instance and its name.
Global Replication	Create or Join Global Replication Group	The instance is not part of a global replication group. Click **Join** or **Create Global Replication Group** to apply for membership. For detailed steps, see Creating a Global Replication Group. Before joining a global replication group, please familiarize yourself with its implementation mechanism and usage restrictions.
	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	The database instance's deployment architecture diagram.	

More Operations

Renaming instance

1. In the **Instance List**, hover your mouse over the instance name you want to modify, and click the icon on its right side.
2. In the instance name input box, enter a new name that is easy to identify and manage. The name 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 **Customized List Fields** page, select the fields you want to display.

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 to export the entire list.

Related APIs

Related APIs	API Description
describeInstances	Retrieve Instance Information List

Assigning Instance to Project

Last updated: 2024-11-01 16:19:32

Scenario

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.
Assign a suitable project to your database instances to facilitate multi-party collaboration. This enables you to manage cloud resources globally across the entire project and stay informed about the project's expenditure details at all times.

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

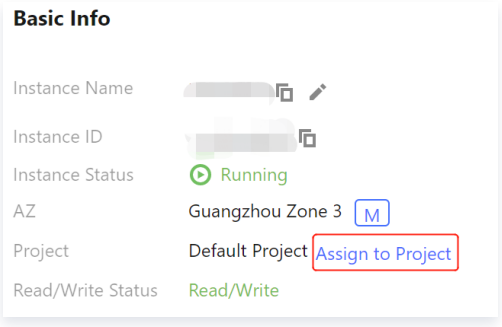
> **Note:**
> Assigning and moving database instances across projects will not affect the services provided by the instances.

Preparations

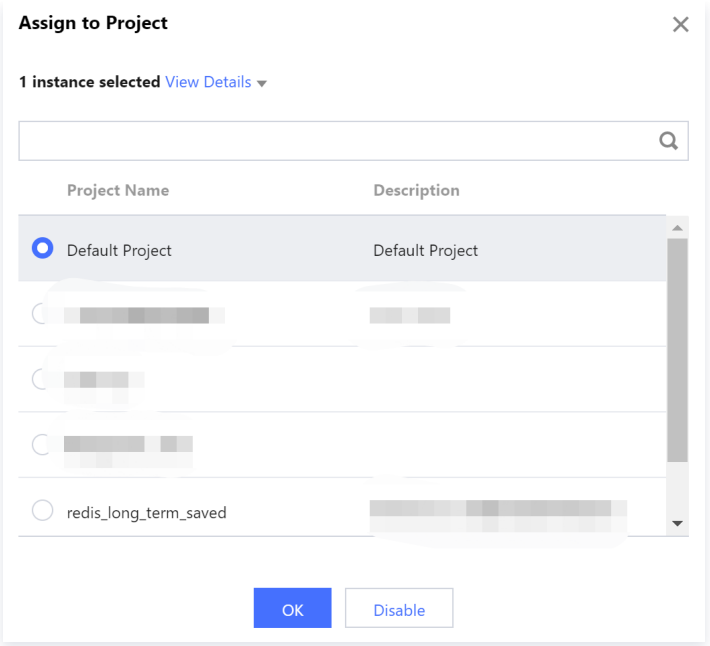
- You have created a TencentDB for Redis® instance and assigned it to a project. Unassigned instances belong to the **default project**.
- The database instance is in **Running** status, with no ongoing tasks.
- You have created the target project in Project Management.

Instructions

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: 2024-11-01 16:19:57

Scenario

Tags, composed of a key and a value, can be used to label TencentDB for Redis® instances. If your Tencent Cloud account has multiple types of cloud resources with various associations, and the number of resources is constantly increasing, managing them can become challenging. You can use tags to group and categorize resources with similar functions or associations. During routine maintenance or troubleshooting, you can quickly search for resources based on tags and perform batch operations for efficient management.

Billing description

Tag management is a complimentary service provided by Tencent Cloud for your Tencent Cloud account, with no additional fees. Simply access the Console to use the product.

Instructions for Use

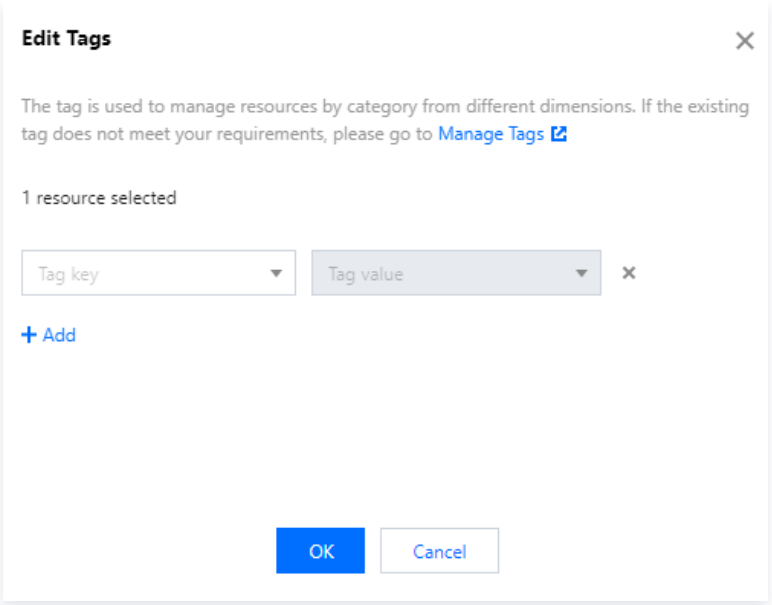
- A tag consists of one tag key and one tag value (tagKey:tagValue).
- Up to 50 tags can be bound to an instance.
- For each instance, a tag key can correspond to only one tag value.

Preparations

- 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.

Instructions

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 .
5. On the **Edit Tags** page, reselect the appropriate tag key from the **Tag Key** dropdown list, and choose the corresponding tag value in the **Tag Value** input box.



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, familiarize yourself with the important information regarding 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.
   - English letters, digits, and Chinese characters are supported.
   - It can contain the following special symbols: plus sign "+", equal sign "=", underscore "_", hyphen "-", dot ".", colon ":", slash "/", at "@", English parentheses "()", Chinese parentheses " （ ） ", square brackets "[]", and Chinese brackets " 【 】 ".

   6.5 Click **OK** to complete the creation.
   6.6 Return to the **Edit Tag** page of the database instance, click **Reload** in the drop-down list of tag keys, select the newly created tag key, and then select the corresponding tag value.
7. Click **OK** to complete the configuration.

Additional References

For more information on tag management, see Tag.

Setting Maintenance Time

Last updated: 2024-11-01 16:20:17

Scenario

Maintenance time is a crucial concept for TencentDB for Redis®. To ensure the stability of your TencentDB for Redis® instance, the backend system periodically performs maintenance operations and troubleshooting on the instance during the maintenance time. We recommend setting an acceptable maintenance time for your business instance, typically during off-peak hours, to minimize the impact on your operations.
Additionally, it is recommended to perform operations involving data migration, such as instance version or instance architecture upgrades, during the maintenance time. For example, when upgrading a database instance version, full and incremental data must be synchronized from the lower version instance to the new version instance, involving data migration. This will cause a momentary database connection disconnection upon completion. By selecting **Switch Time** as **During Maintenance Time** when initiating the upgrade, the instance version switch will be initiated within the next **Maintenance Time** after data synchronization is completed. Please note that when choosing **During Maintenance Time** as the switch time, the database version upgrade will not switch immediately but will maintain synchronization until the instance's **Maintenance Time** to initiate the switch, which may prolong the overall instance upgrade duration.

> **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. Please be attentive to these messages.

Release notes

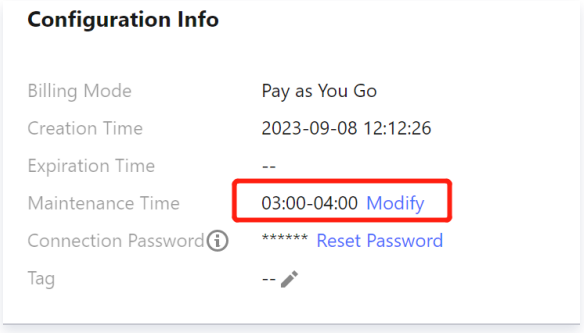
The default maintenance time is from 03:00 to 04:00 AM. All current Redis versions support adjusting the maintenance time according to your business's off-peak hours.

Preparations

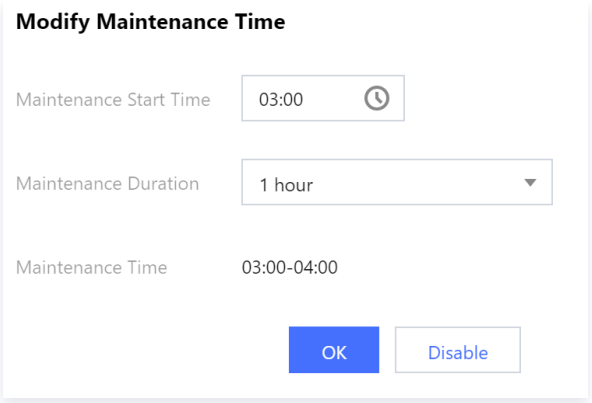
- You have created a TencentDB for Redis® instance.
- The instance is in **Running** status.
- The instance maintenance time needs to be adjusted.

Instructions

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 **Instance Details** page, click **Modify** on the right of **Maintenance Time** in the **Configuration** section.



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



Parameter name	API 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 **Confirm** to complete the configuration.

Related APIs

Related APIs	API Description
DescribeMaintenanceWindow	This API is used to query the instance maintenance window.
ModifyMaintenanceWindow	Modifies instance maintenance time

Changing instance specifications

Last updated: 2024-11-01 16:20:45

Scenario

TencentDB for Redis® supports adjusting instance specifications, offering flexible scaling operations. You can adapt the Redis instance specifications based on your business's actual needs, thereby better satisfying resource utilization and real-time cost optimization requirements. This document describes the process of modifying instance specifications in the TencentDB for Redis® console.
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, and all replicas participate in supporting the system's high availability. Therefore, the more replicas, the higher the availability. When 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 assigning different keys to multiple shard nodes in the sharding mode of instances in cluster architecture to adjust the number of shard nodes, so that the system performance can be horizontally scaled.

Release notes

- 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

- Upgrade configuration: The system will calculate the price difference between instance specifications and deduct it from your cloud account. Subsequently, you will be charged according to the new specifications. If your account balance is insufficient, you will need to recharge it first.
- 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 the original configuration: Effective order amount of the original configuration – used value of the original configuration.
    - Effective order amount of the original configuration: The amount paid for the currently effective order, excluding discounts and vouchers.
    - Residual value of 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 price of the new configuration: Current official price of the new configuration * 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.

Preparations

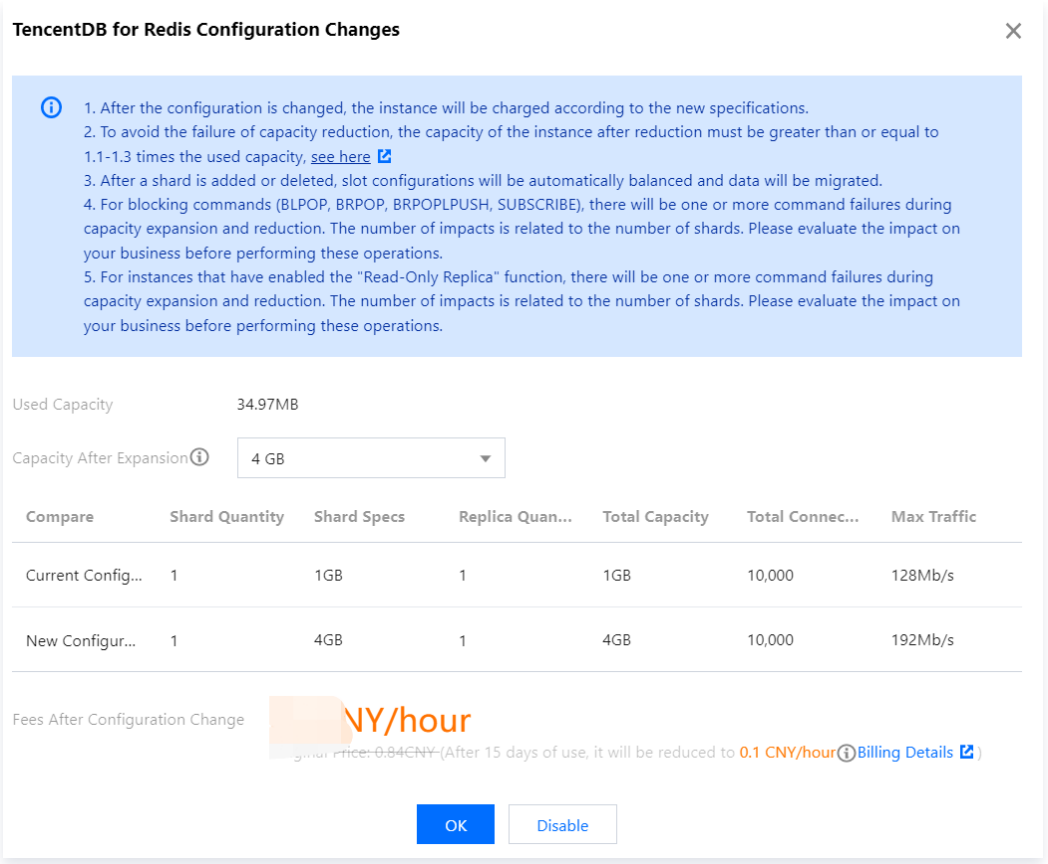
- You have created a TencentDB for Redis® instance.
- The instance and its associated instances are in **Running** status and are not executing any tasks.
- You have calculated the required specifications and understood the fees. Make sure that your Tencent Cloud account balance is sufficient.

Redis Memory Edition (Standard Architecture)

Note:

- After the configuration is adjusted, the instance will be charged at the price of the new specification.
- 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, perform actions such as expanding or reducing nodes, and adding or removing 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 for reduction. The configuration parameters for reducing nodes are similar to those for expanding nodes. The **Capacity After Reduction** refers to the capacity specification of each shard after reduction. The instance capacity after reduction must be greater than or equal to 1.3 times the used capacity. Please compare the capacity specifications before and after reduction to confirm if they meet the requirements.



Parameter	Description
Used Capacity	The used capacity of the current instance.
Min Memory	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.
Comparison	Compare the current configuration with the new configuration, including **Shard Quantity**, **Shard Specs**, **Replica Quantity**, **Total Capacity**, **Total Connections**, and **Max Traffic**.
Configuration Change Fees	This parameter will be displayed if the billing mode is yearly/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.

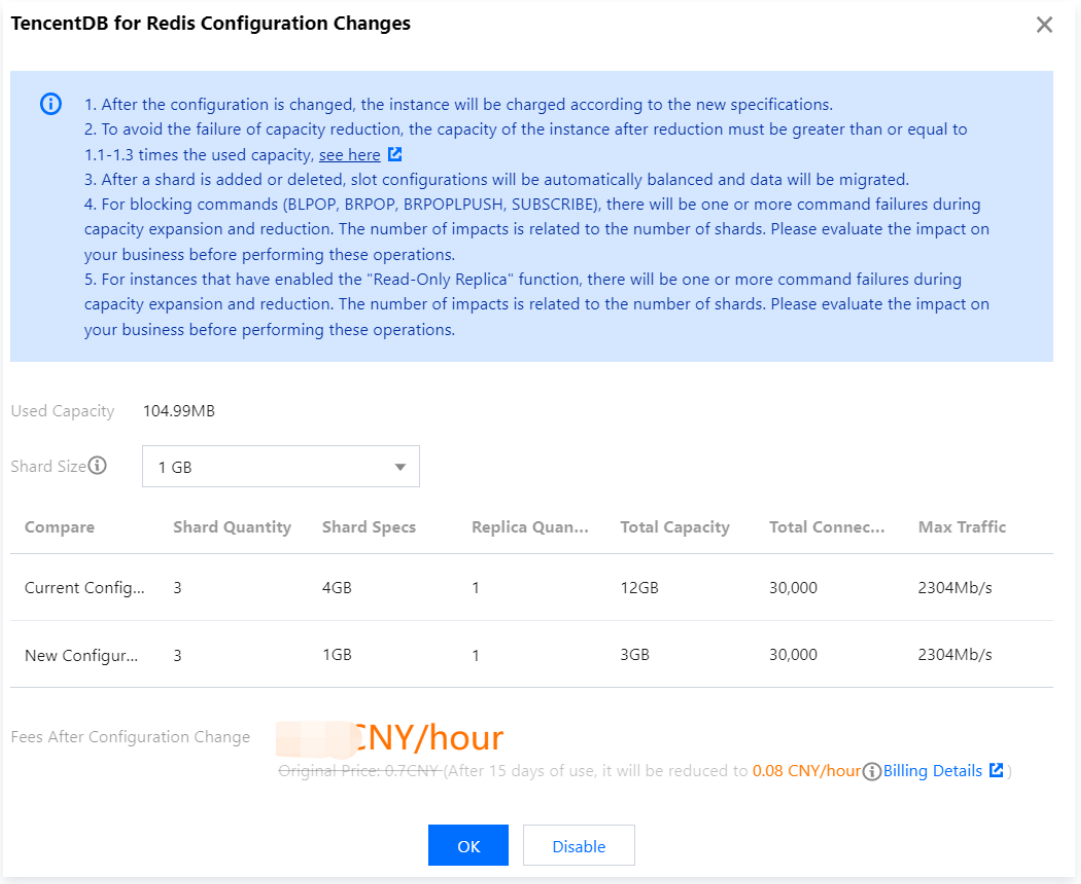
- Select **Configure** > **Add Replica** to enter the **TencentDB for Redis® Configuration Changes** page. Select the desired number of replicas in the drop-down list next to **Replica Quantity**. Other parameters are similar to those for node expansion. For specific operations on adding replicas to multi-Availability Zone instances, please see Adding Replicas to Multi-AZ Deployed Instances.
- 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 specification.
> - 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. It is recommended to perform such operations during off-peak period so as 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, perform actions such as expanding or reducing nodes, adding or removing replicas, and increasing or decreasing 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 reduce the capacity of each shard node. The configuration parameters for reducing nodes are similar to those for expanding nodes. The **Shard Size** refers to the capacity specification of each shard after reduction. The capacity of the instance after reduction must be greater than or equal to 1.3 times the used capacity. Please compare the capacity specifications before and after reduction to confirm whether they meet the requirements.



Parameter name	Description
Used Capacity	The used capacity of the current cluster instance.
Min Memory	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.
Comparison	Compare the current configuration with the new configuration, including **Shard Quantity**, **Shard Specs**, **Replica Quantity**, **Total Capacity**, **Total Connections**, and **Max Traffic**.
Configuration Change Fees	This parameter will be displayed if the billing mode is yearly/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 specific operations on adding replicas to multi-Availability Zone instances, please see Adding Replicas to Multi-AZ Deployed Instances.
- 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	Feature
UpgradeInstance	Upgrades the instance configuration

Adjusting the Number of Connections

Last updated: 2024-11-01 16:21:06

Scenario

When there are a large number of concurrent application requests, the current database specifications may not be sufficient to meet the demand, resulting in an insufficient number of connection configurations and a potentially high **Connection Utilization** metric. You can directly increase the number of connections in the console to cope with sudden demand peaks.

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:

- **Disabling Read-Only Replica**
  The value range for the maximum number of connections per shard is [10,000, 40,000].
- **Enabling Read-Only Replica**
  The value range for the maximum number of connections per shard is [10,000, 10,000 x (number of replicas + 3)].

Supports and Limits

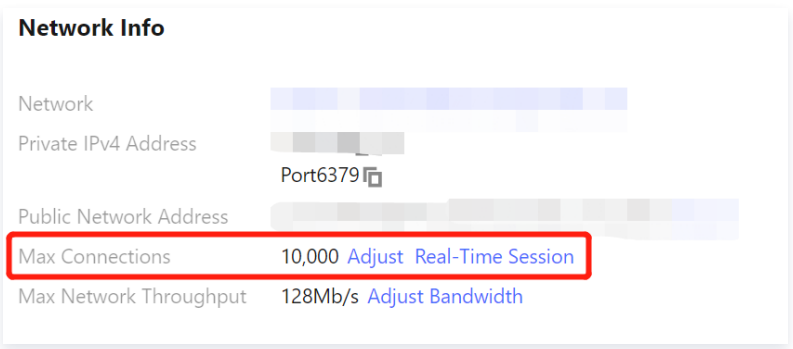
- Increasing the maximum number of connections has no impact on the business.
- If the maximum number of connections is decreased, new connections may fail to be established when the number of connections reaches the upper limit.
- If increasing the number of connections does not resolve your issue, please contact customer support or submit a ticket for assistance.

Preparations

- You have created a TencentDB for Redis® instance.
- The database instance is in **Running** status, with no ongoing tasks.

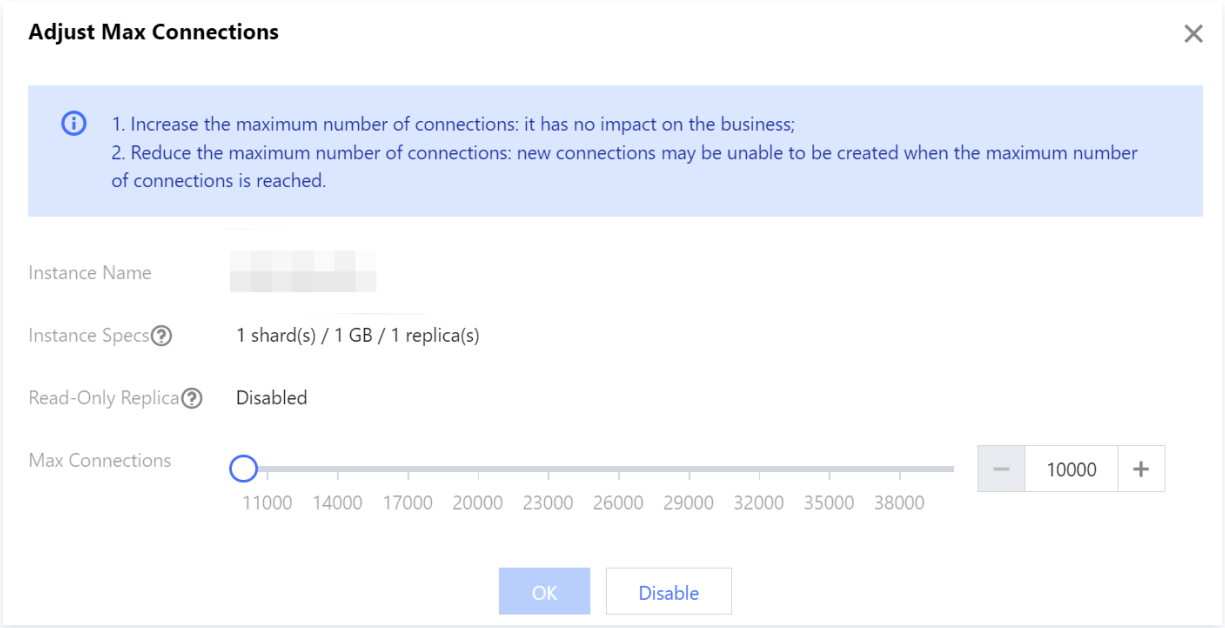
Instructions

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 **Instance Details Network Information** section, you can view the current maximum number of connections after **Maximum Connections**. Click **Adjust**.



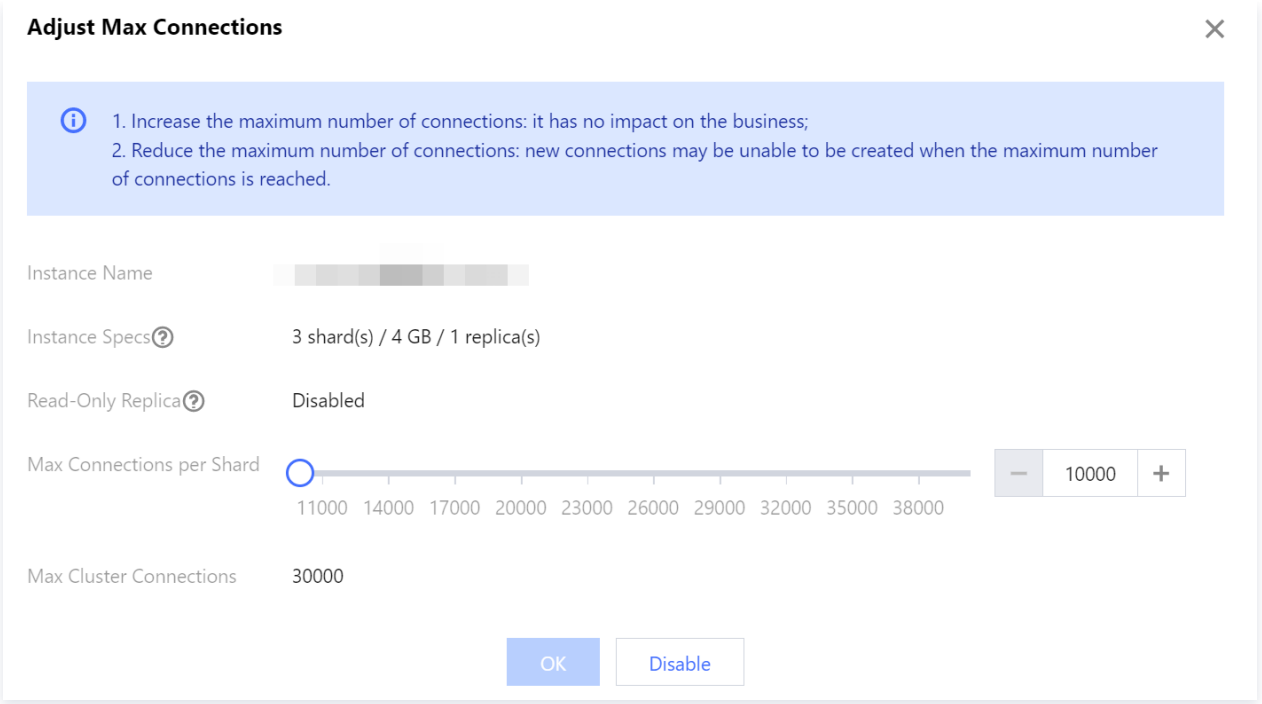
6. In the **Adjust Maximum Connections** dialog, confirm the instance information and specifications, and increase the number of connections.

- **Standard Architecture**



Parameter name	Description
Instance Name	The instance name.
Instance Specification	The instance configuration information includes: shard quantity/total memory capacity/replica quantity. The standard architecture has one shard.
Read-Only Replica	Specify whether to enable read-only replicas, that is, whether to enable read/write separation.
Max connections	Adjust the maximum number of connections on the slider bar.

- **Cluster Architecture**



Parameter name	Description
Instance Name	The instance name.
Instance Specification	The instance configuration information includes: shard quantity/shard capacity/replica quantity.
Read-Only Replica	Specify whether to enable read-only replicas, that is, whether to enable read/write separation.
Maximum number of connections	Adjust the maximum number of connections per shard using the slider.
Max Cluster Connections	The maximum number of connections for the entire instance is automatically calculated, which is the maximum number of connections per shard multiplied by the number of shards.

7. Click **Confirm**, then click **Task Management** in the left navigation to view the task execution progress. Wait for the task to complete, and you can view the adjusted maximum number of connections for the entire instance in the **Instance Details** page under the **Network Information** section, next to **Max Connections**.
确认是OK 还是confirm
确认是OK 还是confirm

Related APIs

API	Related APIs
describeInstances	Viewing Instance Information List
ModifyConnectionConfig	Frequently Asked Questions

Clearing Instances

Last updated: 2024-11-01 16:21:22

Scenario

TencentDB for Redis® supports quickly clearing all instance data in the console. By performing the **FLUSHALL** operation on the instance, all its data will be thoroughly removed and cannot be restored. Proceed with caution.

Supports and Limits

- Once cleared, the data cannot be recovered. Please make sure you have backed up all data before submitting the 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.

Instructions

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 **Confirm**.
7. On the left sidebar, select **Task Management** and wait for the task to complete.

Related APIs

API Name	Feature
ClearInstance	Clears a Redis instance

Returning and Isolating Instance

Last updated: 2024-11-01 16:21:41

Scenario

- Before the expiration of a prepaid yearly/monthly subscription instance, if you no longer need the instance resources and wish to return the instance, you can initiate the return process directly in the console and settle the refund amount. Returned instances are moved to the recycle bin and retained for 7 days. Within these 7 days, you can restart the instance, renew the subscription, and restore the instance resources. If no renewal action is taken after 7 days, the system will automatically destroy the resources, and all data will be erased and cannot be recovered.
- 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.

> **Note:**
> - After the instance is terminated, all data will be cleared and cannot be recovered. Please be sure to back up your data first before submitting a termination task.
> - When an instance is terminated, its IP address will be released.

Refund Policy

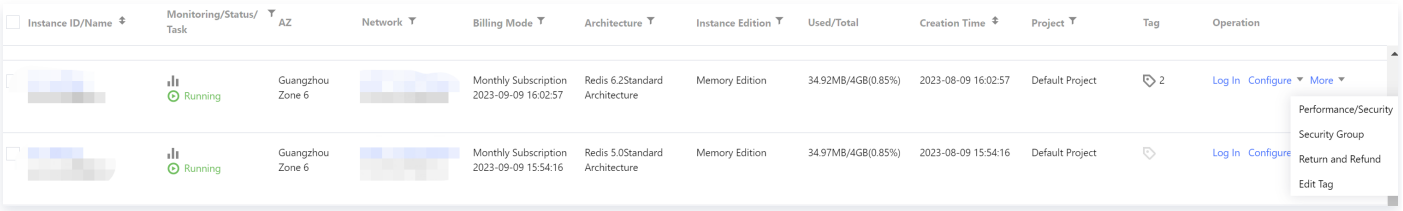
After an instance is terminated, the refund procedures are as detailed below:

- The 5-day free returns will be refunded to your Tencent Cloud account.
- For normal monthly-subscribed instances, the payment will be returned to your Tencent Cloud account by the proportion of the cash and gift cards paid for the purchase after the termination.
- 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 calculation rules and examples, see Refund Policy.

Returning a Yearly/Monthly Subscription Instance

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, select the monthly subscription instance you want to terminate. In the **Action** column, choose **More** > **Return and Refund**.

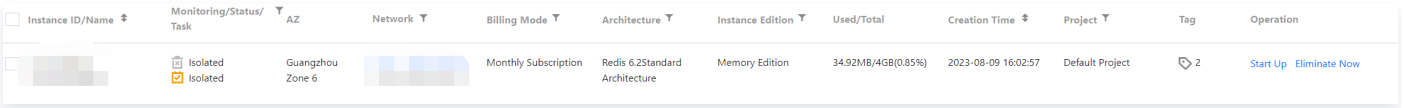
> **Note:**
> When the **Return and Refund** button is unavailable, it indicates that the account has reached the limit for monthly subscription self-service returns, meaning that the monthly subscription instance cannot be manually terminated and will be automatically destroyed upon expiration.



4. In the **Return and Refund** dialog box, click **View Details** to confirm the information of the instance to be returned and understand the precautions for terminating instances. Click **Termination Rules** to learn about the refund calculation method, select **I have read and agree**, and click **View Refund Details**.
5. On the **Please confirm the refund information** page, review the refund details, including: instance information, network information, region and availability zone, refund method, and verify the refund amount. Click **Confirm Refund**.
6. In the pop-up dialog box, please double-check the refund method and amount, then click **Confirm Refund**.
7. On the **Submission Successful** page, you can perform the following operation.

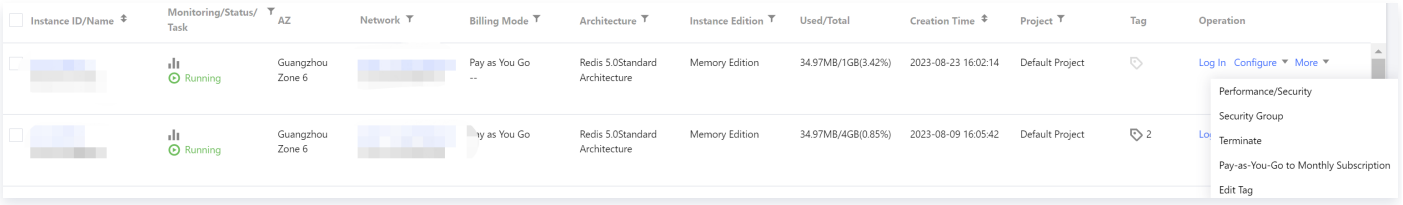
- Click **View Refund Progress** to enter the **Billing Center Order Management** page. You can track the refund progress in the **Prepaid Orders** tab. As shown below, click **Details** in the **Action** column to view the detailed information of the order. Click **Income and Expense Details** in the right navigation bar to check the refund status.

- Click **Enter Product Management Console**, and on the left sidebar, select **Recycle Bin**. You can see that the returned instances are isolated in the recycle bin, with the status **Isolated**. Click **Start** to restore the instance, or click **Terminate Now** to destroy the instance resources. For detailed instructions, see Recycle Bin.

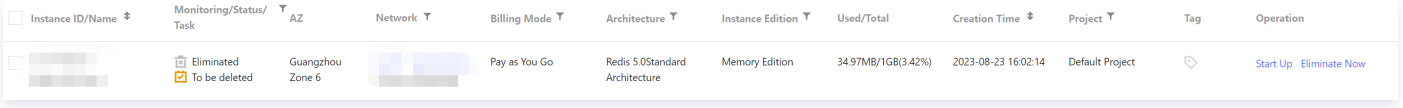


Terminating a pay-as-you-go instance

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, select the pay-as-you-go instance you want to terminate, and in the **Action** column, choose **More** > **Terminate**.



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** to restore the instance, or click **Eliminate Now** to directly terminate the instance resources. For detailed instructions, please see Recycle Bin.

Related APIs

API Name	Feature
DestroyPostpaidInstance	Terminates a pay-as-you-go instance.

DestroyPrepaidInstance	Returns a monthly subscribed instance.

Restoring Isolated Instance

Last updated：2024-11-01 16:22:00

Scenario

Tencent Cloud Recycle Bin is a cloud service recycling mechanism. When you return an instance, a yearly/monthly subscription instance expires, 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. Yearly/monthly subscription and pay-as-you-go instances have different recycling mechanisms, but both support the restoration of instance resources during the retention period.

Yearly/Monthly Subscription Recycling Mechanism

Before a yearly/monthly prepaid instance expires, if you no longer need the instance resources, you can actively return the instance to the recycle bin for isolation. If you do not renew the yearly/monthly prepaid instance in time after it expires, the system will automatically move it to the TencentDB recycle bin. Instances isolated in the recycle bin will no longer incur fees and will be retained for 7 days. Within these 7 days, you can restart the instance, renew it, and restore its resources. After 7 days, if you still haven't renewed the instance, the system will automatically terminate the resources, and all data will be cleared and become irrecoverable. For more information, please see the Recycling Mechanism in the overdue payment description.

Pay-as-you-go instance repossession mechanism

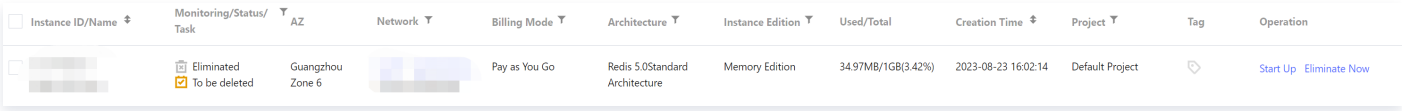
- When using pay-as-you-go postpaid instances and your Tencent Cloud account has no overdue payments, if you no longer need the instance resources, you can actively return the instance to the recycle bin for isolation. To return an instance, please see Returning and Isolating Instances. Terminated instances are retained in the recycle bin for two hours, and you can start them up to restore them within this period. After two hours, the system will directly eliminate the instance resources, and all their data will be permanently deleted and cannot be recovered.
- For pay-as-you-go postpaid instances, when your Tencent Cloud account balance is less than 0, you can continue to use and be charged for the instance within 24 hours. After 24 hours, the instance will be automatically moved to the recycle bin for isolation, and billing will stop, making the instance resources unavailable. Under overdue payment, the retention period for instances in the recycle bin is 24 hours. If you top up your account within 24 hours, you can restore the instance resources. After 24 hours, if you haven't topped up your account, the system will automatically terminate the resources, and all data will be cleared and cannot be recovered.

Preparations

- 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. 单击列表上面的续费，在续费所选实例对话框，您需要执行如下操作。



5. In the instance list in the recycle bin, find the target instance you want to restore, and click **Start** in the **Operation** column.
6. Confirm the instance information and restore it.
   - For yearly/monthly subscription instances, in the **Renew Selected Instances** dialog box, you need to perform the following steps: click **OK** to restore the instance.

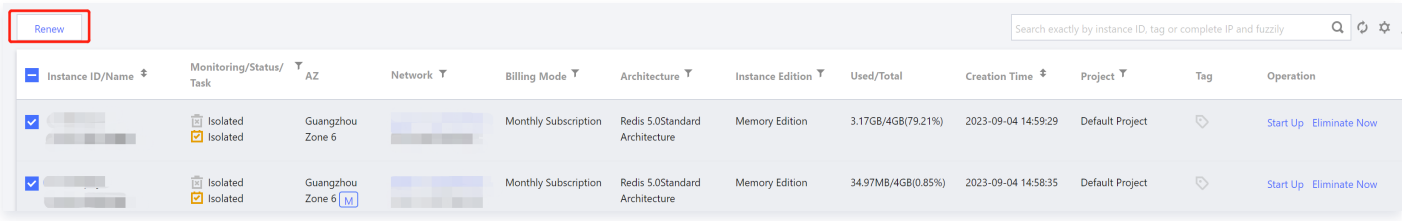
- Click **View Details** to confirm the instance information.
- In the **Renewal Period** section, select the desired renewal duration for the instance, either in whole months or years.
- Confirm the total renewal amount for the instance next to **Total Cost**.

- For pay-as-you-go instances, in the **Start Instance** dialog box, confirm the information of the instance you want to restore, and click **OK** to restore the instance.
  这一部分都需确认页面

> **Note**
> Due to the in-memory database characteristics of Redis, if you need to use the batch restore instance feature, please submit a request through online customer service.

Restoring multiple isolated instances

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. 单击列表上面的续费，在续费所选实例对话框，您需要执行如下操作。
5. In the instance list in the recycle bin, select the target instances.



6. Click **Renew** at the top of the list, and in the **Renew Selected Instances** dialog box, you need to perform the following actions.
   - Click **View Details** to confirm the instance information.
   - In the **Renewal Period** section, select the desired renewal duration for the instance, either in whole months or years.
   - Confirm the total renewal amount for the instance next to **Total Cost**.
7. 接口含义

Related APIs

API Name	恢复已隔离实例
StartupInstance	Deisolates instance

Eliminating Instance

Last updated: 2024-11-01 16:22:35

Scenario

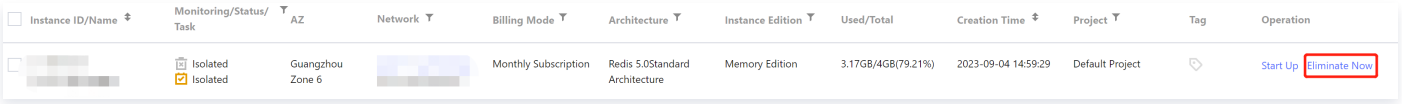
When you return an instance, your monthly subscribed instance expires, or your account balance is insufficient to pay the fees of a pay-as-you-go instance, the instance will be moved to the recycle bin. If you have backed up your data and you are sure that you don't need the instance any more, you can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

Preparations

- The instance is isolated in the recycle bin, and the data has been backed up.
- The instance is no longer needed.

Instructions

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** to destroy the instance resources directly.

> **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	API Meaning
CleanUpInstance	This API is used to eliminate an instance in the recycle bin immediately.

Instance Upgrade
Upgrading Instance Version

Last updated: 2024-11-01 16:23:36

Scenario

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.

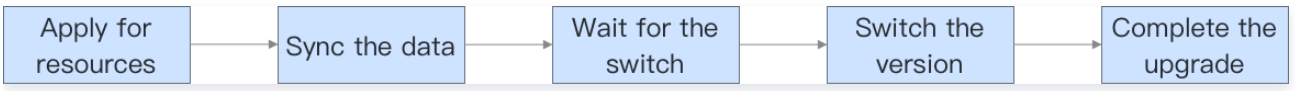
Version Differences

Compatible Version	Minor Version	Optimizations and Fixes
Redis 4.0	4.3.0	• Supported multiple databases, with up to 256 databases available. • 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	• Supported multiple databases, with up to 256 databases available. • Performance optimization: `zmalloc_get_rss()` is executed in the BIO thread to avoid blocking the main thread and increasing the request latency. • Fixed the rdbLoadRio() function to prevent potential crashes in certain scenarios.
Redis 6.2	6.2.5	• Supported Redis 6.2. • Supported multiple databases, with up to 256 databases available.

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 will automatically detect 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 version: When the switch conditions are met (data sync is almost completed, and the requirements for the switch time 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.

Impact of Upgrade

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

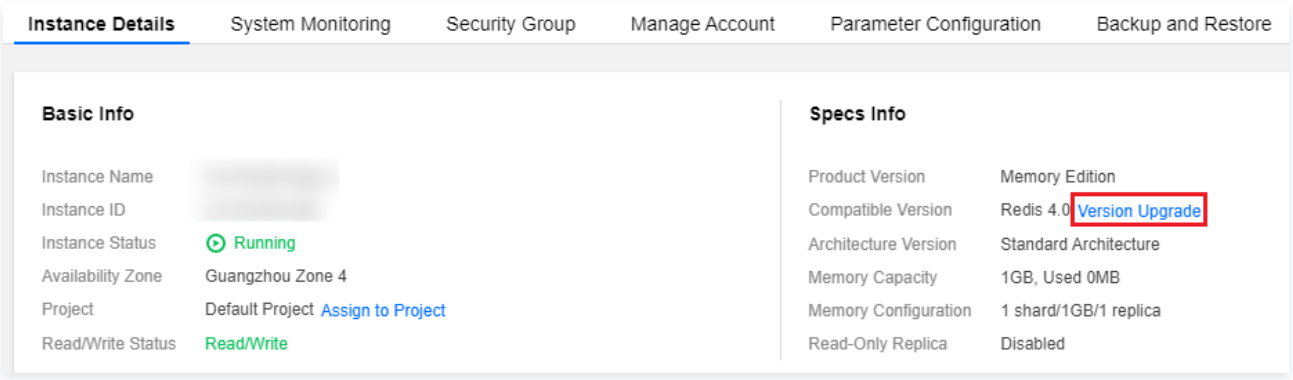
- During data sync, the service will not be affected.
- During switch, the instances will become read-only for less than 1 minute (to wait for the data sync completion), and a momentary disconnection (within seconds) will occur; therefore, your business should have an automatic reconnection mechanism.

Preparations for Upgrade

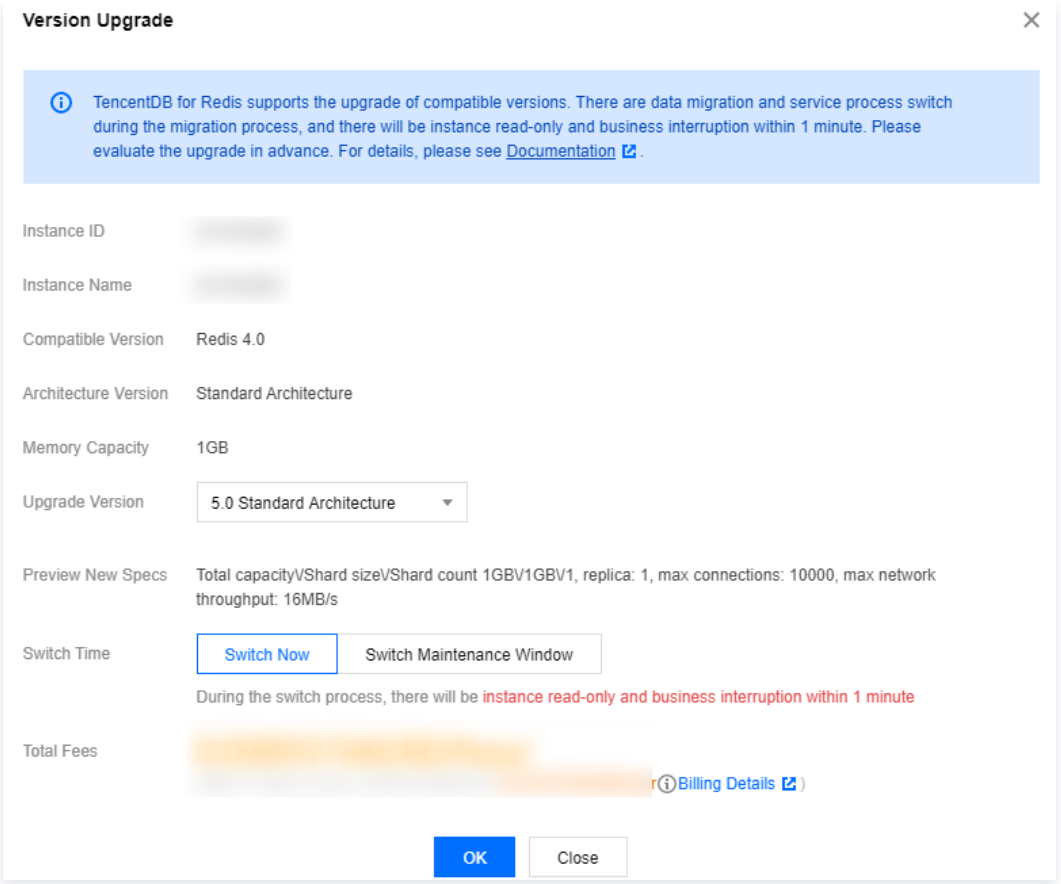
- The instance to be upgraded is in **Running** status and is not executing any tasks.
- The target version is confirmed.

Upgrading the version

1. Log in to the TencentDB for Redis® console.
2. Above the **Instance List** 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 **Version Upgrade** after **Compatible Version**.



6. In the pop-up dialog, confirm the information of the instance to be upgraded based on the table below, configure the target upgrade version, and click **OK**.



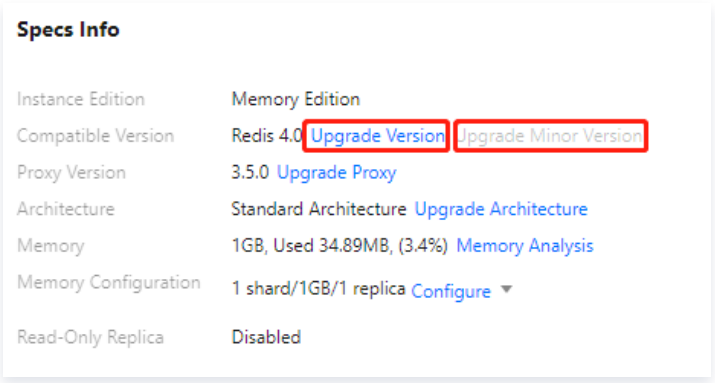
Parameter name	Description
Unique instance ID	ID of the instance to be upgraded.
Instance Name	Name of the instance to be upgraded.
Compatible Version	The current compatible Redis version of the instance to be upgraded.
Architecture Version	Architecture information of the instance to be upgraded. Currently,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

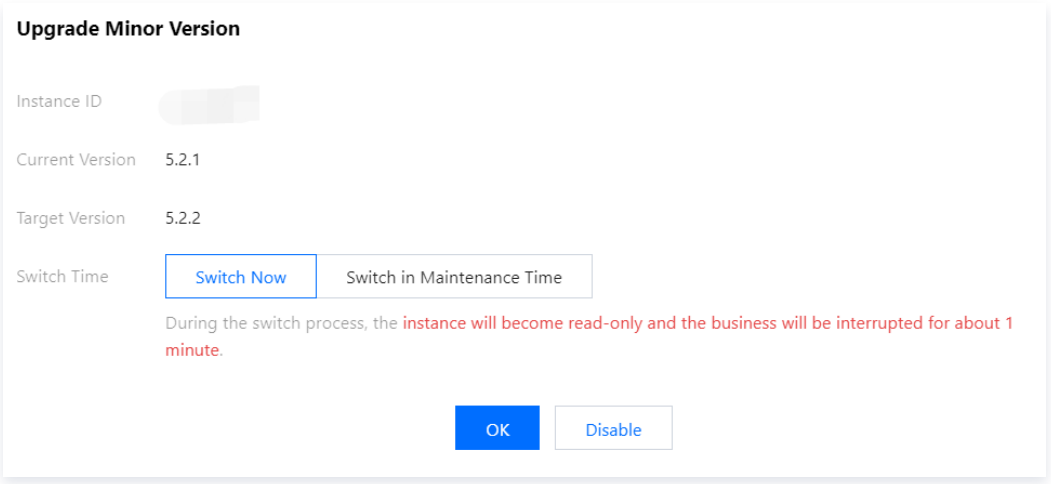
Upgrading Minor Version

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 Minor Version** after **Compatible Version**.

> **Note:**
> 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** dialog, confirm the instance information and the version to be upgraded, then switch the time and select the upgrade 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 Name	Feature
UpgradeInstanceVersion	Upgrading Instance Version
UpgradeSmallVersion	Upgrades the minor version of an instance

Upgrading Instance Architecture

Last updated: 2024-11-01 16:23:56

Scenario

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) within 3 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 multi-key access. For Cluster Architecture, commands can be categorized into supported, custom, and unsupported. For a detailed command compatibility list, see Command Compatibility.
- For more information about compatibility check, see Check on Migration from Standard Architecture to Cluster Architecture.

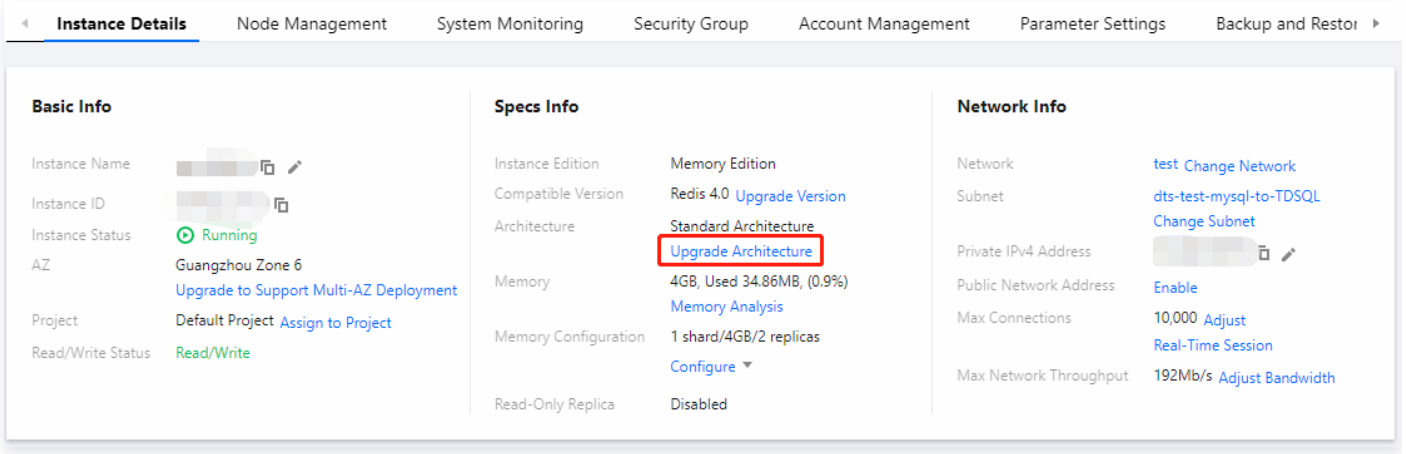
Impact of Upgrade

- 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.

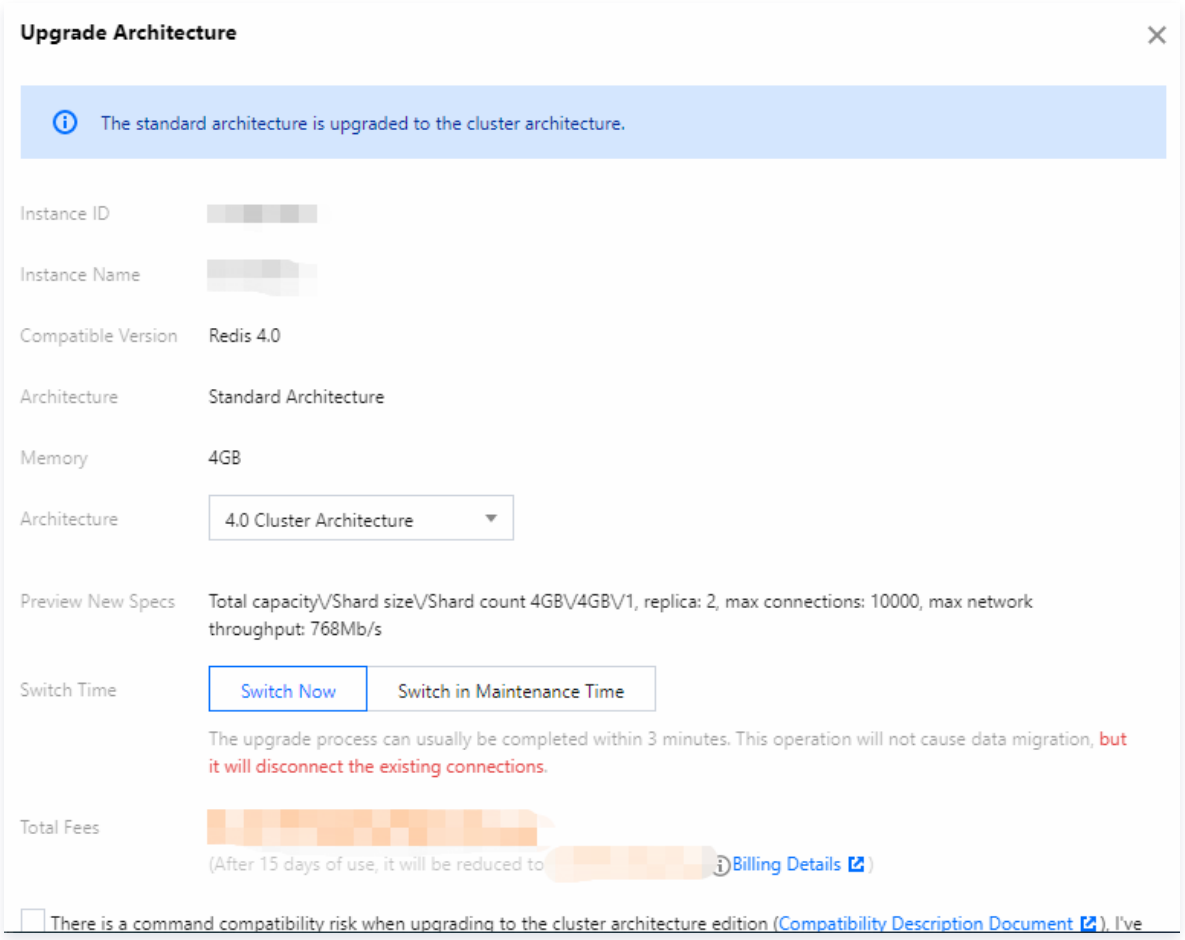
Instructions

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 Version**.



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



Parameter	Description
Instance ID	ID of the instance to be upgraded.
Instance Name	Name of the instance to be upgraded.
Compatible Versions	The current compatible Redis version of the instance to be upgraded.
Architecture	Architecture information of the instance to be upgraded.

Memory Capacity	Memory size of the instance to be upgraded.
Architecture	Select the target architecture version in the drop-down list. Currently, only upgrading from standard architecture to cluster architecture is supported, and downgrading from cluster architecture to standard architecture is not supported.
New Specification Preview	Preview information of the instance specifications after upgrade.
Switch Time	• **Switch Now**: Perform the switch action immediately. • **Switch in Maintenance Time**: The switch will be performed during the instance maintenance time. You can modify the **Maintenance Window** on the instance details page. It is recommended to perform the operation during off-peak business hours.
Total Cost	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 version. Click **Compatibility Documentation**, confirm the compatibility risk, select **There is a command compatibility risk when upgrading to the cluster architecture version (Compatibility Documentation), I have confirmed the compatibility risk and will proceed with the upgrade,** and click **OK** to continue the upgrade.
8. On the product order page, verify the order information and the required payment. Hover over to understand the detailed calculation of the payment. Once confirmed, click **Submit Order**. After the payment is completed, return to the instance list. Once the instance status changes to **Running**, you can verify in the instance list or instance details that the instance's architecture version has been upgraded to the cluster architecture.

Related APIs

API Name	Feature
UpgradeInstanceVersion	Upgrades the instance architecture

Upgrading Proxy

Last updated: 2024-11-01 16:24:18

Scenario

TencentDB for Redis® periodically releases minor versions of the proxy to enhance database features or fix known defects.

Proxy Version	Proxy Minor Version	Addition, Optimization, and Fixation
Proxy 5.0	5.6.3	Added getex, auth name pwd, and xautoclaim commands.
	5.6.0	• The cluster architecture supports the 'wait' command. • Supported SSL encryption to implement encrypted data transfer. • Optimized the performance of `cluster nodes/slot` command execution.
	5.5.0	• The cluster architecture supports the 'wait' command. • 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/flushdb commands, which can be distributed to all primary shard nodes in the cluster architecture while retaining the specified nodeid parameter. • Supported monitoring metrics for large value request counts. • Supported the Scan command in Cluster Edition instances to traverse all shards. • Fixed the issue where executing the `select` command after a transaction might result in the return of `ERR unknown command 'select' command`. • Fixed the issue that using watch+ transaction in pipeline scenarios caused locked connections to be released in a delayed manner, resulting in commands being sent to the wrong node and triggering a Move error.
	5.4.0	Optimized the statistical strategy for P99 monitoring indicators, including 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. • Fixed the error "ERR MULTI calls cannot be nested."
	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 error "ERR MULTI calls cannot be nested."

Notes on Upgrade

- The system automatically detects the proxy's minor version. If the **Upgrade Proxy** button is grayed out, the instance 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.

Impact of Upgrade

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

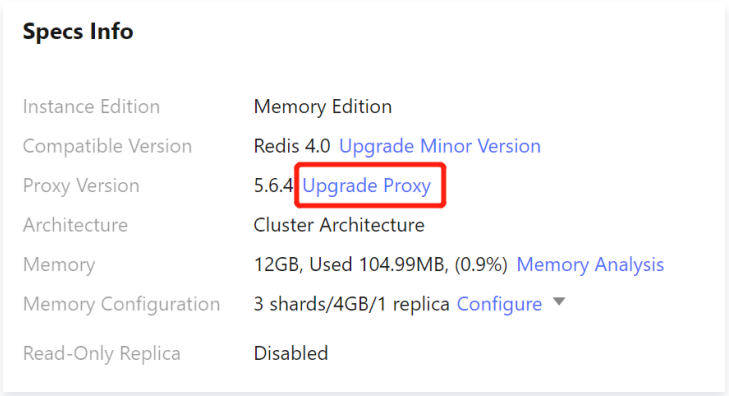
- During data sync, the service will not be affected.
- During switch, the instances will become read-only for less than 1 minute (to wait for the data sync completion), and a momentary disconnection (within seconds) will occur; therefore, your business should have an automatic reconnection mechanism.

Preparations for Upgrade

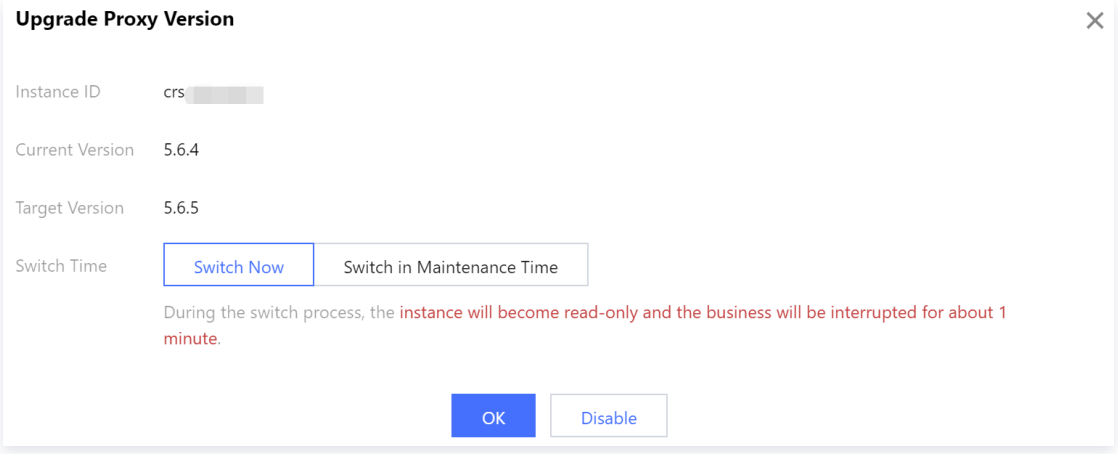
- The instance to be upgraded is in **Running** status and is not executing any tasks.
- We recommend that you perform upgrade in the maintenance time during off-peak hours.

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 **Specification Information** section on the **Instance Details** page, click **Upgrade Proxy** after **Proxy Version**.



6. In the pop-up dialog, confirm the information of the instance to be upgraded based on the table below, configure the target upgrade version, and click **OK**.



Parameter name	Description

Unique instance ID	ID of the instance to be upgraded.
Current Version	Current minor version of the Proxy.
Target version	The target version after the proxy upgrade. Selecting a target version is not supported.
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. Once the instance status changes to **Running**, you can verify that the instance version has been upgraded in the instance list or instance details.

Related APIs

API Name	Feature
UpgradeProxyVersion	Upgrading Proxy Version

Node Management
Viewing Node Information

Last updated: 2024-11-01 16:25:27

Scenario

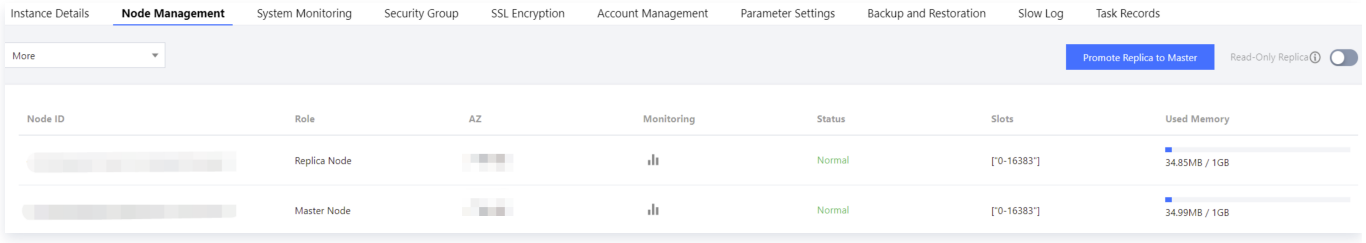
TencentDB for Redis® allows you to view instance node information, including node ID, role, operational status, and capacity usage. It also supports node management, such as adjusting node specifications, promoting replica nodes to master nodes, enabling read-only replicas, and configuring master/replica failover. Operations personnel can efficiently manage instance nodes and identify related anomalies during node operation through node management.

Release notes

- TencentDB for Redis® versions 4.0, 5.0, and 6.2, whether deployed in a single availability zone or multiple availability zones, support node management for both standard and cluster architecture instances.
- TencentDB for Redis®

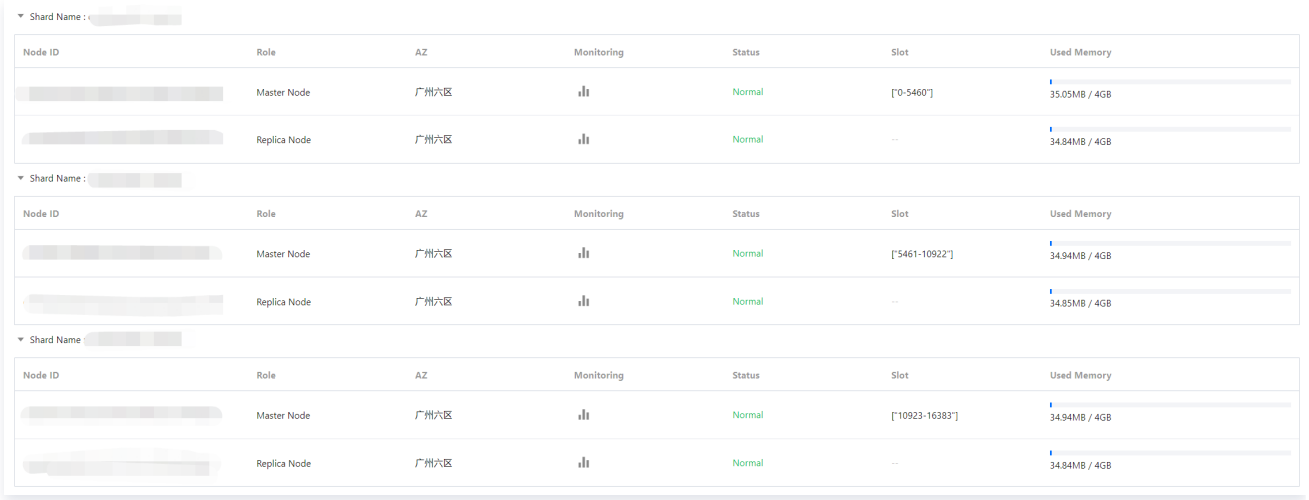
Viewing node information

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.
   - **Standard Architecture**

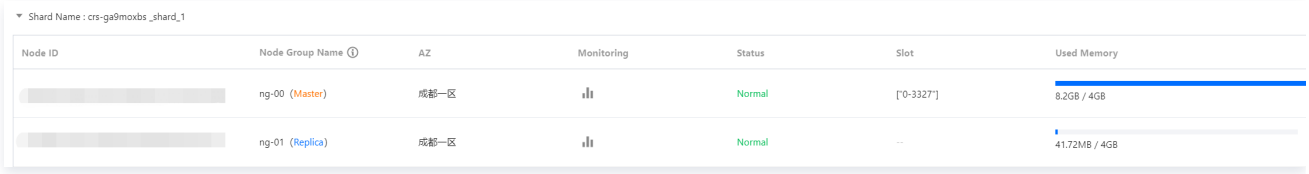


Parameter	Description
Node ID	Database instance node ID.
Role	The current node's role, either master or replica node.
Availability Zones	AZ of the current node.
Monitoring	Click to view the monitoring metrics of the node in the right-side monitoring panel. For more information, see Monitoring Features.
Status	Status of the current node.
Slots	The value range of Slots on a node.
Memory usage	Node memory capacity usage.

   - **Cluster Structure**
     - **Single Availability Zone**



- **Multi-AZ**

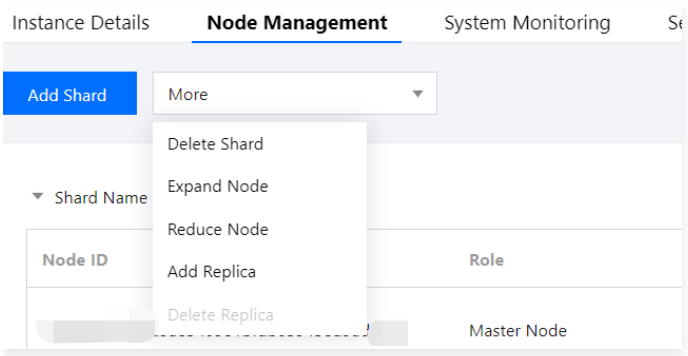


Parameter	Description
Shard Name	The shard name in a sharded database cluster is composed of the instance ID, the word "shard", and a shard number identifier, concatenated with underscores. The shard number identifier starts from 1 and follows a natural number sequence. For example: crs-mufy****_shard_1.
Node ID	The node ID of the database instance is a randomly generated unique identifier assigned by the system to distinguish each node.
Node Group Name	The cluster architecture with multiple availability zones displays this parameter. • In a multi-shard setup, master nodes logically form a master node group, while replica nodes logically form a replica node group. • Each node group name is composed of 'ng-' followed by a number, starting from 00 and arranged in natural number order. • Each node group name indicates the master and replica roles of the nodes.
Role	In a cluster architecture with a single availability zone, this parameter is displayed, indicating the master and replica roles of the node.
Availability Zones	The node belongs to an AZ. For instances deployed in multiple availability zones, the master and replica availability zones will be indicated.
Monitoring	Click to view the monitoring metrics of the node in the right-side monitoring panel. For more information, see Monitoring Features.
Status	Status of the current node.
Slots	The value range of Slots on a node.
Memory usage	Node memory capacity usage.

More Operations

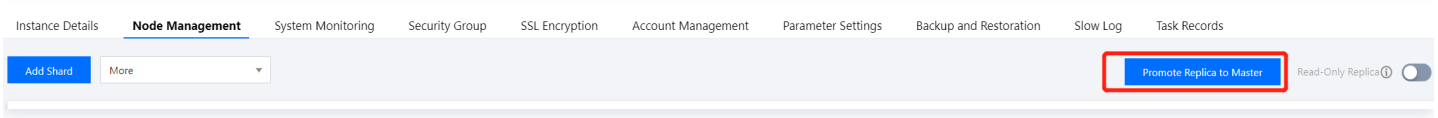
Configuration Modification

On the **Node Management** page, you can adjust the instance configuration specifications, including expanding/reducing node memory capacity, adding/deleting replicas, and adding/deleting shards (for cluster architecture). The entry point is shown in the figure below. To access the configuration change window and learn how to configure, please see Changing Instance Specification.



Promoting Replica Nodes to Master Nodes

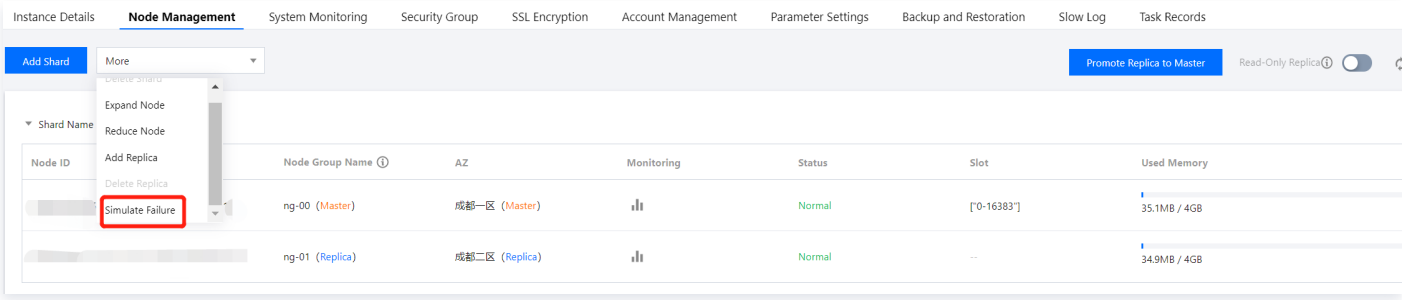
For single-AZ deployed instances, TencentDB for Redis® supports manually promoting replica nodes to master nodes. For multi-AZ instances, TencentDB for Redis® supports promoting replica nodes to master nodes, and in cluster architecture, promoting replica node groups to master node groups, elevating the AZ of the source replica node or replica node group to the master AZ. For detailed instructions, please see Manually Promoting Master Node.



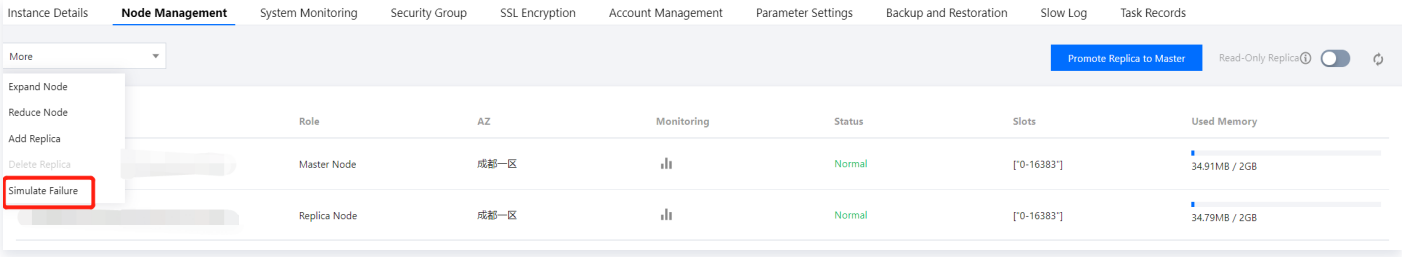
Simulating Failure

To facilitate fault simulation testing for your business, TencentDB for Redis® provides a fault simulation feature for multi-AZ deployed instances. You can use this feature on the **Node Management** page in the console. For detailed instructions, please refer to Fault Switching.

- **Cluster Architecture** instance fault simulation entry:

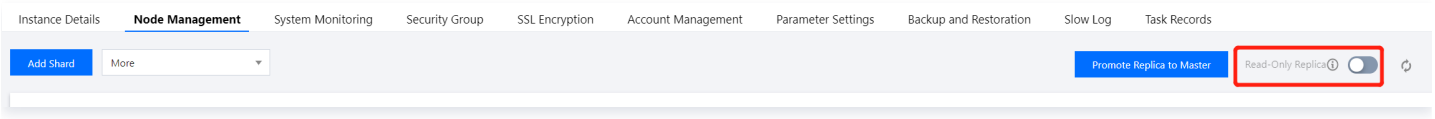


- **Standard architecture** instance fault simulation entry:



Read-Only Replica

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



Related APIs

Related APIs	API Description
DescribeInstanceNodeInfo	Queries instance node information
DescribeInstanceZoneInfo	Querying Redis node details

Promote Replica to Master

Last updated: 2024-11-01 16:25:58

You can deploy the master node to a specified availability zone (AZ) or replica (group) based on your business requirements. The principle of promoting a replica node to a master node (group) in a TencentDB for Redis® instance is to execute the 'cluster failover' command within the cluster, which switches the role of the master node to the replica node.

Promotion in Standard Architecture

In a standard architecture instance, there can only be one master node. You can promote a specified node to the master node using the promotion feature. For a multi-AZ deployed instance in the standard architecture, switching the master node may cause cross-AZ access, resulting in increased latency and a decrease in QPS.

Promotion in Cluster Architecture

In a cluster architecture instance with multiple replicas per shard, TencentDB for Redis® divides the master node and multiple replicas into different node groups for easier node management. Each node group has a name. For a multi-AZ deployed instance in the cluster architecture, you can promote all nodes within a specified node group to the master node group. If the master nodes of some shards are switched to other node groups, you can use the promotion feature to switch all master nodes back to the specified node group. The execution process is as follows:

1. Promote all nodes within the node group to master nodes.
2. Promote the node group to master node group. If a server-level failover occurs after the promotion is completed, some of the nodes in the node group will be demoted to replica nodes and promoted back to master nodes.
3. Promote the AZ to master AZ.

Supports and Limits

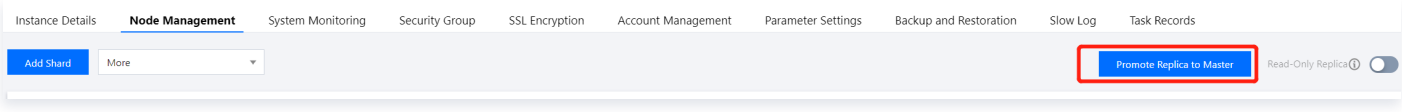
- Promoting nodes will trigger a master-replica switch. During the switch, database access will be affected for few seconds to 3 minutes, and blocking commands (including BLPOP, BRPOP, BRPOPLPUSH, and SUBSCRIBE) may fail one or more times before they can be executed successfully.
- During the replica promotion process, the instance will be read-only for up to 1 minute, and there may be a momentary disruption to the business. The execution process may fail, but it can usually be resolved by retrying.

Preparations

- The instance has been configured for Multi-AZ Deployment.
- The database is on v4.0 or later.
- The instance is in **Running** status.

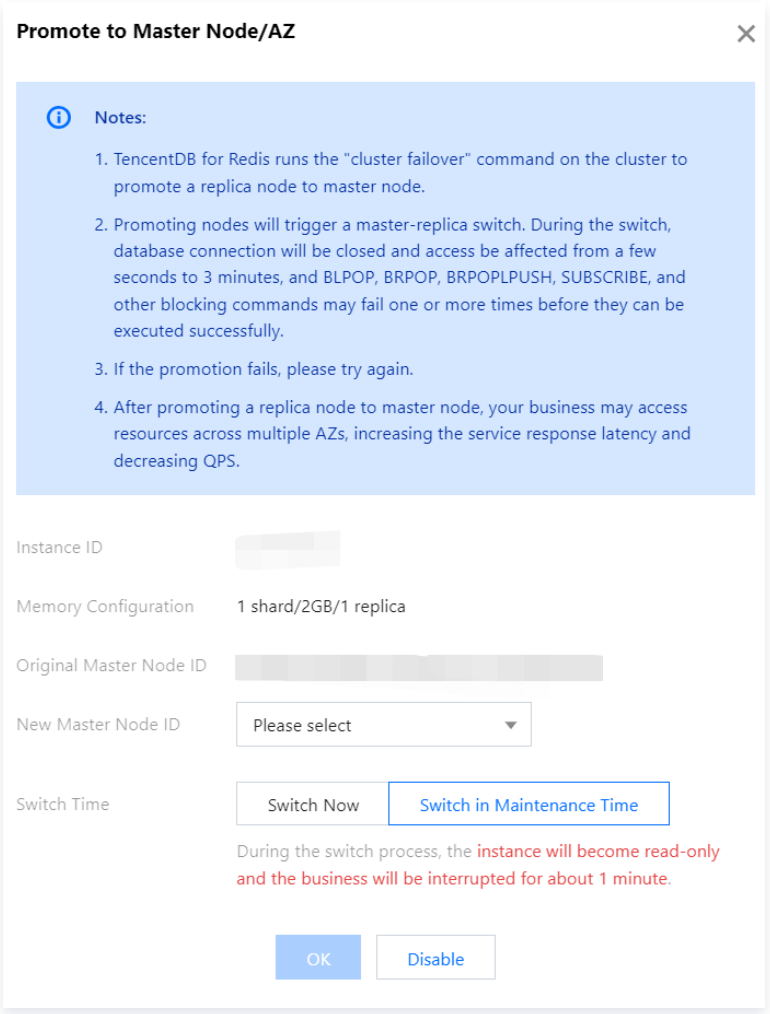
Instructions

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 instance for which you want to promote a replica.
4. Click the **Instance ID** to enter the **Instance Details** page.
5. On the **Instance Details** page, click the **Node Management** tab, and then click **Promote Replica** in the top-right corner of the page.

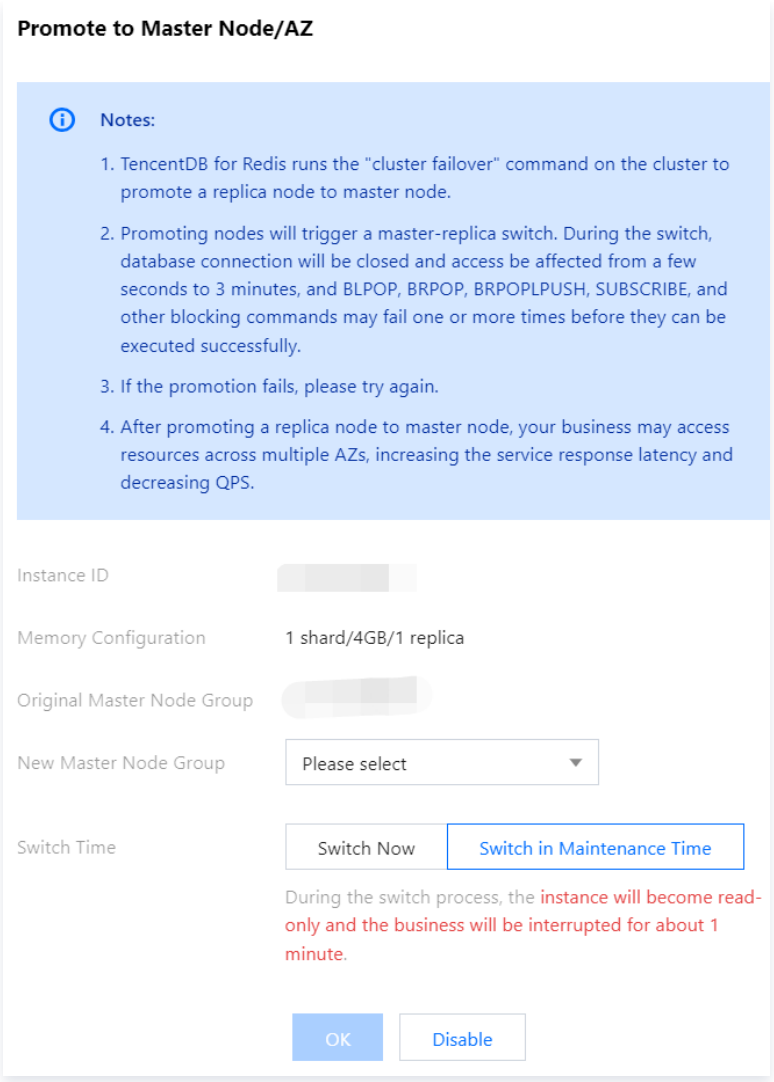


6. In the **Promote to Master Node or Master AZ** window, configure the new master node and set the switch time, as shown in the image below.

- Standard Architecture



- Cluster Architecture



Parameter	Description	Configuration Method
Instance ID	The ID of the current multi-AZ instance awaiting replica promotion.	Verify that the instance is correct.
Memory Configuration	Current instance memory configuration specification.	Verify the instance's specification information.
Original Master Node ID	Display the current master node's ID in the standard architecture.	Confirm the current master node's ID information.
New Master Node ID	In the standard architecture, this parameter is displayed, and you need to configure the replica node that will be switched to the master node.	Please select the replica node to be promoted to the master node from the dropdown list.
Original Master Node Group	In the cluster architecture, this parameter is displayed, showing the current master node group name.	Confirm the current master node group name.
New Master Node Group	In the cluster architecture, this parameter is displayed. Configure the replica node group that needs to be switched to the master node group.	Please select the replica node group to be promoted to the master node group from the dropdown list.

Switch Time	Select the switch time period for promoting the replica.	• **Switch Now**: Initiates the switch task immediately. • **Switch in Maintenance Time**: Initiate the switch task during the maintenance time window. For information and configuration related to the maintenance time window, please see Setting Maintenance Time.

7. After confirming that the configuration is correct, click **OK**.

Related APIs

Related APIs	API Meaning
ChangeReplicaToMaster	This API only supports promoting replica groups in multi-AZ instances and single-AZ replica promotion.

Read-Only Replica

Last updated: 2024-11-01 16:26:21

Scenario

For business scenarios with more reads but fewer writes, TencentDB for Redis® supports designating read requests for frequently accessed data to replica nodes, i.e., read-only replicas. This separates read and write operations by routing them to different database nodes, avoiding read-write conflicts, enhancing the database's concurrent processing capabilities, and improving the overall system performance and stability. For more information, please see Read/Write Separation.

Billing Issues

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

Supports and Limits

- Before enabling the read-only replica feature with read/write separation, confirm that your business allows for potential data inconsistencies (data in the replica node may lag behind the master node).
- Disabling the read-only replica feature for read/write separation may cause existing connections to be briefly interrupted. It is recommended to perform this operation during off-peak business hours.

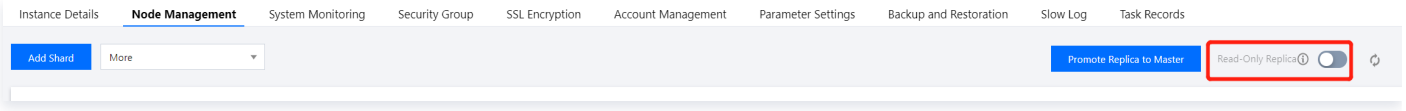
Preparations

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

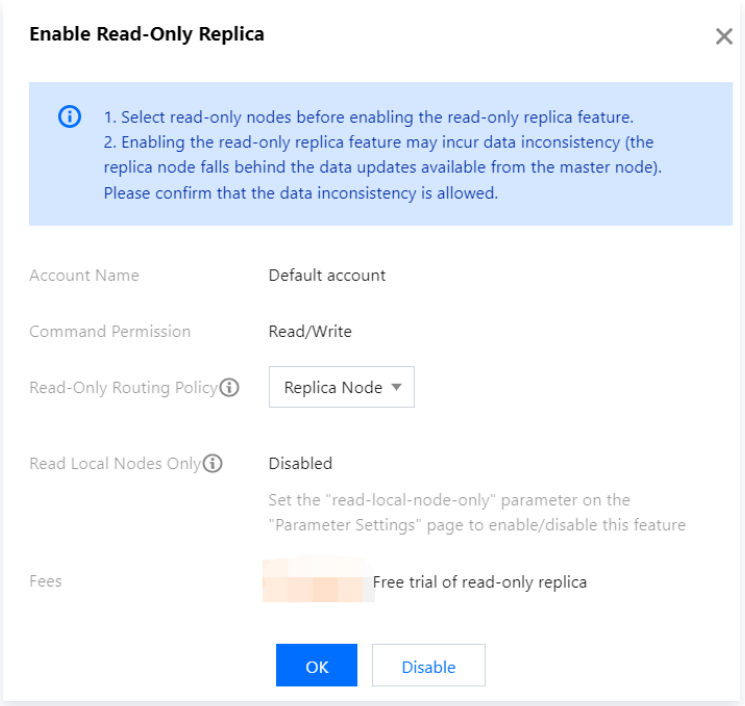
Instructions

Instance with read-only replica enabled

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 on the instance ID to enter the **Instance Details** page, then click on the **Node Management** tab.
5. In the top-right corner of the **Node Management** page, click the icon next to **Read-Only Replica**.



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



Parameter name	Description
Account Name	Fixed to **Default Account**, meaning the system only enables read-only replica 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	By default, the **replica node** is selected, but you can also choose the **master node**, or select both the replica and master nodes. Read requests will be automatically load balanced to the configured read-only nodes by the system.
Read Local Nodes Only	This parameter will be displayed when the instance is deployed across multiple availability zones. The proximity access feature is set to **Disabled** by default. You can enable or disable this feature by configuring the read-local-node-only parameter on the **Parameter Settings** page in the console.
Cost	This feature is current in free trial.

7. After confirming that the parameter settings are correct, click **OK**.
8. **When the instance status** changes to **In Progress**, wait for it to become **Running**. In the **Instance Details** page, under the **Specification Information** section, you can see that **Read-Only Replica** is **Enabled**, and you can now experience read/write separation.

Disabling Read-Only Replica

> ⚠ **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 on the instance ID to enter the **Instance Details** page, then click on the **Node Management** tab.

5. In the top-right corner of the **Node Management** page, click the icon next to **Read-Only Replica**.
6. In the **Disable Read-Only Replica** pop-up window, understand the impact of disabling the read-only replica, confirm the action, and click **OK**.
7. **Instance status** changes to **In progress**. Wait for it to become **Running**. In the **Instance Details** page, under the **Specification Information** section, you can see that **Read-only Replica** is set to **Disabled**, indicating that the feature has been successfully closed.

Related APIs

API Name	Feature
EnableReplicaReadonly	Enabling read/write separation
DisableReplicaReadonly	This API is used to disable read/write separation.

Fault simulation

Last updated: 2024-02-19 16:04:49

Scenario

TencentDB for Redis supports the automatic failover of proxy nodes and Redis servers (data storage nodes) to ensure service availability.
TencentDB for Redis offers a fault simulation feature, facilitating fault simulation testing in conjunction with your business. You can experience this feature through the console. The system triggers automatic HA (High Availability) logic by sending the `shutdown` command to all Redis master nodes for fault 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

Both Redis Standard Architecture and Cluster Architecture use the native cluster management mechanism of Redis Cluster, using the Gossip protocol between nodes to determine node status. The timeliness of node failure detection depends on the cluster-node-timeout, which has a default value of 15000ms and is recommended not to be changed. For more information on node failure detection, see Redis Cluster Native Design.

Instructions for Use

- Only instances in **Running** status can perform fault simulation operations.
- Only instances deployed in **Multi-AZ** can perform fault simulation operations. Instances deployed in the same AZ cannot.

Supports and Limits

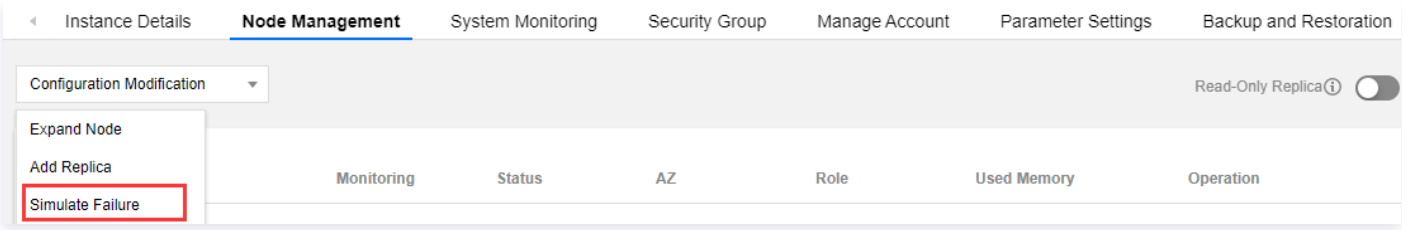
- A failure simulation will make Redis unavailable for less than one minute until the failover is completed. The data written during the simulation may be lost.
- Service unavailability caused by fault simulation will not be included in the Service SLA promised by TencentDB for Redis.

Preparations

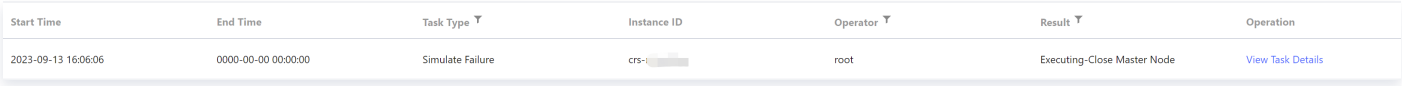
- The instance has been configured for Multi-AZ Deployment.
- The database is on v4.0 or later.
- The instance is in **Running** status.

Instructions

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. On the **Instance Details** page, click the **Node Management** tab, and in the **More Actions** dropdown list, select **Simulate Fault**.



6. In the **Simulate Failure** pop-up window, confirm the instance name and ID, learn about the principles and warning information of fault simulation, and click **OK**. The instance status will change to **In Progress**.
7. On the left sidebar, click **Task Management** and wait for the task to complete. When the instance status returns to **Running**, it indicates that the simulation was successful.



Related APIs

Related APIs	API Description
KillMasterGroup	Performs failure simulation
SwitchProxy	Proxy failure simulation API

Multi-AZ Deployment Management
Configuring multi-AZ deployment

Last updated: 2024-11-01 16:26:56

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

Scenario

- 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.

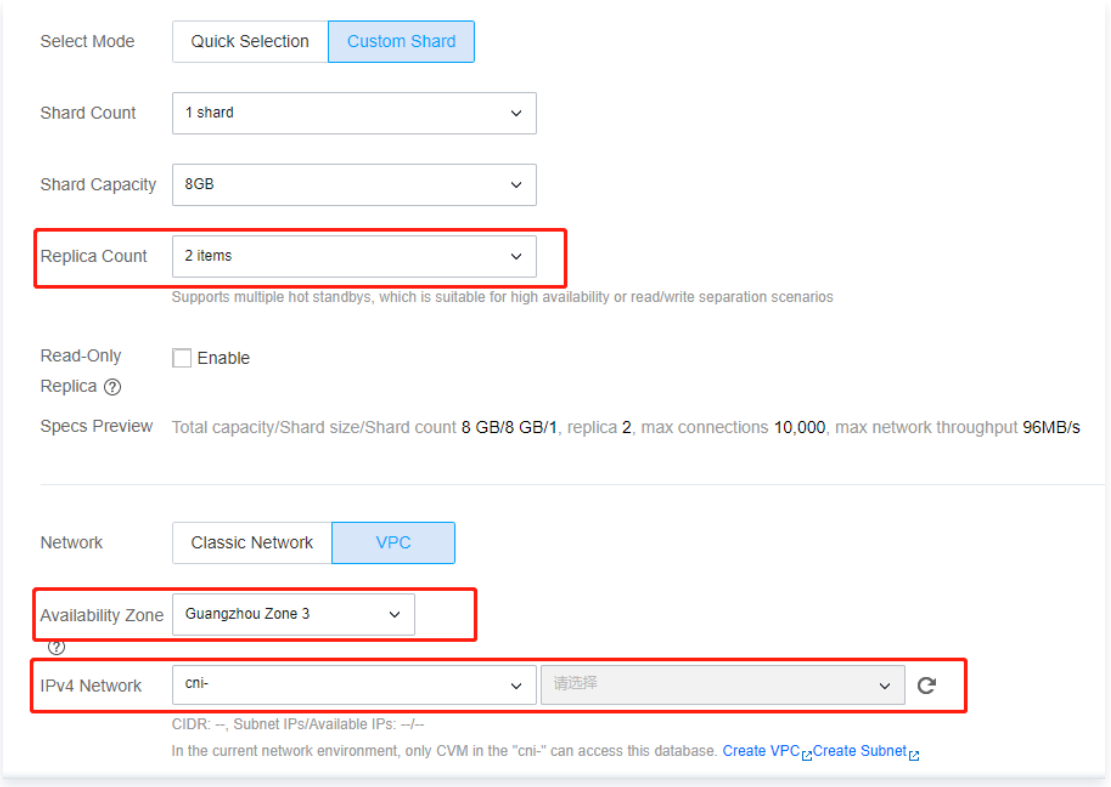
> **Note:**
> Currently, multi-AZ deployment is supported only for TencentDB for Redis® 4.0, 5.0, and 6.2 standard architecture and cluster architecture.

Configuring multi-AZ deployment

1. Log in to the TencentDB for Redis® purchase page.
2. Set parameters such as billing mode, region, product edition, compatible version, architecture, and memory capacity. For more information, see Creating TencentDB for Redis® Instance.
3. When setting **replica quantity**, the number of replicas determines the maximum number of AZs, which is equal to the replica quantity plus 1. Please set the parameter based on your actual business requirements.

   > **Note:**
   > Deploy Redis nodes in different AZs. Up to six AZs are supported.

4. When selecting **AZ**, check **Enable Multi-AZ Deployment** and set the AZs for the master and replica nodes.



5. Set information such as network, port, parameter template, instance name, and password.
6. Confirm the price. After 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

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

Upgrading to Multi-AZ Deployment

Last updated: 2024-11-01 16:27:18

Scenario

Multi-Availability Zones (Multi-AZ) are physical regions formed by combining multiple single-Availability Zones (single-AZ) within the same geographical area. Compared to single-AZ clusters, Multi-AZ clusters offer higher disaster recovery capabilities to protect against database instance failures or AZ interruptions, mitigating data center-level faults. Upgrading existing single-AZ clusters to Multi-AZ clusters can be automatically accomplished through online data migration, without any impact on your business operations.

Billing description

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.

Usage Limits

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

Instructions for Use

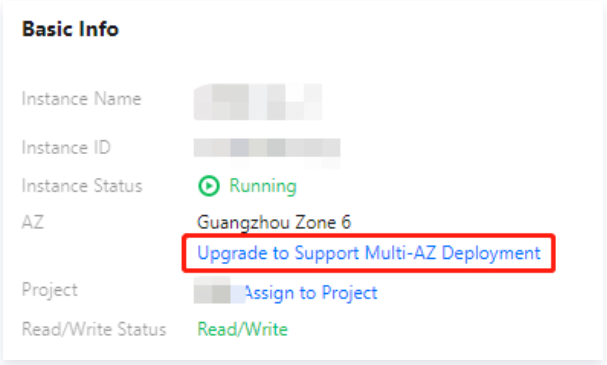
- Without the need for the Read Local Nodes Only feature, upgrading to Multi-AZ only involves the migration of management metadata and has no impact on services. In most cases, the upgrade can be completed within 3 minutes.
- To support the Read Local Nodes Only feature, you need to upgrade the Proxy version and Redis kernel minor version, which involves data migration and may take several hours. Within the last 3 minutes of the upgrade, there may be one or multiple momentary disconnections. Please ensure your business has an automatic reconnection mechanism in place.

Preparations

- The cluster region has at least two AZs.
- The database is on v4.0 or later.
- The instance is in **Running** status.

Instructions

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, click the **Instance ID** in blue font to enter the **Instance Details** page.
5. In the **Basic Info** section of the **Instance Details** page, click **Upgrade to Support Multi-AZ Deployment** after **AZ**.



6. In the **Upgrade to Support Multi-AZ Deployment** dialog box, learn about the impact of upgrading the AZ, confirm whether the Proximity Access feature is supported, and click **OK**.
7. In the **Basic Info** section, wait for the **Instance Status** to change to **Upgrading to Support Multi-AZ Deployment** and then to **Running** to complete the operation.

> **Note:**
> After upgrading to a multi-AZ instance, you need to manually adjust the replica's AZ to deploy the primary and replica nodes in different AZs. For details, see Adding Replicas to Multi-AZ Deployed Instance.

Related APIs

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

Adding Replicas to Multi-AZ Deployed Instance

Last updated: 2024-11-01 16:27:49

Scenario

Multi-AZ instances are designed with a default architecture of one replica. We recommend adopting a deployment model with one master and two replicas, where the primary AZ (Available Zone) contains one master and one replica, and the replica AZ contains one replica. This deployment model maximizes business availability and significantly reduces latency caused by host failures. If the master node fails, a replica in the primary AZ can be prioritized for promotion, ensuring that access latency in the primary AZ is not affected by the master node switching to the backup AZ. For more information, see Multi-AZ Deployment. Set the number of replicas based on your business requirements.

Billing Description

- Monthly Subscription: When adding a replica, the system calculates the price difference between instance specifications and deducts it from your cloud account. Subsequently, you will be charged according to the new specifications.
- Pay-as-you-go: The system bills the instance based on the new configuration specifications at the next hour under the first tier, and fees will be settled at each hour (Beijing Time). For specific billing prices, see Product Pricing.

Instructions for Use

- 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.

Preparations

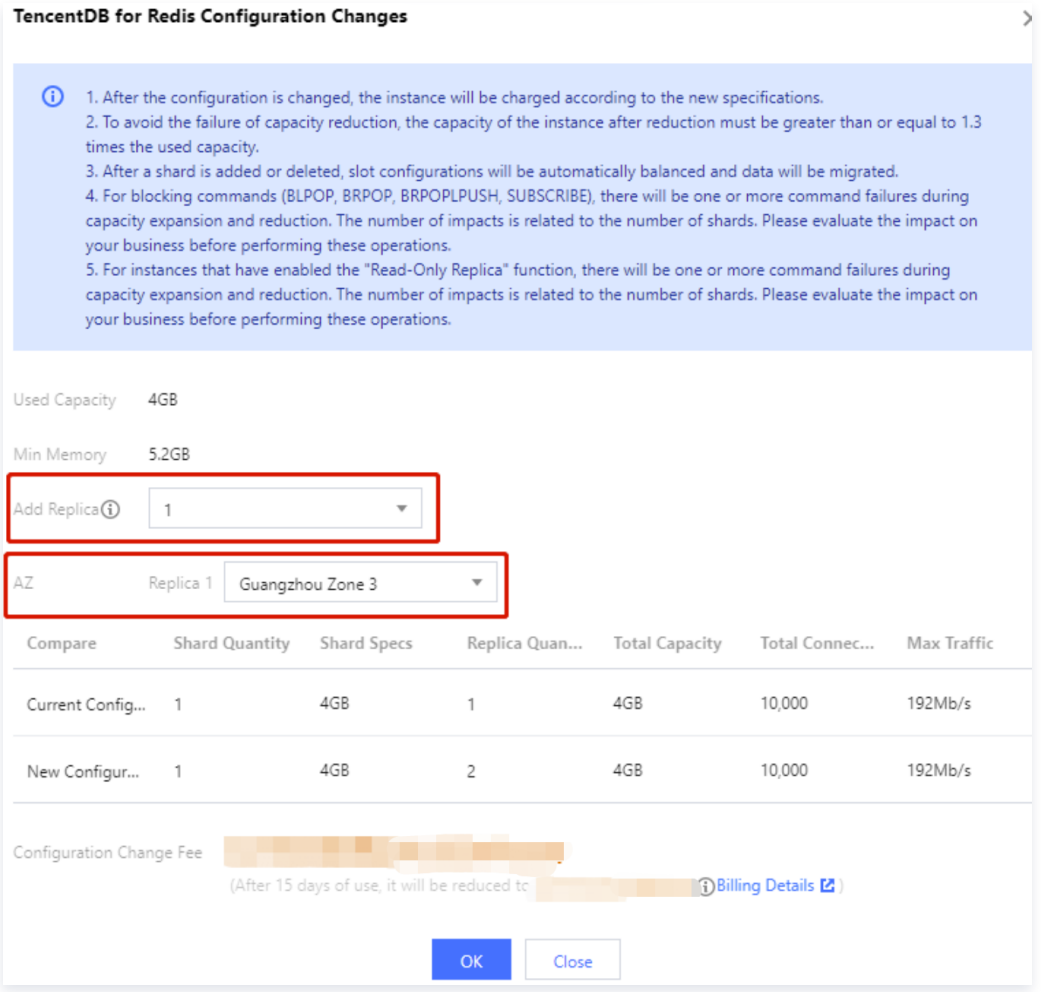
- The instance has been configured for Multi-AZ Deployment.
- The database is on v4.0 or later.
- The instance is in **Running** status.

Instructions

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** dialog box, select the desired number of additional replicas from the drop-down list next to **Add Replica Quantity**, and assign an available zone to the new replica from the drop-down list next to **Available Zone**. For detailed parameter explanations, see the table below.

Parameter name	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.
Availability Zones	Assign a specific AZ to the newly added replica. For configuration guidance, see Multi-AZ Deployment.

Compare	Comparison of the specification before and after adding replicas. Check whether the new configuration meets your expectations.
Configuration Change Fees	Please learn about the billing information after the configuration change. For monthly subscriptions, the fees for the new configuration will be displayed on a monthly basis, while for pay-as-you-go, the hourly unit price for the new configuration will be shown.



6. Once the configuration is complete, click OK. The **Instance Status** will change to **In Progress**. Wait for the status to become **Running**, indicating that the operation is complete.

Related APIs

API Name	Description
UpgradeInstance	Upgrade the instance specifications, including: shard size, shard quantity, and replicas.

Accessing Multi-AZ Deployed Instances

Last updated: 2024-11-01 16:28:09

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

Private IPv4 addresses

- 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 and 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, and all clients in the entire region can access the instance by visiting the address.
- 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

- Access a TencentDB for Redis® instance using an SDK in various programming languages. For details, see Connecting to a TencentDB for Redis® Instance.
- To reduce the access latency of a multi-AZ deployed instance, TencentDB for Redis® offers the ability to access services with reduced latency by utilizing local access. For details, see Local Access.

Backup and Restoration Overview

Last updated: 2024-11-01 16:29:49

Overview

No system is immune to failures, and some failures may lead to catastrophic damage to the database. TencentDB for Redis® backend service periodically backs up instance data on a scheduled basis, supports immediate backups at any time, and persistently stores the generated RDB backup files in Tencent Cloud COS service. TencentDB for Redis® backend service supports rapid cloning of a brand-new instance based on the source instance if error occurs to database data, ensuring database reliability.

Automatic backup

By default, the service backs up data between 02:00 AM and 08:00 AM every day. 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 view the daily backup data in **Backup and Restoration** in the TencentDB for Redis® console. For details, see Backing up Data.

Manual backup

In addition to regular automatic backup performed by the system backend, manual backup is also supported. You can manually back up your instance in the TencentDB for Redis® console whenever you want. For details, see Backing up Data.

Backup File

The backup list displays all the backup files of the instance and their information. TencentDB for Redis® provides two different addresses for downloading backup data from public network and private network, separately. You can choose the download method based on your requirements. For details, see Configuring Backup File Download Methods.

- 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 backup data from being dragged, TencentDB for Redis® allows you to download backup files from specified servers on the allowed private network. You must download backup files from the region where the Redis instance resides but cannot download backup files across regions.

Restore data

Only TencentDB for Redis® 2.8 supports instance restoration based on backup files of the source instance, and the restoration process affects services. For details, see Restoring Data.

Clone Instance

TencentDB for Redis® Memory Edition (excluding v2.8) supports instance clone, that is, creating a brand-new instance based on a backup file. The data of the new instance is the same as that in the backup file. This allows you to analyze history data. You can also roll back an instance by swapping the IPs of the new instance and the original instance. For details, Clone Instance.

Data Backup

Last updated: 2024-11-01 16:30:08

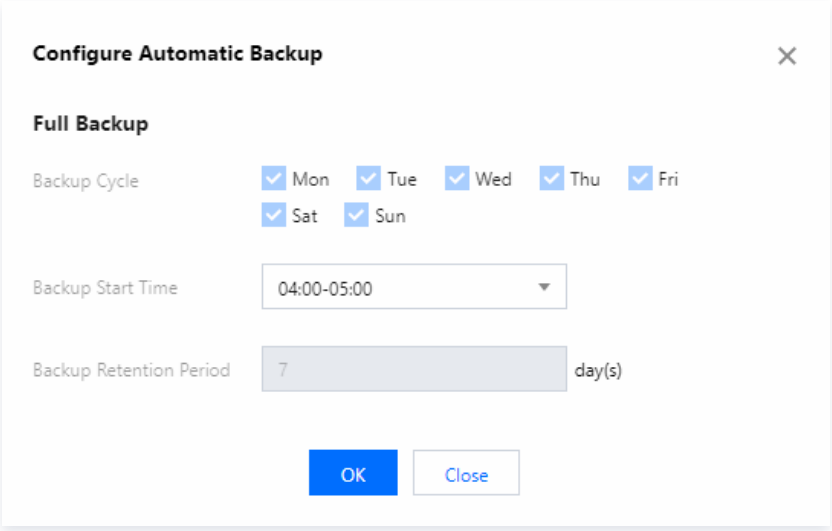
Scenario

TencentDB for Redis® backend service regularly backs up instance data. By default, a full data backup is performed every day between 02:00 AM and 08:00 AM, generating an RDB format database backup file, which is persistently stored in Tencent Cloud COS service. In addition to the regular system backup, the TencentDB for Redis® console also supports manual backup, allowing full data backup at any time outside the automatic backup schedule, catering to various backup requirements for 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 an instance ID to enter the **Instance Details** page.
3. On the **Instance Details** page, choose 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**: Please select a time period for the backup to begin from the drop-down list.
   - **Backup Retention Period**: It is 7 days by default. To change it, please submit a ticket.



6. The backup task starts within the specified time period every day. After the backup is completed, you can view the backup files using the following two methods.
   - 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 an 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 dialog box, please enter the backup task's note information in the input box following **Note**.
5. Click **OK**.

Note:
- The backup task will start after around one minute.
- During the backup process, select **Task Management** in the left sidebar to view the progress and details of the backup task.
- After the backup is completed, you can select **Database Backup** in the left sidebar, and view the backup files with **Backup Mode** being **Manual Backup** on the **Backup List** tab. You can also view the backup files with **Backup Mode** being **Manual Backup** in the **Backup List** on the **Backup and Restoration** tab of the **Instance Details** page.

Related APIs

Related APIs	API Function
ManualBackupInstance	The API is used to backup instance data.
DescribeInstanceBackups	Query the instance backup list
ModifyAutoBackupConfig	This API is used to configure automatic backup.
DescribeAutoBackupConfig	Gets automatic backup configuration

Cloning Instance

Last updated: 2024-11-01 16:30:33

Scenario

TencentDB for Redis® Memory Edition master and replica instances support cloning a complete new instance based on the current backup file within the same AZ or across AZs. The data of the instance is consistent with the backup file, allowing you to analyze history data. Additionally, you can achieve rollback by swapping the IPs of the cloned new instance and the original instance through IP modification.

Release notes

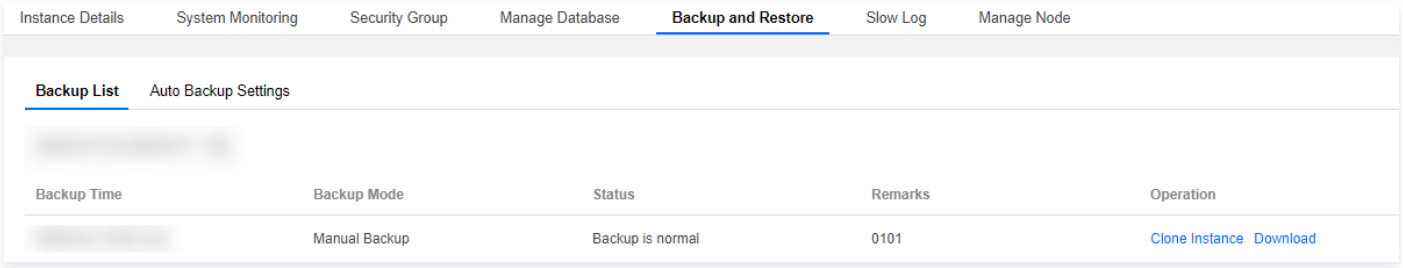
Currently, all Redis versions except 2.8 and 6.2 support instance cloning.

Preparations

- 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.

Instructions

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 an **Instance ID** to enter the **Instance Details** page.
4. On the **Instance Details** page, choose the **Backup and Restoration** tab.
5. In **Backup List**, select the time point to roll back the clone instance based on the backup time. Then, click **Clone Instance** in the **Operation** column.



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

Parameter name	Required	Parameter Explanation
Billing Mode	Supported	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	Supported	The region of the clone instance is the same as that of the source instance and cannot be modified.
Product Version	Supported	The edition and architecture of the clone instance are the same as those of the source instance and cannot be modified.
Clone Backup	Supported	• 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.

Specification Preview	Supported	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 type	Supported	Currently, only **VPC** is supported.
Availability Zone	Supported	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 can be 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	Supported	Select the specific Virtual Private Cloud (VPC) and subnet. We recommend choosing the same VPC in the same region as the cloud server. • 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. • After purchasing an instance, you can switch to a private network as detailed in Switching Networks. You can also click **Create VPC** and **Create Subnet** to establish the desired network environment. For more information, refer to Creating a VPC.
Port	Supported	Customize the port number, which is 6379 by default and ranges from 1024 to 65535. You can modify it after the purchase.
Parameter Template	Supported	Select a parameter template for the clone instance. • The system will automatically adapt the corresponding default template based on the selected compatible version and architecture, allowing you to configure parameters in batches for the instance. For more information, see the default template in Managing Parameter Templates. • To create your own template, click **Create Parameter Template**. For detailed instructions, see Managing Parameter Templates.
Project	Supported	Assign the clone instance to a project in the drop-down list for easy management. • By default, the specified project is **Default Project**. • Click **Create Project** to enter the **Project Management** page and customize the project name to manage your Tencent Cloud resources.
Tags	Not required	Assign tags to the clone instance for organized management. Click **Add** to select tag keys and values.
Security Group	Supported	Set security group rules for the clone instance to manage the inbound traffic to the database. You can either select an existing security group from the **Choose Existing Security Group** dropdown menu, or click on **Custom Security Group** to establish new inbound rules for the security group. For detailed information, please refer to Configuring Security Group.
Instance name	Supported	Set the name for the clone instance, which can contain up to 60 letters, digits, hyphens, and underscores.
Custom Password	Supported	Choose an authentication method, either **Password Authentication** or **No Password Authentication**. The default is **Password Authentication**.
Password	Not required	**Set Password**: When choosing **Password Authentication**, you need to set an access password for the cloned instance. The password complexity

		requirements are as follows: • 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	Not required	Enter the access password for the clone instance again.
Quantity	Supported	For monthly subscription plans, you can purchase up to 100 instances at a time. For pay-as-you-go, you can purchase up to 30 instances at a time, with a purchase range of 1-100 instances per region.
Validity Period	Not required	If you select the **Monthly Subscription** billing mode, you need to choose the duration of the instance purchase. The longer the duration, the greater the discount, allowing you to select based on your actual business needs.
Auto-renewal	Not required	If you select the **Monthly Subscription** billing mode, you can choose whether to enable the **Auto-Renewal** feature. If the feature is enabled, when the instance expires, the system automatically deducts fees from your Tencent Cloud account to renew the subscription.
Terms of Service	Supported	• Click **TencentDB Terms of Service** to fully understand the content, service fees, usage rules, intellectual property rights, and other related terms for using TencentDB services. • Click **Service Level Agreement** to learn about the agreement that needs to be followed when using TencentDB for Redis®. • Select **I have read and agreed to the TencentDB Service Level Agreement and Service Level Agreement.**

7. Click **Buy Now**. After the purchase is completed, you will be redirected to the instance list. After the instance status becomes **Running**, 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 data rollback.

> **Note:**
> After the instance is cloned, the source instance can be retained or terminated based on your needs.

Restoring Data

Last updated: 2024-11-01 16:30:47

Scenario

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 instances by recovering data in the source instance according to a backup file. TencentDB for Redis® Memory Edition (excluding v2.8) creates a complete new instance based on the current instance, and achieves data rollback by swapping the IPs of the cloned new instance and the original instance.
- 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.

Preparations

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

Instructions

1. Log in to the TencentDB for Redis® console, click an instance ID in the instance list to enter the instance management page.
2. On the instance management page, choose the **Backup and Restoration** tab, select the backup file you want to restore, and click **Restore Instance**.
3. In the pop-up dialog box, confirm that everything is correct and click **OK**.

   Note
   If the instance has a password, you need to enter the user-defined instance password, not the **Instance ID: Instance Password** connection password used when accessing the instance.

4. Return to the instance list. The instance status will display as **Restoring backup by backup ID**. After the status changes to **Running**, the instance can be used normally.

Data Migration
Migration Scheme Overview

Last updated: 2024-11-01 16:33:30

Utilizing Data Transmission Service (DTS) and the migration tools in the redis-port collection, including redis-sync, redis-dump, and redis-restore, TencentDB for Redis® offers a variety of data migration solutions to meet the diverse requirements of Redis data migration across various business scenarios.

migration tools

- DTS: Tencent Cloud Data Transmission Service (DTS) for Redis helps users effortlessly migrate databases to the cloud without interrupting their business operations. It supports full + incremental data migration, allowing both historical data in the source database prior to migration and newly written data during the migration process to be migrated together.
- redis-sync: Facilitates data migration between Redis instances by simulating redis-sync as a replication node that syncs data from the source instance and translates the replicated data into write commands to update the target instance.
- redis-dump and redis-restore: redis-dump backs up Redis data into RDB files in an offline environment, and then redis-restore imports the RDB files into the specified Redis instance.

Migration Solutions

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 Tencent Cloud VPCs through Direct Connect. For detailed directions, see Getting Started. • VPN Access: The source database can be interconnected with Tencent Cloud VPCs through VPN connections. For detailed directions, see Quick Start Guide for VPN Access. • Intranet (only suitable for Tencent's internal businesses): You need to submit a ticket to enable Tencent Cloud.	• 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, 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 to enable	• The migration is online, with full + incremental data sync supported. • The source Redis database must allow the `SYNC` or `PSYNC` command.

			Tencent Cloud.	• Supported source Redis versions are 2.8, 3.0, and 4.0, while supported target Redis versions include 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 Private Network.	• 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 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 self-built Redis	Migration from TencentDB for Redis® to self-built Redis (migration off the cloud and multi-cloud sync)	DTS (Recommended)	The following five access types are supported: • 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 Tencent Cloud VPCs through VPN Access. • CCN: The source database can be interconnected with VPCs through CCN.	• The migration is online, with full + incremental data sync supported. • You need to submit a ticket to initiate the service with Tencent Cloud. • 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 database.	• 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 database. • VPC: The source database can be accessed through Private Network.	• 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

				instructions, see Upgrading Versions Using DTS.
	Migration across Tencent Cloud accounts	DTS (Recommended)	Database: the source database is a TencentDB database.	• 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 to Tencent Cloud to enable migration across accounts.
	Migration from Tencent Cloud standard architecture to cluster architecture	DTS (Recommended)	Database: the source database is a TencentDB database.	• 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 in advance to avoid service execution errors after the upgrade.
	Migration from legacy TencentDB for Redis® Cluster Edition instance (purchased before January 1, 2018)	redis-restore	Offline (VPC)	Only full data sync is supported. Business 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 Tencent Cloud VPCs through VPN Access.	• 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 VPCs through Direct Connect. • VPN Access: The source database can be interconnected with Tencent Cloud VPCs through VPN Access. • VPC: The source database can be accessed through Private Network. • Intranet: You need to submit a ticket to enable Tencent Cloud.	• The migration is online, with full + incremental data sync supported. • All SSDB kernel versions are supported.
	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 Tencent Cloud VPCs through VPN Access. • VPC: The source database can be accessed through Private Network. • Intranet: You need to submit a ticket to enable Tencent Cloud.	• 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. Internal business cloud development within the

			Tencent Cloud VPCs through VPN Access. • VPC: The source database can be accessed through Private Network. • Intranet: You need to submit a ticket to enable Tencent Cloud.	organization requires activation.
	Migration from Tencent istore to TencentDB for Redis®	DTS	Intranet: You need to submit a ticket to enable Tencent Cloud.	• 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
Version Upgrade with DTS

Last updated: 2024-02-23 11:21:40

Scenario

TencentDB for Redis 4.0, 5.0, 6.2 Memory Edition (Standard Architecture) and 4.0, 5.0, 6.2 Memory Edition (Cluster Architecture) offer more flexible configuration options, higher performance, and enhanced features. Therefore, for better cloud database service experience, it is recommended to upgrade your Redis to 4.0, 5.0, or 6.2.
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	Note
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)	6.2 Memory Edition (Standard Architecture)	6.2 Memory Edition (Cluster Architecture)
2.8 Memory Edition (Standard Architecture)	✓	✓	✓	✓	✓	✓
4.0 Memory Edition (Standard Architecture)	–	✓	✓	✓	✓	✓
4.0 Memory Edition (cluster architecture)	–	–	✓	✓	✓	✓
5.0 Memory Edition (Standard Architecture)	–	–	–	✓	✓	✓
5.0 Memory Edition (Cluster Architecture)	–	–	–	–	✓	✓
6.2 Memory Edition (Standard Architecture)	–	–	–	–	–	✓

Preparations

- The source TencentDB for Redis instance should be running properly.
- You have already purchased TencentDB for Redis 4.0, 5.0, or 6.2 Memory Edition (Standard Architecture) or Redis 4.0, 5.0, or 6.2 Memory Edition (Cluster Architecture) instances.

> **Note:**
> For instances with data volume less than 12 GB, subsequent data growth not exceeding 60 GB, and QPS not exceeding 40,000, or if transaction support is required, it is recommended to choose Redis 4.0, 5.0, or 6.2 Memory Edition (Standard Architecture). Otherwise, consider selecting Redis 4.0, 5.0, or 6.2 Memory Edition (Cluster Architecture).

Instructions

1. To use DTS to migrate data from a TencentDB for Redis source instance to a Redis 4.0, 5.0, or 6.2 Memory Edition (Standard Architecture) or Redis 4.0, 5.0, or 6.2 Memory Edition (Cluster Architecture) instance, please refer to Migration with DTS.
2. Once data synchronization is complete and the business side has verified the data accuracy, you can choose a suitable time to disconnect the source Redis instance based on business QPS and other indicators, and switch the connection to the target Redis instance. There are two methods for switching:
   - **Switching between consoles:**
     - Document the old IP address of the Redis source instance and change it to another IP address.
     - To complete the business switch, modify the network information of the target Redis instance to be in the same VPC subnet as the source Redis instance, and change the target instance's IP address to the old IP address of the source instance. For detailed steps on modifying network information and IP address, see Configure Network.
   - **Switching IP address in code:** Update the IP of the source Redis instance in the code to the IP of the target Redis instance.

Migration with redis-port

Last updated: 2024-11-01 16:34:06

Tools

redis-port (Linux 64-bit) is a collection of open-source tools primarily used for synchronization, data import, and data export between Redis nodes, supporting cross-version data migration for Redis. The collection includes 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 Version

- 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

Migration Principle

- 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.

> **Note:**
> - 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 reasons such as network failure, among others, you need to empty the destination 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. You are recommended to 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.

Use case

```
./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"@10.0.5.8:6379
2019/02/21 09:56:00 sync.go:76: [INFO] sync: master = "127.0.0.1:6379", target =
"xxx2018@10.0.5.8:6379"
2019/02/21 09:56:01 sync.go:103: [INFO] +
2019/02/21 09:56:01 sync.go:109: [INFO] sync: runid = "f63e2ad58e2fcc15c8cc122f15778389a012c1a4",
offset = 18576271
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.skip/rdb.forward,rdb.skip)
2019/02/21 09:56:02 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=
(1703936/71754,0/0,0) ~ (1.62mb/-,-/-,-) ~ speed=(1.62mb/71754,0/0,0)
2019/02/21 09:56:03 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=
(3407872/153850,0/0,0) ~ (3.25mb/-,-/-,-) ~ speed=(1.62mb/82096,0/0,0)
2019/02/21 09:57:54 sync.go:250: [INFO] sync: rdb = 9063349 - [100.00%] (r/f,s/f,s)=
(80487526/411969,0/1587212,0) ~  (76.76mb/-,-/-,-) ~ speed=(0/0,0/0,0)
```

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. You are recommended to 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.

Use case

```
./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.

> **Note**
>
> 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. You are recommended to 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.

Use case

```
./redis-dump  127.0.0.1:6379 -o dump.rdb
```

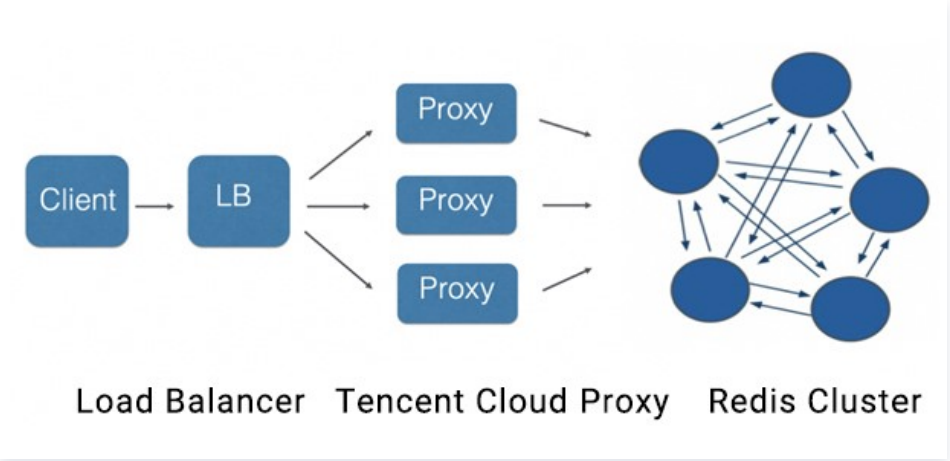
Check on Migration from Standard Architecture to Cluster Architecture

Last updated: 2024-11-01 16:34:24

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:

Multi-key operations

TencentDB for Redis® Memory Edition (Cluster Architecture) uses the hash algorithm to distribute keys to 16,384 slots. For more information on the principle, 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 multi-key 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 EVAL and EVALSHA commands must be passed a Key parameter; otherwise, the commands will not execute.
- 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

TencentDB for Redis® Memory Edition (Cluster Architecture) encapsulates VIP to provide a Standard Edition user experience in cluster mode, which greatly facilitates business usage. However, it is not transparent enough for operations and maintenance. Custom commands are introduced to address this issue, supporting access to each node in the cluster. A new parameter "Node ID" is added to the right of the parameter list of the original command: COMMAND arg1 arg2 ... [Node ID]. The Node ID can be obtained through the 'cluster nodes' command or from the console.

```
10.1.1.1:2000> cluster nodes25b21f1836026bd49c52b2d10e09fbf8c6aa1fdc 10.0.0.15:6379@11896 slave
36034e645951464098f40d339386e9d51a9d7e77 0 1531471918205 1
connectedda6041781b5d7fe21404811d430cdffea2bf84de 10.0.0.15:6379@11170 master - 0 1531471916000 2
connected 10923-1638336034e645951464098f40d339386e9d51a9d7e77 10.0.0.15:6379@11541 myself,master -
0 1531471915000 1 connected 0-546053f552fd8e43112ae68b10dada69d3af77c33649 10.0.0.15:6379@11681
slave da6041781b5d7fe21404811d430cdffea2bf84de 0 1531471917204 3
connected18090a0e57cf359f9f8c8c516aa62a811c0f0f0a 10.0.0.15:6379@11428 slave
ef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 0 1531471917000 2
connectedef3cf5e20e1a7cf5f9cc259ed488c82c4aa17171 10.0.0.15:6379@11324 master - 0 1531471916204 0
connected 5461-10922

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.
- Compatibility with more commands. Codis does not support blocking commands such as `BLPOP` and `SUBSCRIBE`.

- If a large key occurs in data migration with Codis, services may become unavailable. In contrast, TencentDB for Redis® supports lossless expansion with no fear for large 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 tool and use it to execute the info commandstats command. This tool helps analyze whether cross-slot related commands have been executed in the Standard Edition, assisting in determining potential compatibility issues.

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

> **Note:**
> `host, port, password` : Enter the information of Redis Standard Edition.

2. Refer to the Compatibility Description above, and evaluate each item on the business side to determine if it can pass.

Dynamic evaluation

Download the cluster_migrate_online_check dynamic verification tool to simulate the execution of the psync command by the client, syncing incremental data from Standard Edition to TencentDB for Redis® Memory Edition (Cluster Architecture) in real time. This real-time sync can help you determine whether there are compatibility issues with write commands. However, this tool cannot cover the compatibility testing 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 by default
export pipeline = 2000  // Number of concurrent pipelines, which is 1,000 by default
```

> **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. Refer to the Compatibility Description above, and evaluate each item on the business side to determine if it can pass.

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. We recommend you 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: 2024-11-01 16:34:34

Scenario

TencentDB for Redis® Legacy Cluster Edition (purchased before January 1, 2018) has a lower version and certain stability risks. It is recommended to migrate to Redis 4.0 Memory Edition (Standard Architecture) or 4.0 Memory Edition (Cluster Architecture).
Redis 4.0 offers more flexible configuration options, higher performance, and more comprehensive features. We will assist you in upgrading your Redis Legacy Cluster Edition instance to Redis 4.0 Memory Edition (Standard Architecture) or 4.0 Memory Edition (Cluster Architecture) as soon as possible. For details, see Redis Memory Edition (Standard Architecture) and Redis Memory Edition (Cluster Architecture). If you encounter any issues during the migration, please provide feedback through Online Support.

> **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.

Preparations

- 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.
- Ensure that the data import tool, redis-port, is installed. For usage instructions and download links, see Tool Introduction.

Instructions

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 the data backup is completed, you can view the previously backed-up files in the backup list. Click **Export** to generate an RDB file. Once the RDB file is generated, a corresponding download link will be created. Click the download link to copy the internal network address and download the backup file from an internal network CVM. Cross-AZ downloads are not supported.
4. Initialize the password for the newly purchased Redis 4.0 Memory Edition (Standard Architecture) or 4.0 Memory Edition (Cluster Architecture), and use the redis-port tool to import the downloaded RDB file into the new instance. Here's an example:

```
./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. To migrate your application to the new instance, simply replace the IP of the legacy instance in the code with the IP of the new instance.

Pika-to-Redis Data Migration Scheme

Last updated: 2024-11-01 16:34:42

Migration Principles

Migrates data from Pika to Redis online, supporting both full and incremental synchronization. Utilizes the pika-migrate tool, virtualizing it as a Pika replica, then obtains data from the master and forwards it to Redis, while also supporting incremental synchronization to achieve 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

Applicable to Pika 3.2.0 and later versions, in standalone mode with a single database. If the Pika version is lower than 3.2.0, the kernel must be upgraded to 3.2.0. For more information, see Upgrading Pika Kernel Version to 3.2.0.

Supports and Limits

- 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 long time for pika-port to write the full data to Redis, leading to the clearance of the original binlog offsets in the master. It is essential to retain 10,000 binlog files on the Pika master, ensuring that the corresponding binlog files still exist when the tool requests incremental synchronization.
> - 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.

```
93   ####################
94
95   target-redis-host : 127.0.0.1
96   target-redis-port : 6379
97   target-redis-pwd  :
98
99   sync-batch-num    : 100
100  redis-sender-num  : 10
```

   - **target-redis-host**: Specifies the IP address of Redis.
   - **target-redis-port**: Specifies the port number of Redis.
   - **target-redis-pwd**: Specifies the password of the default Redis account.
   - **sync-batch-num**: Specifies that pika-migrate packages and sends sync-batch-num data received from the master to Redis together, enhancing the forwarding efficiency.
   - **redis-sender-num**: Specifies the number of threads for forwarding data packets. The forwarding command distributes data to different threads for sending based on the hash value of the Key, eliminating concerns about data confusion 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 the successful establishment of the master-replica relationship, pika-migrate forwards data to the target Redis simultaneously. Execute the following command to check the synchronization delay. Write a unique key to the master and then check on the Redis side if the key can be immediately retrieved, determining if the data synchronization is complete.

```
info Replication
```

SSDB-to-Redis Data Migration Scheme

Last updated: 2024-11-01 16:34:53

Migration Principles

- The Siphon migration tool, developed using Golang, is disguised as an SSDB Slave for data subscription and synchronization 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, suitable for all SSDB kernel versions.
- If your SSDB involves large keys or keys at the level of hundreds of millions, please submit a ticket to apply for the modified version Siphon-V2 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.

Supports and Limits

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 initiation of full data synchronization.
   - **Copy Stop**: Indicates that the full data synchronization has ended.
3. Wait for new data to be generated and incrementally synced to Redis without exiting the process.

Frequently Asked Questions about DTS Migration

Last updated: 2024-11-01 16:35:34

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. DTS supports migrating data from the same source to multiple targets and from multiple sources to the same target. However, running multiple tasks concurrently may increase the access pressure on the source and target databases, affecting the migration speed. If you need to create multiple tasks for the same source database, after creating the first migration task, you can quickly create similar tasks by clicking **Operation** column **More** > **Create similar task**.

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 monitor the progress of the migration task on the Tencent Cloud DTS console's 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. 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?

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 the migration type is set to either "Structure Migration" or "Full Migration", the task will automatically terminate upon completion, eliminating the need for manual termination by the user.
- If the migration type is selected as **Full + Incremental migration**, after full migration is completed, incremental data sync automatically starts, and the sync process does not stop automatically. You need to click **Complete** to manually stop the incremental data sync.
  - Please manually complete incremental data sync and business switchover at appropriate time.
  - You can see that the migration task is in the incremental sync stage, and there is no latency. Stop writing the source database for several 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 second.

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

Due to the differing fragmented spaces between the source and target databases, the source database may contain data gaps. Consequently, after a full migration, the table storage space in the target database may be smaller than that in the source database. It is recommended that users perform a data consistency check after migration to verify the consistency of the contents in both databases.

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	Destination
Self-built database Redis 2.8, 3.0, 3.2, 4.0, 5.0, and 6.2 (including Tencent Cloud CVM-based databases)	TencentDB for Redis® 2.8, 3.0, 3.2, 4.0, 5.0, and 6.2.
TencentDB for Redis® 2.8, 3.0, 3.2, 4.0, 5.0, and 6.2.	
Databases in other clouds on Redis 2.8, 3.0, 3.2, 4.0, 5.0, and 6.2	

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 from multiple source databases to the same target database, 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.

Account and Password
Password-Free Access

Last updated: 2024-11-01 16:36:29

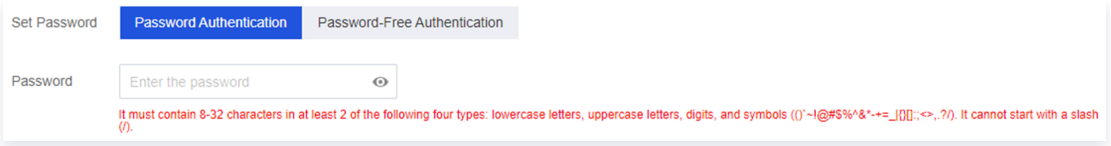
TencentDB for Redis® supports password-enabled and password-free access.

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. On the purchase page, under the **Set Password** option, select **Password-Free Authentication**. After the instance is successfully created, it can be accessed without password authentication.

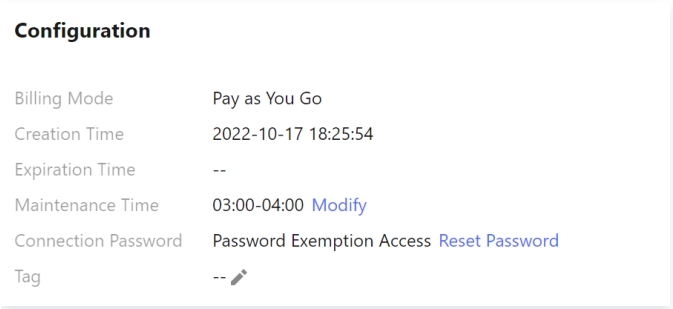


Enable password-free access for existing instances

In the instance list, click an instance ID to enter the instance details page. In the **Connection Password** section of the configuration information, click **Password-Free Access** to enable password-free access.

Viewing Password-Free Access Status

In the instance list, click an instance ID to enter the instance details page, and check whether password-free access is enabled for the instance in the **Connection Password** section of the configuration information.



Disabling Password-Free Access

To disable password-free access, go to the **Connection Password** section of the configuration information and click **Reset Password**.

Related APIs

Related APIs	API Function
ResetPassword	The API is used to enable password-free access. If you do not specify the Password parameter, you can switch to password-free access.

Managing Account

Last updated: 2024-11-01 16:36:39

Scenario

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.

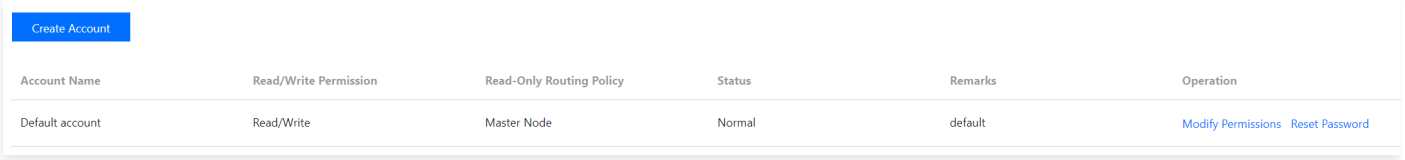
- Account Types
  - Default account: It's an account with only a password. When creating an instance, set the password for accessing the database as instructed in Creating TencentDB for Redis® Instance.
  - Customized account: It's an account with an account name. The authentication method of such an 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 password-free authentication method different from that of Redis Community Edition. Specifically, after password-free 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, read requests cannot be distributed to replica nodes. This feature can be enabled on the **Node Management** page.
  - Instances with accounts that access replica nodes are not allowed to disable the **Read-Only Replica** feature. To disable the **Read-Only Replica** feature, you must first delete the account.

Preparations

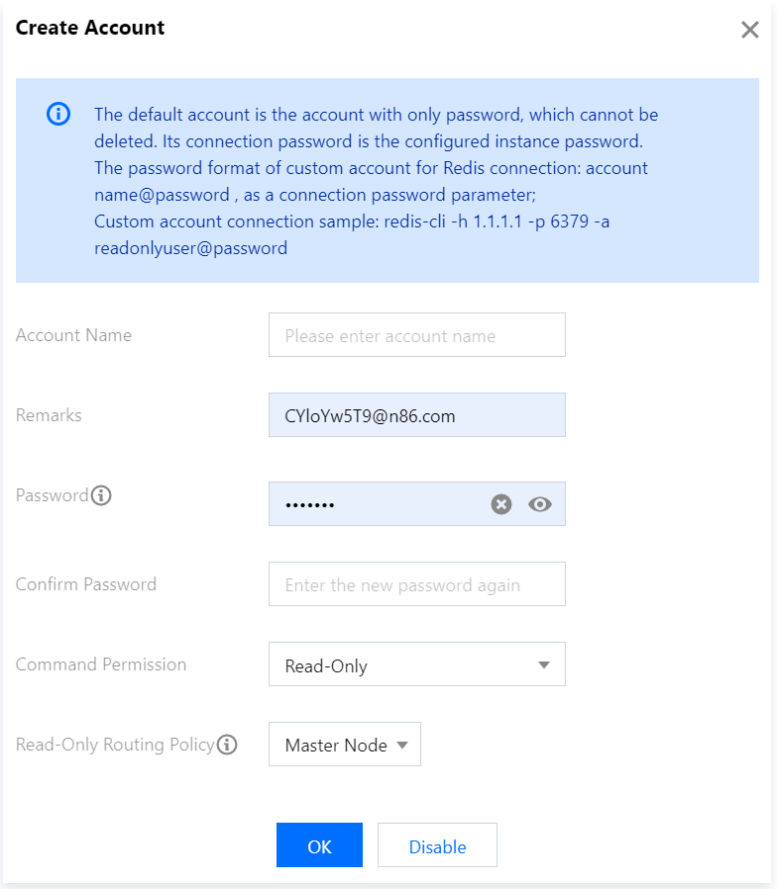
- You have created a TencentDB for Redis® instance, and the instance is running.
- You have planned the accounts to be defined and their passwords, read-write permissions, and read-only routing policies.

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**.



6. In the **Create Account** dialog box, set the custom account name and password information as shown in the following image.



Interface Parameter	Description
Account name	Set the custom account name. • Contain only letters, digits, or symbols (-_). • Contain up to 32 characters.
Password	Set password for the customized account. The password must meet the following requirements: • Contain [8,30] characters. • Contain at least two of the following four types: lowercase letters, uppercase letters, digits, and symbols (()`~!@#$%^&*-+=_|{}[]:;<>,.?/). • It 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 Policy	Distribute read requests from the specified account to the master node or replica node. 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.

7. Click **OK** to complete the creation, and you can use this customized account to access the database. For specific connection methods, see Connecting to Instances.

Deleting a custom account

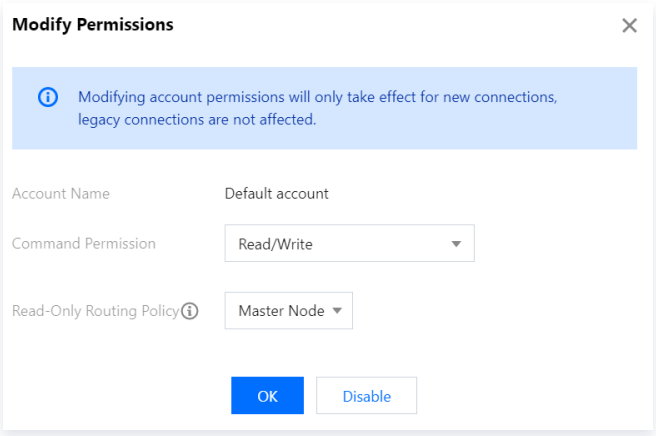
1. Log in to the TencentDB for Redis® console.
2. Above the **Instance List** on the right, select the region.

3. In the instance list, find the target instance.
4. Click the target instance ID to enter the **Instance Details** page.
5. Click the **Account Management** tab, and in the account list, locate the customized account you wish to delete.
6. In the **Operation** column, click **Delete**.
7. In the **Delete Account** dialog box, confirm the accounts 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 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. Click the **Account Management** tab, and in the account list, locate the customized account for which you want to modify permissions.
6. Click **Modify Permissions** in the **Operation** column.
7. In the **Modify Permission** dialog box, select the read-write permission of the account from the drop-down list following **Command Permission**; choose the read-only routing distribution node from the drop-down list after **Read-only Routing Policy**, as shown in the image below.



> **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 nodes for the read-only routing policy. For detailed directions, see Enabling/Disabling Read/Write Separation.

8. Click **OK** to complete the configuration.

Related APIs

Related APIs	API Meaning
CreateInstanceAccount	Creates a custom account for the instance
ModifyInstanceAccount	Modifies a sub-account of the instance

DeleteInstanceAccount	Deletes a custom account of the instance
DescribeInstanceAccount	Views the account information of the instance

Reset Password

Last updated: 2024-11-01 16:36:49

Scenario

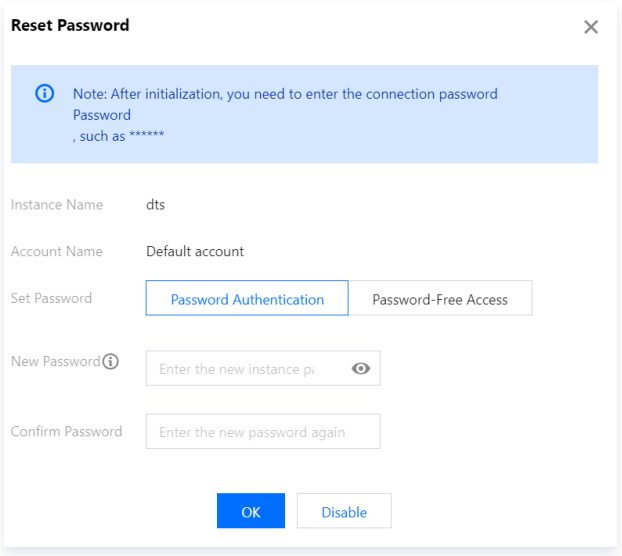
If you forgot or want to change your password, you can reset it directly in the console.

Preparations

- You have created a TencentDB for Redis® instance. For more information, see Creating TencentDB for Redis® Instance.
- The database instance is in **Running** status.

Instructions

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 open the **Reset Password** dialog box using any of the following methods to reset the default account password.
   - On the **Instance Details** page, under the **Configuration Information** section, click **Reset Password** on the right of **Connection Password** to set a new password for the default account.
   - In the account list on the **Account Management** page, find the **Default account** or customized account for which to reset the password, and click **Reset Password** in the **Operation** column.
5. In the **Reset Password** window, enter the **New Password** and **Confirm Password**. The password must meet the following requirements:
   - It must contain 8–32 characters, preferably 12 or more.
   - It 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**. The new password will take effect immediately.

Related APIs

API Name	Feature
ResetPassword	Reset password

Parameter Configuration
Setting Instance Parameters

Last updated: 2024-11-01 16:37:42

This document describes how to configure instance parameters in the TencentDB for Redis® console.

Scenario

You can view and modify certain instance parameters and query parameter modification logs in the TencentDB for Redis® console.

> **Note:**
> To ensure the instance stability, the console allows you to modify certain parameters only as displayed on the parameter configuration page in the console.

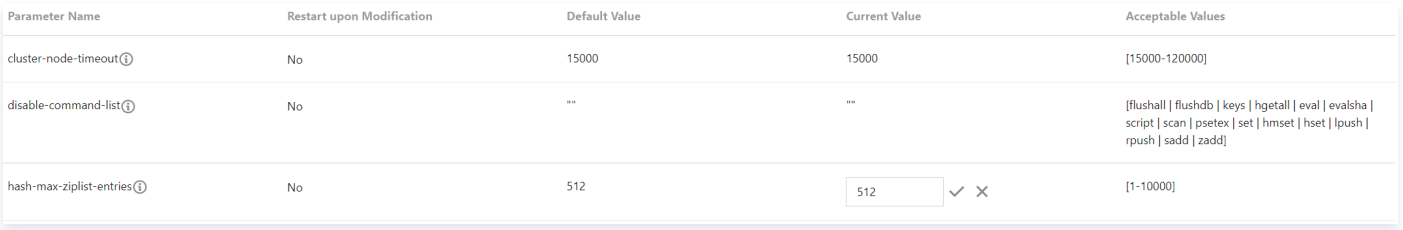
Editing Parameters

Editing one parameter

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, choose 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.

Parameter Name	Restart upon Modification	Default Value	Current Value	Acceptable Values
cluster-node-timeout	No	15000	15000	[15000-120000]
disable-command-list	No	""	""	[flushall | flushdb | keys | hgetall | eval | evalsha | script | scan | psetex | set | hmset | hset | lpush | rpush | sadd | zadd]
hash-max-ziplist-entries	No	512	512	[1-10000]
hash-max-ziplist-value	No	64	64	[1-10000]

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.



Parameter Name	Restart upon Modification	Default Value	Current Value	Acceptable Values
cluster-node-timeout	No	15000	15000	[15000-120000]
disable-command-list	No	""	""	[flushall | flushdb | keys | hgetall | eval | evalsha | script | scan | psetex | set | hmset | hset | lpush | rpush | sadd | zadd]
hash-max-ziplist-entries	No	512	512	[1-10000]

Editing parameters in batches

1. Log in to the TencentDB for Redis® console.
2. Above the **Instance List** on the right, select the region.
3. Click the ID of the target instance in the instance list to enter the **Instance Details** page.
4. Choose 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** to apply the changes.

Importing a parameter template into the current instance

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. Choose 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** on the right, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. Choose 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.

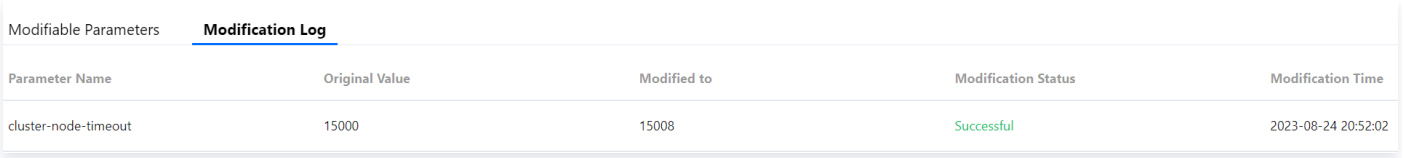
Saving Parameters as a Template

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** on the right, select the region.
3. In the instance list, click the instance ID to enter the **Instance Details** page.
4. Choose 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** on the right, 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

Category	Note	Supported Versions
auto-failback	When deploying instances across multiple availability zones, this parameter determines whether the master node automatically switches back to the master node group (Cluster) after a failback.	Redis 4.0、5.0

disable-command-list	Disable commands that have high time complexity or are highly risky. The disabled commands will not be allowed to run in this instance. To disable multiple commands, separate them with commas, such as `flushdb,keys`.	Redis 2.8、4.0、5.0、6.2
hz	Set the execution frequency.	Redis 2.8、4.0、5.0、6.2
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 and delete keys with a set TTL using the LRU algorithm, which prioritizes the eviction of the least recently used keys among those with an expiration time (TTL). • allkeys-lru: Evict all keys using the LRU algorithm, regardless of whether the data has a timeout attribute set, prioritizing the eviction of the least recently used keys until enough space is freed. • volatile-random: Randomly evict and delete keys with a set TTL expiration time. • allkeys-random: randomly evict keys. • volatile-ttl: For keys with a set TTL, prioritize evicting keys with smaller TTL values among those with an expiration time. • noeviction: Do not evict or delete any keys, reject all write operations, and return error messages when detecting a write operation. • volatile-lfu: Preferentially evict the least frequently used (LFU) keys among those with an expiration time (TTL) set. • allkeys-lfu: Prioritize evicting the least frequently used (LFU) keys. Unlike volatile-lfu, the allkeys-lfu policy evicts all keys, not just those with a set expiration time (TTL). LRU (Least Recently Used) represents the least recently used items; TTL (Time To Live) refers to the configured expiration time; LFU (Least Frequently Used) denotes the least frequently used items. For more information, see Basic Usage Guidelines.	Redis 2.8、4.0、5.0、6.2
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、6.2
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、6.2
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`. • The number of hash entries is smaller than the value of `hash-max-ziplist-entries`.	Redis 2.8、4.0、5.0、6.2
lazyfree-lazy-eviction	This parameter controls whether Redis performs lazy deletion (Lazy Free) and lazy eviction (Lazy Eviction) during memory reclamation. • Lazy deletion: When Redis needs to delete a key-value pair, it does not immediately remove it from memory. Instead, it marks the pair as "to be deleted" and waits for the next access to delete it. This mechanism reduces the overhead of Redis memory reclamation but causes higher memory usage. • Lazy eviction: When Redis needs to free up some memory space, it does not immediately release the memory. Instead, it marks certain key-value pairs as	Redis 2.8、4.0、5.0、6.2

	"to be released" and waits until the next access to release them. This approach reduces the overhead of Redis memory reclamation but causes higher memory usage. The parameter values are as follows. • no: Indicates that lazy deletion and lazy eviction are disabled. Redis immediately deletes key-value pairs or release memory space during memory reclamation. • Yes: Indicates that lazy deletion and eviction are enabled. During memory reclamation, Redis marks some key-value pairs as "to be deleted" or "to be released" and waits until the next access to delete or release them.	
lazyfree-lazy-expire	This parameter controls whether Redis performs lazy deletion (Lazy Free) when deleting expired keys. • no: Indicates that Redis does not perform lazy deletion. Redis immediately deletes the key-value pair when performing the deletion of expired keys. • yes: Indicates that Redis performs lazy deletion. When Redis performs key expiration deletion, it marks some key-value pairs as "to be deleted" and waits until the next access to delete them.	Redis 4.0、5.0、6.2
lazyfree-lazy-server-del	This parameter controls the lazy deletion behavior of Redis when executing the DEL command to delete key-value pairs. • no: Indicates that Redis does not perform lazy deletion, and immediately deletes the key-value pair when executing the DEL command. • Yes: Indicates that Redis performs lazy deletion. When executing the DEL command to delete key-value pairs, Redis marks some pairs as "to be deleted" and wait until the next access to delete them.	Redis 4.0、5.0、6.2
lazyfree-lazy-user-del	This parameter determines whether the default behavior of Redis is the same when executing the DEL command and the UNLINK command. The DEL command is used to delete one or more key-value pairs in Redis, while the UNLINK command is used to delete one or more key-value pairs in Redis with lazy deletion (Lazy Free) performed before the deletion. • no: Redis executes the DEL command to directly remove the key-value pair from memory without performing lazy deletion. • Yes: The default behavior of Redis is the same when executing the DEL command and the UNLINK command, which means that when executing the DEL command, Redis performs lazy deletion first, and then removes the key-value pair from memory.	Redis 4.0、5.0、6.2
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、6.2
read-local-node-only	When the instance is deployed in multiple AZs, this parameter specifies whether to enable or disable the proximity access feature.	Redis 4.0、5.0
sentineauth	This parameter controls whether omitting the Redis password and automatically using the password corresponding to the sentinelauth parameter specified in the Sentinel configuration file when interacting with Redis Sentinel by using the sentinel command.	6.2
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 notifications, all notifications are prefixed with __keyspace@<db>__. • E: Key event notifications, all notifications are prefixed with __keyevent@<db>__. • g: generic commands (non-type specific) like DEL, EXPIRE, RENAME, etc. • $: string commands • l: list commands • s: set commands • h: hash commands • z: sorted set commands • x: expired events (events generated every time a key expires) • e: evicted events (events generated when a key is evicted for maxmemory) • A: Alias for parameter g$lshzxe. Enabling keyspace notifications consumes CPU resources, so they are disabled by default. To define server notifications, input parameters must include K or E; for example, use "Ee" for subscribing to eviction event notifications in key events, and "AKE" for subscribing to all types of notifications.	Redis 2.8、4.0、5.0
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

Related APIs	API Meaning

ApplyParamsTemplate	Applies a parameter template.
CreateParamTemplate	Creates a parameter template.
DeleteParamTemplate	Deletes a parameter template.
DescribeInstanceParamRecords	Queries the list of parameter modifications.
DescribeInstanceParams	Queries the list of instance parameters.
DescribeParamTemplates	Queries the list of parameter templates.
ModifyInstanceParams	Modifies instance parameters.
ModifyParamTemplate	Modifies a parameter template.

Network and security
Configuring Security Group

Last updated: 2024-11-01 17:05:17

Scenario

Security group is a stateful virtual firewall with filtering capabilities, used to set network access control for single or multiple cloud databases. It is an important means of network security isolation provided by Tencent Cloud. A security group is a logical grouping, and you can add cloud database instances with the same network security isolation requirements within the same region to the same security group. Cloud databases share the security group list with CVMs and other instances, and the security group operates based on rule matching. For specific rules and restrictions, please refer to Security Group Detailed Description. You can bind a security group when purchasing an instance or bind it to an existing instance through the console.

Note:

- TencentDB for Redis® security groups currently only support network access control for VPCs and public networks but not the classic network.
- As TDSQL-A for PostgreSQL does not have active outbound traffic, outbound rules are not applicable to it.
- TencentDB for Redis® security groups support master instances, read-only replicas, and disaster recovery instances.

Security Group Configuration for TencentDB

Step 1. Create a security group

1. Log in to the CVM console.
2. Choose **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 from the drop-down list.
     - **Allow all ports**: By default, all ports are open to the public and private networks, which poses certain security risks. Security group rules have been added by default. Click Show Template Rules below to view the **outbound rules** and **inbound rules** of this security group template.
     - **Allow ports 22, 80, 443, 3389, and ICMP protocol**: By default, ports 22, 80, 443, 3389, and the ICMP protocol are allowed, with full access granted within the private network. Security group rules have been added by default.
     - **Custom**: After successfully creating a security group, add security group rules as needed.
   - **Name**: Set a custom name for the security group.
   - **Affiliated Project**: By default, the "Default Project" is selected. You can specify another project for easier management in the future.
   - **Notes**: A concise custom description of the security group for easier future management.
   - **Advanced Configuration**: You can add tags for the security group.
4. If **Template** is set to **Custom**, in the **Note** window, click **Set Now** 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 access source for the database, i.e., the inbound source. The following formats are supported for defining the source.

Source Format	Format Description
CIDR notation	• A single IPv4 address or an IPv4 rangeIn 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 rangeIn 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 Templates	Reference the IP address object or IP address group object in the Parameter Template.

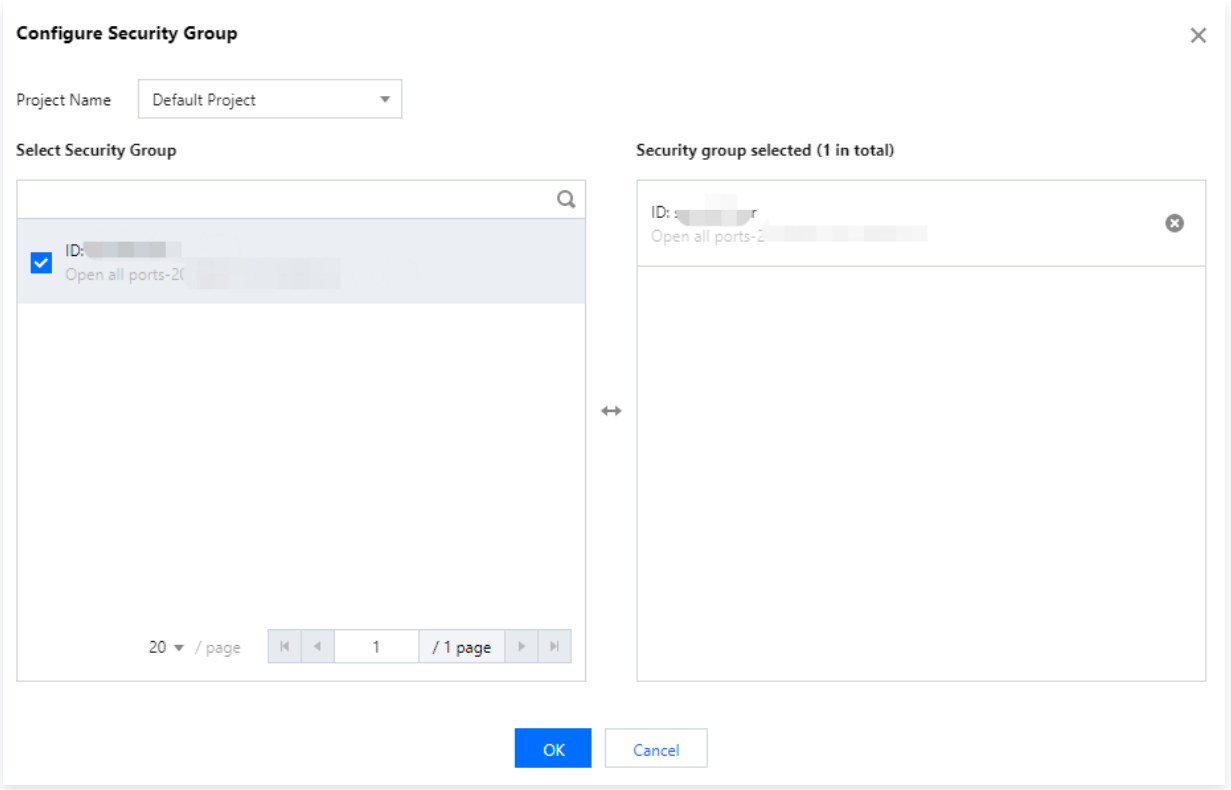
- **Protocol and Port**: Enter the protocol type and port for the client to access TencentDB for Redis®. You can view the port information on the Instance Details page under the **Network Info** section's **Private IPv4 Address**. The default port is 6379. If the access protocol is TCP, you can set it to TCP:6379.
- **Policy**: **Allow** is selected by default.
  - **Allow**: Access requests of this port are allowed.
  - **Reject**: Discards all data packets without any response.
- **Note**: Provide a brief, custom description of the rule for easier future management.

3. Click **Complete** to finish adding the inbound security group rule.

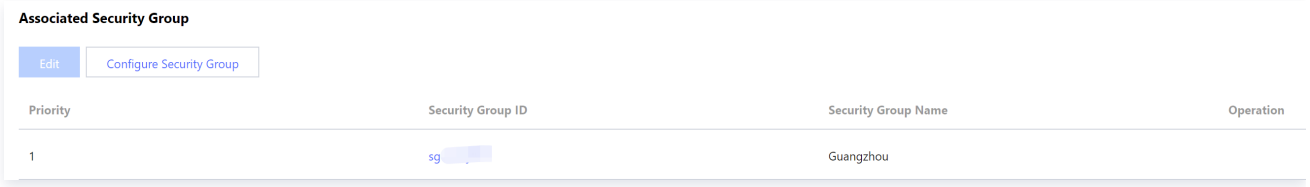
Step 3: Configure a security group for the database instance.

> **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 purchasing, you can follow the steps below to change or add security groups for the instance.

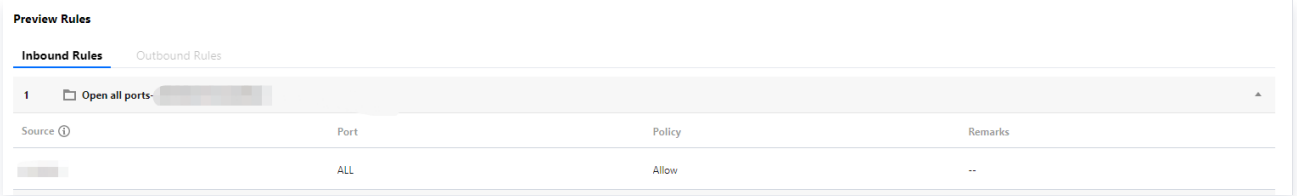
1. Log in to the TencentDB for Redis® console.
2. Above the **Instance List** page on the right, select the region.
3. In the instance list, find the target instance.
4. Click the instance ID to enter the instance management page.
5. On the **Security Group** tab, click **Configure Security Group**.
6. In the **Configure Security Group** pop-up window, select a created security group. You can filter security group by project name.



7. Click **OK** to complete the operation of binding the security group to the cloud database.
   - In the **Associated Security Group** section, you can view the list of security groups associated with the database instance.
     - Click **Security Group ID** to navigate to the **Security Group Rules** page, where you can edit the security group rules. For specific operations, see Modifying Security Group Rules. For more actions, see Managing Security Group Rules.
     - 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.



   - In the **Rule Preview** section, you can view the inbound source information of the security group on the **Inbound Rules** tab.



More Operations

- For more security group operations, see Managing Security Groups.
- For more security group rule operations, see Managing Security Group Rules.
- For security group APIs, see Security Group Related Interfaces.

Related APIs

Related APIs	API Meaning
DescribeProjectSecurityGroup	Queries the security group information of project
DescribeInstanceSecurityGroup	Queries the security group information of instance
ModifyDBInstanceSecurityGroups	Modifies security groups bound to instance
AssociateSecurityGroups	This API is used to bind a security group.
DisassociateSecurityGroups	Unbinding security groups in batches
DescribeProjectSecurityGroups	Queries the security group details of a project
DescribeDBSecurityGroups	Queries the details of a security group

Configuring Public Network Address

Last updated: 2024-11-01 17:05:27

This guide illustrates how to enable or disable public network access to Redis instances via the Redis console. By utilizing the system-assigned domain name and port, users can conveniently access TencentDB for Redis® over the public network. Enabling public network access to facilitate daily testing and management, enhancing the ease of development and usage for users.

Note:
- The instance service downtime caused by public network errors won't be counted into the "Single Instance Service Downtime" in 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.

Supports and Limits

- 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, which is 6379 by default, and public network port) as instructed in Configuring Security Group.

Usage Limits

- The public network address can only be configured for instances in VPCs. 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.

  To use it in other regions, you can access the TencentDB for Redis® instance through iptables-based forwarding. For more information, see iptables Forwarding.

Preparations

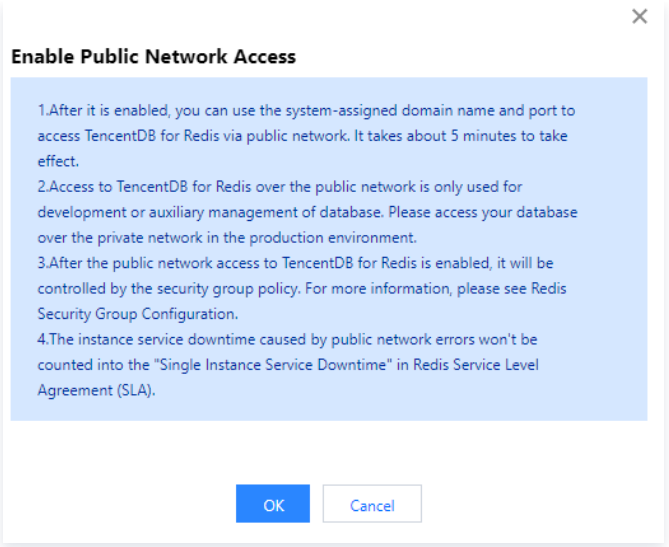
To enable public network access, you need to disable password-free access.

Enabling the Public Network Address

1. Log in to the TencentDB for Redis® console and click an instance ID in the instance list to enter the instance details page.
2. In the **Network Information** section on the instance details page, click **Enable** next to the **Public Network Address**.

   Note:
   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, which is 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 remains unchanged for a long time, refresh the page.
5. When the **Public Network Address** displays an address containing a domain name and port, it indicates that the public network access has been successfully enabled. In this case, you can access the Redis database over the public network.

Disabling the Public Network Address

1. Log in to the TencentDB for Redis® console and click an instance ID in the instance list to enter the instance details page.
2. In the **Network Info** section on the right side of the Instance Details page, click **Disable** next to the **Public Network Address**.
3. In the pop-up dialog box, 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. The **Public Network Address** will no longer be displayed, indicating that the public network access is disabled.

Related APIs

API	Note
AllocateWanAddress	Enables public network access
ReleaseWanAddress	Disables public network access

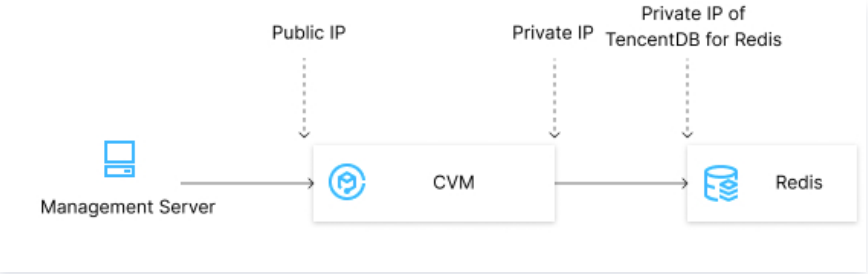
iptables Forwarding

Last updated: 2024-11-01 17:05:35

Scenario

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:**
> Port forwarding with iptables is not stable, so we do not recommend this public network access solution in a production environment.



Instructions

1. Log in to the Cloud Server and enable the IP forwarding feature for the CVM instance.

   > **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 following sample code is to forward access requests of `26.xx.x.2:10001` (CVM public IP and customizable port) to a Redis instance whose private IP and port are `10.0.0.5:6379`.

   ```
   iptables -t nat -A PREROUTING -p tcp --dport 10001 -j DNAT --to-destination 10.0.0.5:6379
   iptables -t nat -A POSTROUTING -d 10.0.0.5 -p tcp --dport 6379 -j MASQUERADE
   ```

3. Configure the CVM security group to allow access to the public network ports of the CVM instance. The security group rules recommend you to only open the source addresses that need to be accessed.
4. To connect to the private network Redis instance via the public network address (in this example, 26.xx.xx.2:10001), use the same connection command as for private network access. For the command, see Connecting with Client Tools.
5. After connecting to TencentDB for Redis®, running the `info` command and receiving database-related information indicates a successful connection.

Accessing Database over Public Network

Last updated: 2024-11-01 17:05:48

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, for example, `D:\Temp\Redis-x64-3.2.100`.
2. On your local device, use the Windows key + R to open the **Run** dialog box, type in cmd, and click **OK** to open the Windows command line window.
3. Run the following command to enter the installation directory of redis-cli.

```
cd /d <path>
```

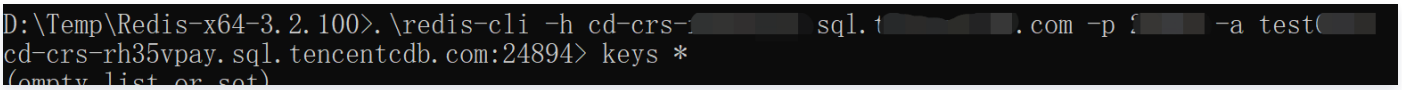
Here, path refers to the installation directory of redis-cli. For instance: cd /d D:\Temp\Redis-x64-3.2.100

4. Run the following command to access the database:

```
redis-cli -h <hostname> -p <port> -a <password>
```

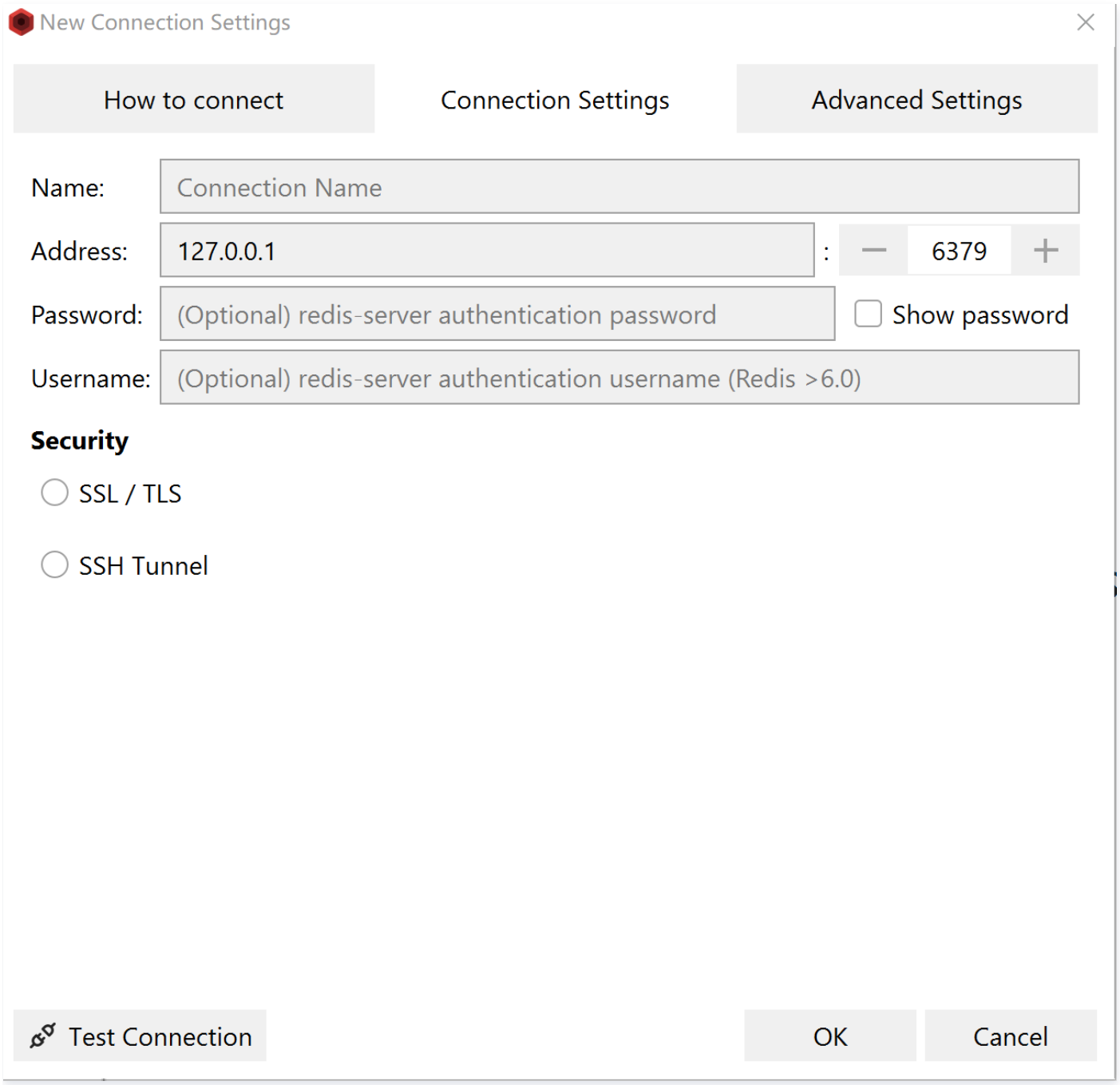
Here, hostname refers to the public network address of the database instance; port is the network port corresponding to the public network address; password is the default access password for the instance account. If a custom account is used for connection, the authentication method for the custom account is `account name@password`, which serves as the password parameter for accessing Redis.

For execution examples, please refer to the image below.



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 name	Description
Name	Name of the connection to the database instance.
Address	Enter the public network address and port number of the database instance.
Verification	Enter the database instance connection password. If using the default account, directly enter the instance access password; if using a custom account, authenticate with `account name@password` .

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 refers to the public network address of the database instance; port refers to the network port corresponding to the public network address; password refers to the access password of the default account for the instance. If a custom account is used for connection, the authentication method for the custom account is `username@password`, which is used as the password parameter for accessing Redis.

Bandwidth Adjustment

Last updated: 2024-11-01 17:06:04

Scenario

The network bandwidth requirements differ among various instance specifications. If traffic surpasses the bandwidth limit, it may lead to congestion and adversely affect service performance. For instance, during time-sensitive flash sales that generate concentrated traffic peaks, or when numerous large key read and write operations occur temporarily, you can swiftly enhance the bandwidth by adjusting the instance's bandwidth, thus preventing any negative impact on the business.

Billing description

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.

Supports and Limits

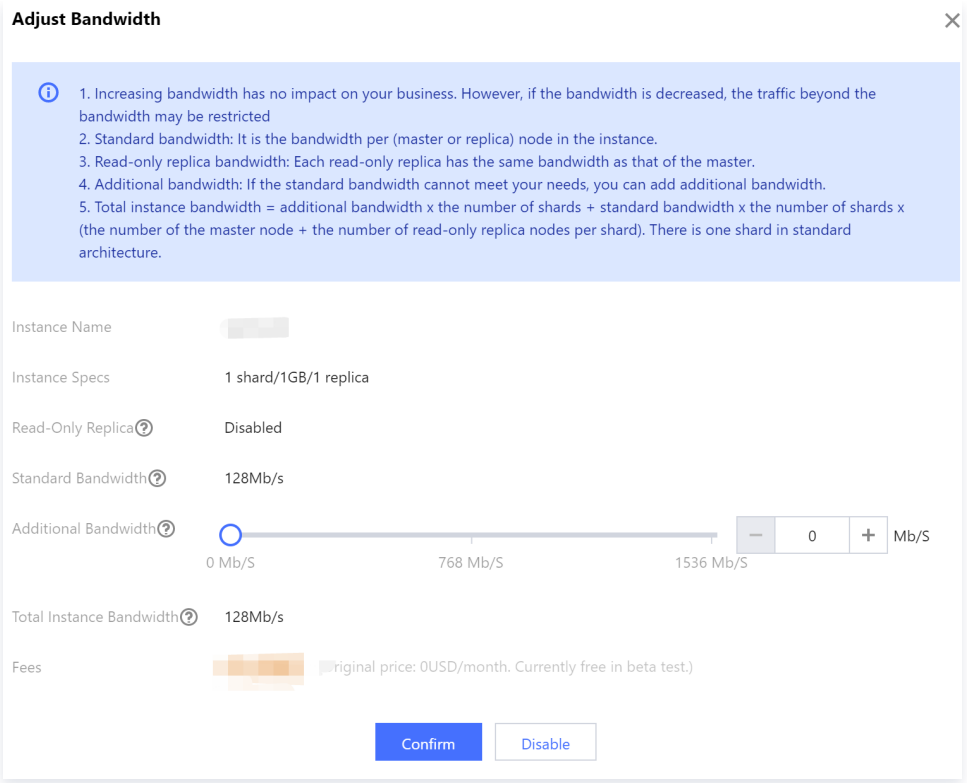
Increasing the bandwidth will not affect your business, but reducing the bandwidth may cause throttling of the traffic that exceeds the bandwidth.

Preparations

- 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.

Instructions

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 on the instance ID, and on the **Instance Details** page, navigate to the **Network Information** section. Click **Adjust Bandwidth** next to **Maximum Network Throughput**.
5. In the **Adjust Bandwidth** dialog box, you can select the desired additional bandwidth on the slider bar next to **Additional Bandwidth** according to your needs.



Parameter name	Description
Instance Name	The instance name.
Instance Specification	The instance specification: Shard quantity, memory, and replica quantity.
Read-Only Replica	The read-only replica status.
Standard Bandwidth	The bandwidth per (master or replica) node in the instance.
Additional Bandwidth	Select the additional bandwidth on the slider bar.
Total Instance Bandwidth	• When 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.
Cost	Free of charge currently.

6. After confirming that the total bandwidth meets your expectations, click **Confirm Adjustment**.
7. When the **Instance Status** changes to **In Progress**, wait for it to become **Running**. In the **Instance Details** page, under the **Network Information** section, you can view the updated **Maximum Network Throughput** as the total bandwidth.

Related APIs

API Name	Feature
ModifyNetworkConfig	The API is used to modify the instance network configuration to adjust the

	bandwidth.

SSL encryption

Last updated: 2024-11-01 17:06:14

Scenario

Secure Sockets Layer (SSL) authentication is the process of verifying clients and cloud database servers. By enabling SSL encryption, you can obtain a CA certificate and upload it to the server. When accessing the database from the client, the SSL protocol is activated, establishing an SSL secure channel between the client and the database server. This ensures encrypted data transmission, preventing data interception, tampering, and eavesdropping, and guaranteeing the security of information exchanged between both parties.

> **Note:**
> SSL encryption is gradually being rolled out. To experience it in advance, please submit a ticket to apply.

Billing description

SSL encryption is free of charge.

Notes

- Enabling SSL encryption ensures the security of data access and transmission 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

- Release notes

  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, please 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 6.2, the feature can be enabled after the proxy is upgraded to 5.6.0. For more information, see Upgrading Proxy.
- Architecture

  Both standard architecture and cluster architecture support SSL encryption.

Preparations

- 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.

Instructions

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**, as shown in the image below. Click **Download Certificate** in the upper right corner.

8. Wait until **Enable SSL** status becomes **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

Related APIs	API Function
OpenSSL	Enabling SSL
CloseSSL	The API is used to disables SSL.

Monitoring and Alarming
Update Notes of Monitoring at Five-second Granularity

Last updated: 2024-11-01 17:08:04

Changes

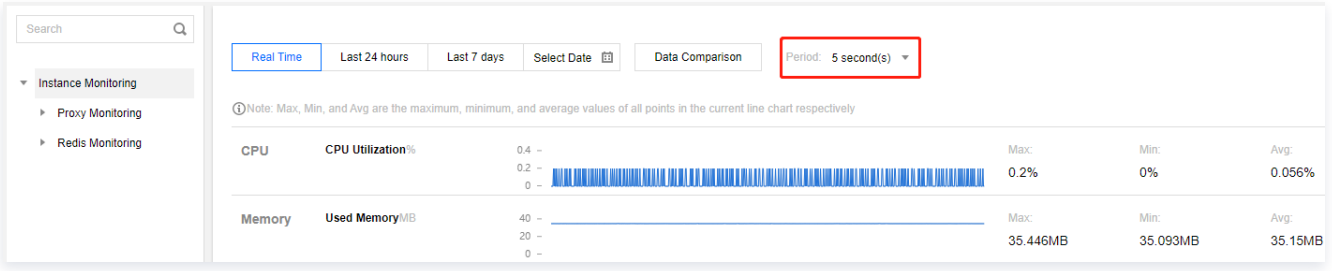
The monitoring feature of TencentDB for Redis® has been comprehensively upgraded, with the main updates as follows:

- 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?

- Access the TencentDB for Redis® console, click on an instance ID to navigate to the instance management page, select **System Monitoring** > **Monitoring Metrics**. If the instance's monitoring time granularity supports 5 seconds, it indicates that this instance supports a 5-second monitoring granularity, otherwise, the instance operates on a 1-minute monitoring granularity.



- The granularity can be determined by the InstanceSet.MonitorVersion field returned by the API interface DescribeInstances: MonitorVersion = 5s indicates a granularity of five seconds, while MonitorVersion = 1m indicates a granularity of one minute.

How do I get information of Proxy or Redis nodes?

You can obtain the Proxy node ID and Redis node ID through the API DescribeInstanceNodeInfo.

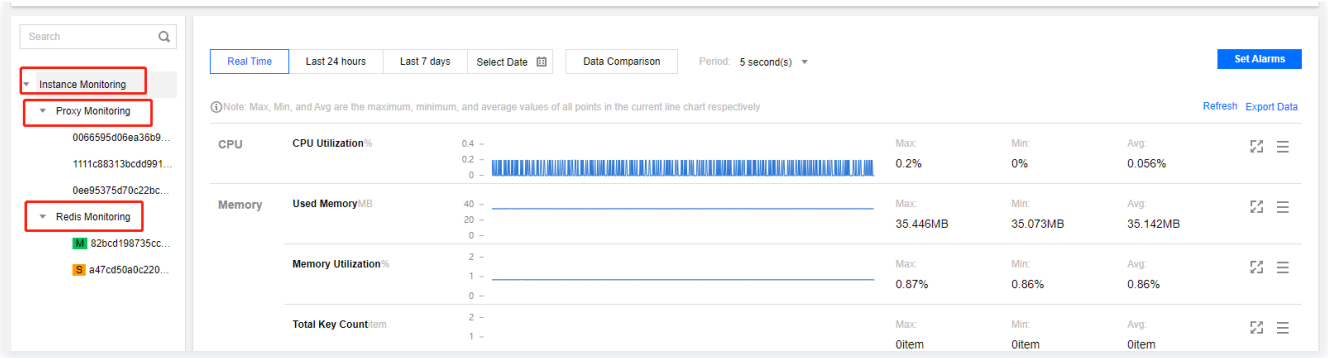
> **Note:**
> The IDs of Proxy and Redis nodes will change when node failover, instance capacity expansion/reduction, data migration, etc., occur. Therefore, we recommend that you get the latest node information from the API in a timely manner.

Monitoring at Five-Second Granularity

Last updated: 2024-11-01 17:08:16

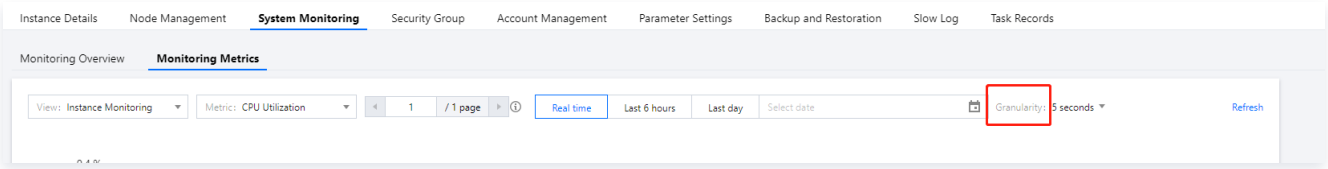
TencentDB for Redis® provides a complete and easy-to-use monitoring service at a 5-second granularity where you don't have to worry about, for example, collecting monitoring data or Ops of the monitoring system. The monitoring service includes Proxy monitoring, Redis monitoring, and instance monitoring which summarizes the monitoring data of an entire instance.

- Proxy monitoring: provides monitoring information of all Proxy nodes in an instance. TencentDB for Redis® instances in standard or cluster architecture have Proxy nodes.
- Redis monitoring: provides monitoring information of master and replica Redis nodes.
- Instance monitoring: summarizes the monitoring data of an entire instance (including Proxy nodes and Proxy nodes) and aggregates data according to the SUM, AVG, MAX, and LAST aggregation algorithms.



Viewing Instance 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**. If the instance monitoring supports a 5-second granularity, it means the instance supports 5-second monitoring granularity. Redis currently supports monitoring metrics at granularities of 5 seconds, 1 minute, 5 minutes, 1 hour, and 1 day. For data retention periods of each granularity, please refer to Usage Constraints.



- The granularity of instance monitoring can be determined by the InstanceSet.MonitorVersion field returned by the DescribeInstances API. If MonitorVersion = 5s, it indicates a granularity of 5 seconds; if MonitorVersion = 1m, it indicates a granularity of 1 minute.

Viewing monitoring data

You can view the monitoring information of TencentDB for Redis® instances through the Redis Instance List, Redis System Monitoring page, and Tencent Cloud Observability Platform (TCOP) 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 to enter the **Instance Details** page. Then, click the **System Monitoring** tab and select the **Monitoring Metrics** tab to view the monitoring data.

Access monitoring data via Tencent Cloud Observability Platform.

Log in to the Tencent Cloud Observability Platform console to view the summary of monitoring data.

Monitoring Metrics Description

Proxy node monitoring

Each TencentDB for Redis® instance contains at least three 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	Metrics	Metric Name	Unit	Metric Description
CPU	CPU utilization	cpu_util	%	Proxy CPU utilization
Requests	Total requests	proxy_commands	Queries/second	The number of Proxy command executions per second
	Key requests	cmd_key_count	keys/second	The number of keys accessed by a command per second
	Mget requests	cmd_mget	Queries/second	The number of Mget command executions per second
	Execution Errors	cmd_err	Queries/second	The number of Proxy command execution errors per second, for example, when a command does not exist, parameters are incorrect, etc.
	Big value requests	cmd_big_value	Queries/second	The number of executions of commands larger than 32 KB per second
Network Monitoring	Connection Count	connections	Connections	Number of TCP connections to an 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 connection utilization among all Proxy nodes is calculated by taking the highest ratio of the current number of connections to the maximum number of connections for each node.
	Inbound Traffic	in_flow	Mb/s	Private Inbound Traffic
	Inbound traffic utilization	in_bandwidth_util	%	The ratio of the actually used private inbound traffic to the

				maximum traffic
	Inbound traffic limit count	in_flow_limit	Threads	The number of times inbound traffic triggers a traffic limit
	TrafficOut	out_flow	Mb/s	Private Outbound Traffic
	Outbound traffic utilization	out_bandwidth_util	%	The ratio of the actually used private outbound traffic to the maximum traffic
	Outbound traffic limit count	out_flow_limit	Threads	The number of times outbound traffic triggers a traffic limit
Latency monitoring	Average execution latency	latency_avg	ms	Average execution latency from Proxy to Redis server
	Max execution latency	latency_max	ms	Maximum execution latency from Proxy to Redis server
	Average read latency	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For read command categories, please refer to Command Categories.
	Average write latency	latency_write	ms	The average execution latency of writing commands between the proxy and the Redis server. For write command categories, see Command Categories.
	Average latency of other commands	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server

Redis node monitoring

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

Group	Metrics	Description	<p>	MetricsDescription
CPU monitoring	CPU utilization	cpu_util	%	Avg CPU usage
Networking	Connection Count	connections	Connections	The number of connections from Proxy to a node
	Connection utilization	connections_util	%	The connection usage of a node
Memory monitoring	MEM Usage	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	mem_util	%	The ratio of the memory actually used to the total memory requested
	Keys	keys	Connections	Total number of keys stored in an instance (first-level keys)

	Expired keys	expired	Connections	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	Connections	The number of keys evicted in a time window, which is equal to the value of `evicted_keys` output by the `info` command
	Replication Delay	repl_delay	Byte	The command delay between the secondary node and the primary node
Request Monitoring	Total requests	commands	Queries/second	QPS, that is, the number of command executions per second
	Read Requests	cmd_read	Queries/second	Number of read command executions. For more information on read command types, see Command types.
	Write Request	cmd_write	Queries/second	Number of write command executions. For more information on write command types, see Command types.
	Other Requests	cmd_other	Queries/second	Executions of commands other than read and write commands
Response Monitoring	Slow query	cmd_slow	Threads	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read request hits	cmd_hits	Threads	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	Threads	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 reflects the cache miss situation.

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	Meaning	Associated View	Metric	<p>	MetricsDescription

CPU monitoring	CPU utilization	Redis node	cpu_util	%	Avg CPU usage
	Max CPU utilization of a node	Redis node	cpu_max_util	%	The maximum CPU utilization of a node (shard or replica) in an instance
Memory monitoring	MEM Usage	Redis node	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	Redis node	mem_util	%	The ratio of the memory actually used to the total memory requested
	Max memory utilization of a node	Redis node	mem_max_util	%	The maximum memory utilization of a node (shard or replica) in an instance
	Keys	Redis node	keys	Connections	Total number of keys stored in an instance (first-level keys)
	Expired keys	Redis node	expired	Connections	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	Redis node	evicted	Connections	The number of keys evicted in a time window, which is equal to the value of `evicted_keys` output by the `info` command
Network Monitoring	Connection Count	Proxy node	connections	Connections	Number of TCP connections 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 Inbound Traffic
	Inbound traffic utilization	Proxy node	in_bandwidth_util	%	The ratio of the actually used private inbound traffic to the maximum traffic
	Inbound traffic limit count	Proxy node	in_flow_limit	Threads	The number of times inbound traffic triggers a traffic limit
	TrafficOut	Proxy node	out_flow	Mb/s	Private Outbound Traffic

	Outbound traffic utilization	Proxy node	out_bandwidth_util	%	The ratio of the actually used private outbound traffic to the maximum traffic
	Outbound traffic limit count	Proxy node	out_flow_limit	Threads	The number of times outbound traffic triggers a traffic limit
Latency	Average execution latency	Proxy node	latency_avg	ms	Average execution latency from Proxy to Redis server
	Max execution latency	Proxy node	latency_max	ms	Maximum execution latency from Proxy to Redis server
	Average read latency	Proxy node	latency_read	ms	The average execution latency of read commands between the proxy and the Redis server. For read command categories, please refer to Command Categories.
	Average write latency	Proxy node	latency_write	ms	The average execution latency of writing commands between the proxy and the Redis server. For write command categories, see Command Categories.
	Average latency of other commands	Proxy node	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server
	P99 Latency	Proxy node	latency_p99	ms	The P99 latency between the proxy and the Redis server
Request Monitoring	Total requests	Redis node	commands	Queries/second	QPS, that is, the number of command executions per second
	Read Requests	Redis node	cmd_read	Queries/second	Number of read command executions. For more information on read command types, see Command types.
	Write Request	Redis node	cmd_write	Queries/second	Number of write command executions. For more information on write command types, see Command types.

	Other Requests	Redis node	cmd_other	Queries/second	Executions of commands other than read and write commands
	Big value requests	Proxy node	cmd_big_value	Queries/second	The number of executions of commands larger than 32 KB per second
	Key requests	Proxy node	cmd_key_count	keys/second	The number of keys accessed by a command per second
	Mget requests	Proxy node	cmd_mget	keys/second	The number of Mget command executions per second
	Slow query	Redis node	cmd_slow	Threads	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read request hits	Redis node	cmd_hits	Threads	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	Threads	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	Threads	The number of command execution errors, for example, when a 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 reflects the cache miss situation.

Command Classification

Command Classification	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,geodist,pfcount
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,georadiusbymember, pfadd,pfmerge,pfdebug

Querying Node Information

The Proxy node ID and Redis node ID can be obtained through the API DescribeInstanceNodeInfo.

> **Note:**
> 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.

Related APIs

Related APIs	API Meaning
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 of an instance

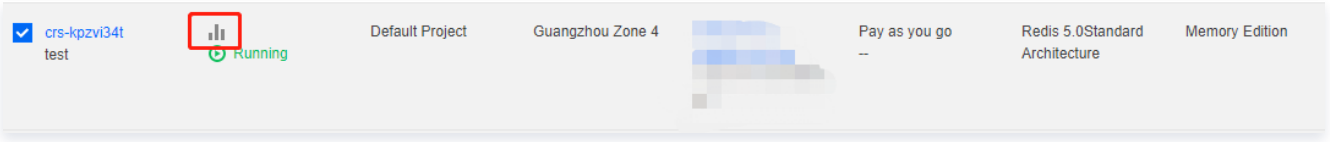
Comparing Monitoring Data Among Instances

Last updated: 2024-11-01 17:08:30

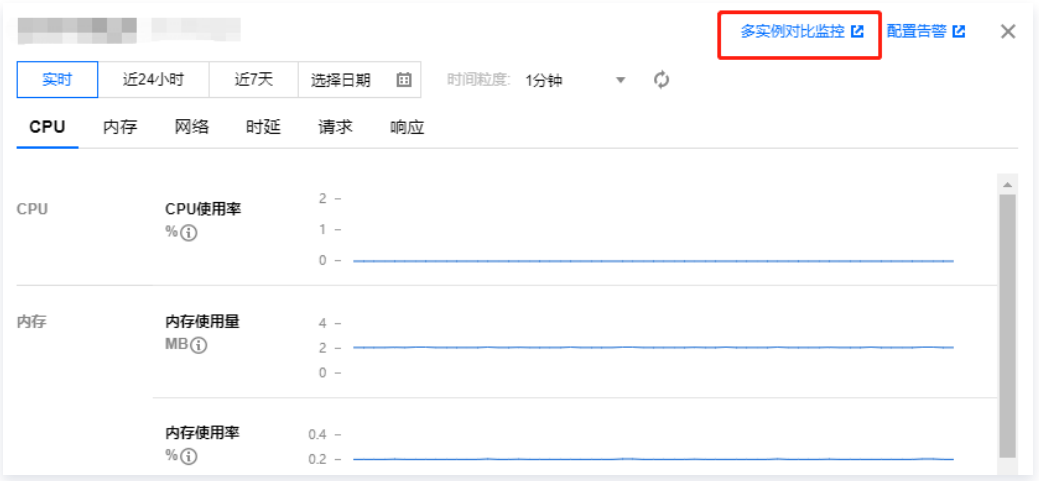
TencentDB for Redis® offers multi-instance comparison monitoring, allowing you to conveniently troubleshoot discrepancies within a single chart. This document guides you on how to set up multi-instance comparison monitoring through the TencentDB for Redis® console.

Instructions

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.



3. On the Dashboard page of the Tencent Cloud Observability Platform, create a dashboard and a chart to view and compare monitoring data of multiple instances within the same chart.

Configuring Alarm

Last updated: 2024-11-01 17:08:38

Scenario

To prevent certain monitoring metrics from affecting the normal operation of your system when they reach a specific value, you can establish alarm rules for these metrics. This prompts the alarm system to automatically inspect monitoring data and send notifications to administrators when the data meets the specified conditions. This enables you to promptly identify and address any business anomalies.

Billing description

- TCOP allows you to configure alarm policies to monitor the key metrics of instances and offers a free trial.
- Currently, charges apply only to **alert SMS and phone calls**. For more information, see Tencent Cloud Observable Platform Billing Overview.

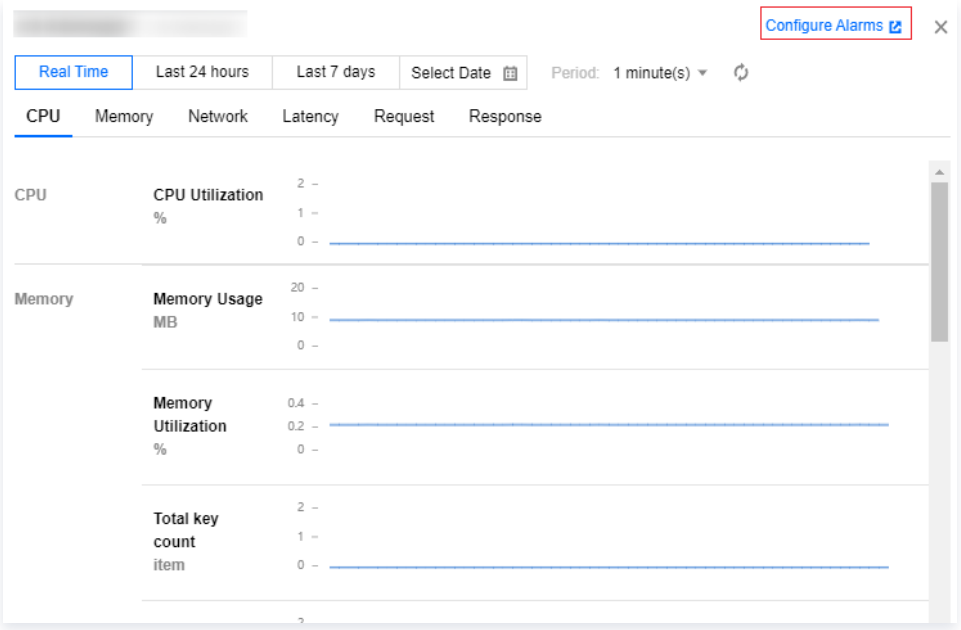
Preparations

- Activate the Tencent Cloud Observability Platform (TCOP) service.
- 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.

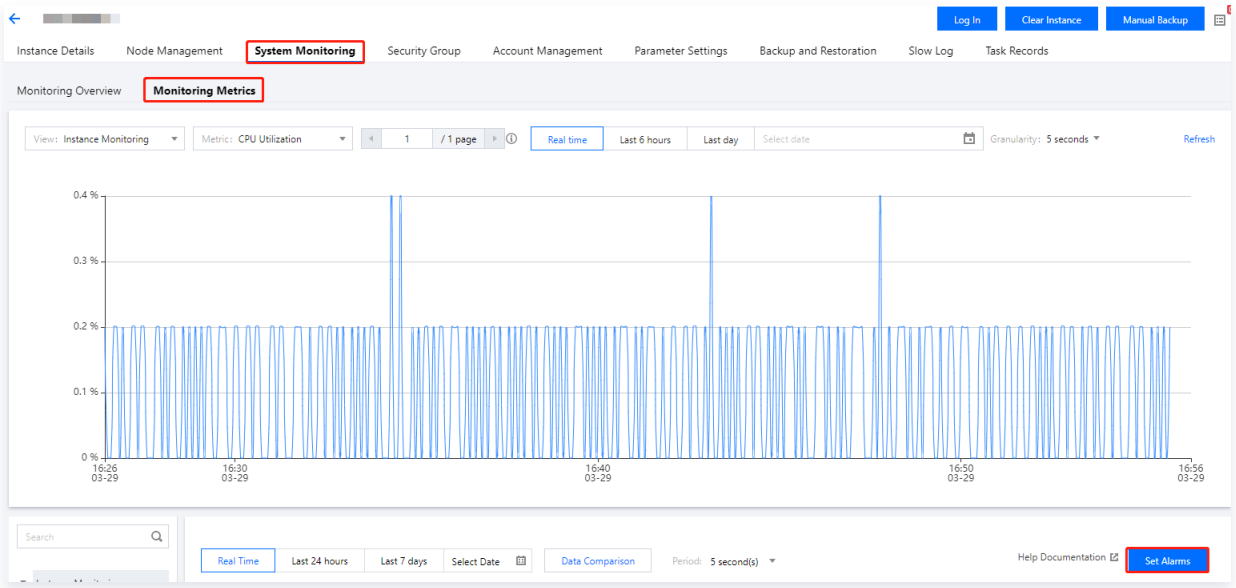
Instructions

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 Alarm Policy** page of TCOP in any of the following ways:

- Click in the **Monitoring/Status/Task** column, and then click **Configure Alarms** in the top-right corner of the instance monitoring data panel.



- 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 Alarm Policy** page, configure the alarm policy according to the table below. For basic concepts of alarm policies, please refer to Creating Alarm Policy.

Parameter name	Description
Rule Name	Customize the alarm policy name for easier identification
Remarks	Briefly describe the alarm policy for easier identification
Test mode	Select **Cloud Product Monitoring**.
Policy types	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.
Alert object	• Select **Instance ID**: associate the alarm policy with the specified database instance. • Select **Instance Group**: The alarm policy will be associated with the specified database instance group. For information on creating an instance group, see Instance Group. • Select **All Objects**: The alarm policy will be bound to all instances the current account has permission on. • Select **Tag**: The alarm policy will be bound to all instances associated with the current tag key and tag value.
Trigger condition	• Select **Choose Template**: Choose a template file from the drop-down list, and the alarm will be reported based on the preset trigger conditions in the template file. For specific configurations, see Configure Trigger Condition Template. Select **Manual Configuration**: You need to configure the threshold conditions for triggering alarms for each metric individually in the **Metric Alarm** area. • In the **Metric Alarm** area, select the **Threshold Type**: Choose Static for a constant threshold set manually, which triggers an alarm when the conditions are met; Choose **Dynamic** for a threshold boundary determined by machine learning algorithms to detect anomalies. For more information, see Creating Alarm Policy.
Alert notifications	You can choose between system preset notification templates and user-defined templates, with a maximum of three notification templates bound to each alarm policy. For more information, see Notification Templates.

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: 2024-11-01 17:08:48

Scenario

TencentDB for Redis® has been integrated with Tencent Cloud Observability Platform and supports reporting events to the platform. All Tencent Cloud Observability Platform events will be automatically delivered to the Cloud Service Event Set in Event Bus (EventBridge).

Tencent Cloud Observability Platform Events

The currently supported **Tencent Cloud Observability Platform events** are shown in the table below.

EventName	EventParameter	Event type	Dimension	Recoverable	Event description	Troubleshooting Methods
Active/Standby switch	MasterSlaveSwitched	Status changed	TencentDB for Redis® instance	-	A TencentDB for Redis® switch occurs	This event will cause Redis service disconnection and brief unavailability. Check whether your business has an automatic reconnection mechanism to ensure fast business recovery
Unavailable service	ServiceNotAvailable	Errors	TencentDB for Redis® instance	Yes	A TencentDB for Redis® fault occurs and the service is unavailable	We will recover the service as soon as possible and send a service recovery notice when the service is recovered. If you have a disaster recovery instance, try to switch your business over to it
Read replica failover	ReadonlyReplicaSwitched	Status changed	TencentDB for Redis® instance	Yes	A TencentDB for Redis® read-only replica switch occurs	We will recover the service as soon as possible and send a service recovery notice when the service is recovered. If you have a disaster recovery instance, try to switch your business over to it or add a read-only replica
The read-only replica is unavailable	ReadonlyReplicaNotAvailable	Errors	TencentDB for Redis® instance	Yes	A TencentDB for Redis® read-only replica fault occurs	We will recover the service as soon as possible and send a service recovery notice when the service is recovered. If you have a disaster recovery instance, try to switch your business over to it

						or add a read-only replica
Instance migration due to server failure	Serverfailure InstanceMigration	Status changed	Tencent DB for Redis® instance	Yes	A server error occurs in TencentDB for Redis®.	The system detects that there is a hardware risk on the server and automatically performs a switch. The switch time is subject to the maintenance time. To change the time, do it promptly.

Delivery Target

An event rule can have multiple event targets. Before creating an event rule, please plan the event target types. EventBridge currently supports the following **event targets**.

- Message Push Target (supported by rules only in the Tencent Cloud service event bus)
- CLS Logs
- SCF
- Ckafka

Billing description

Tencent Cloud offers EventBridge with a **pay-as-you-go** pricing model. For more information, see Event Bus > Product Pricing.

Type	Pay-as-you-go
Payment Method	Fees are charged hourly by the number of events actually delivered to the event bus.
Billing Unit	CNY/million events
Use Cases	Low or fluctuating message volumes

Instructions

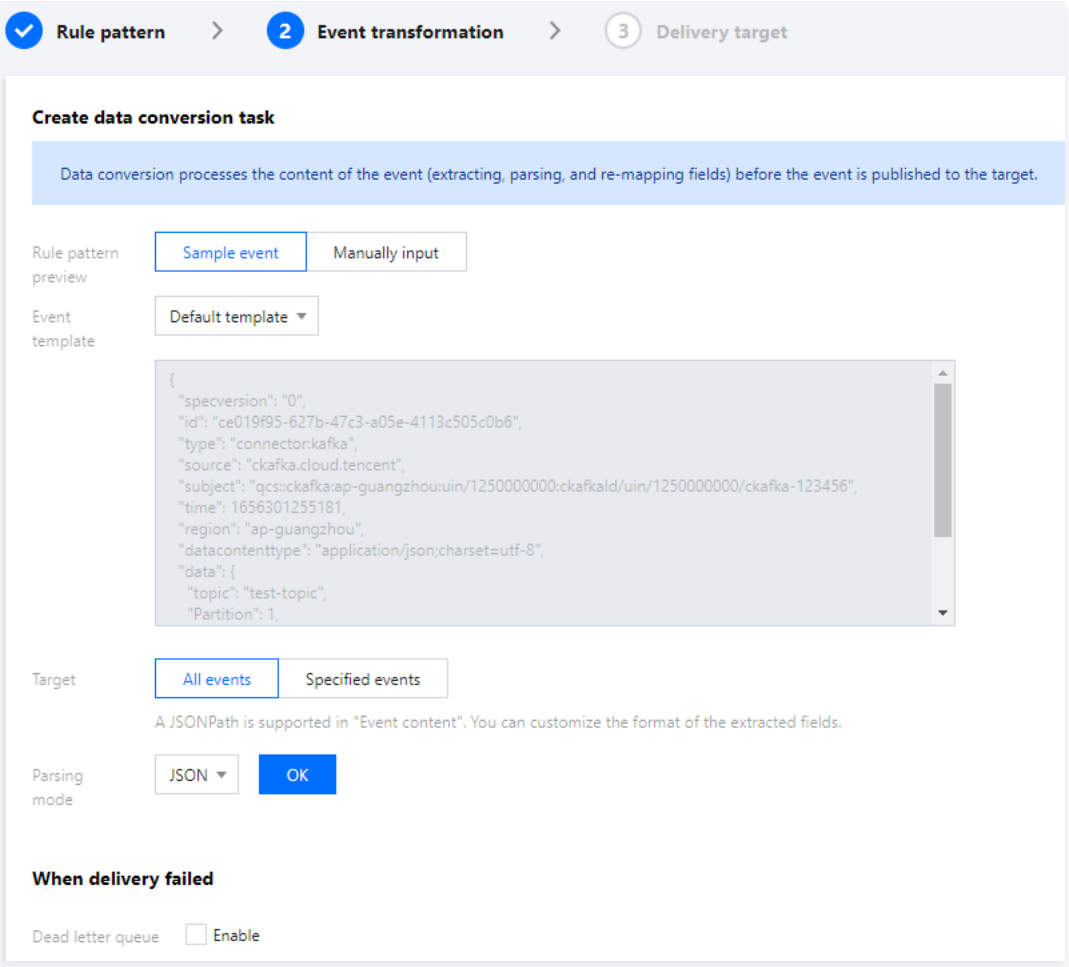
1. Log in to the EventBridge console and choose Event Rules in the left sidebar.
2. At the top of the right-hand page, select **Region** as **Guangzhou**, and choose **default** from the **Event Set** dropdown list.
   - The Cloud Service Event Set is designed to collect monitoring and auditing events generated by Tencent Cloud services across all regions. It is created by default in Guangzhou and cannot be deleted.
   - Choose **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	Interface Parameter	Description
Basic Information	Regions	The region where to create an event rule.
	Event Set	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 Transformation	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	Mode	• Form Mode: In this mode, you can select **Cloud Service Type** and **Event Type** to generate the event matching rules. • Custom events: Customize the event matching rule.
	Tencent Cloud service	When **Editing Mode** is set to **Form Mode**, this parameter is displayed. In the drop-down list, select **TencentDB for Redis®**.
	Event type	When **Editing Mode** is set to **Form Mode**, this parameter is displayed. In the drop-down list, select a supported event type.
	Event matching rule preview	Preview the generated event matching rule.

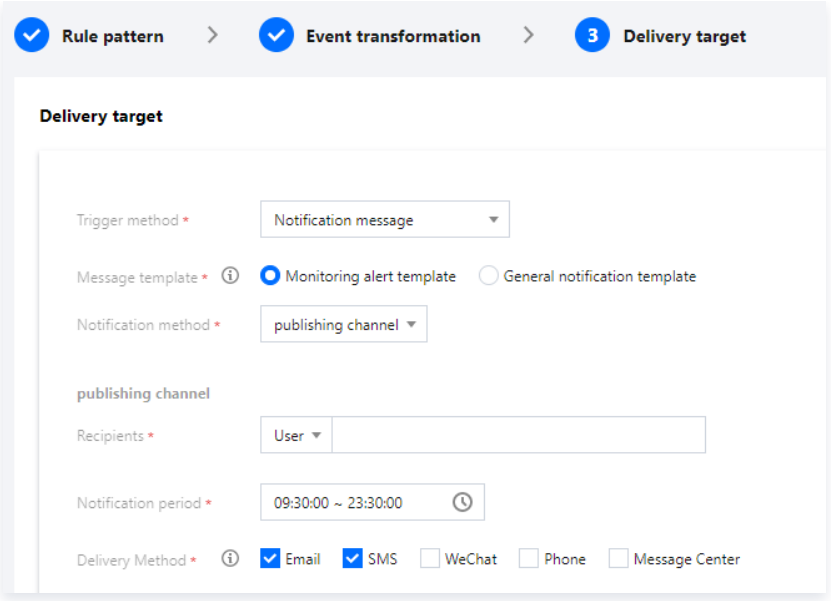
4. Click **Test Matching Rule** to test the defined event matching rules.
5. (Optional) To convert the data format, display the event transformation page as shown below. Configure the data conversion formats and fields according to the parameter explanations in the following table, and click **Confirm** to start parsing the data. Wait until the data parsing is complete, and set the filter rules and data processing methods. For details, see Configuring Data Conversion.

> **Note:**
> 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.



Interface Parameter	Description
Rule pattern preview	Select **Sample Event** to use an event template, or choose Manual Input to customize event fields in the input box below.
Event template	In the **Rule Pattern Preview**, select **Sample Event** to display the parameter. In the drop-down list, search for Redis and select a Redis event template. The input box below will display the specific field information of the selected event template.
Target	• **Full events**: Route the entire event field structure to the event target. • **Partial events**: EventBridge extracts event parameters from CloudEvents through the JSONPath-configured event fields, routing the specified event fields to the event target.
JSONPath	When selecting specific events for conversion, this parameter is displayed. Please 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, check **Enable event rule now** and click **Finish**.

Event Rule APIs

API Name	Feature
CheckRule	Check 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

To view, edit, or delete an event rule, see Managing Event Rules.

FAQs Overview

For inquiries regarding event rule concepts and billing-related common questions, see Event Bus > Frequently Asked Questions.

Viewing Alarms

Last updated: 2024-11-01 17:08:56

Scenario

You can view the following three types of alarms triggered within a specific time frame in the Tencent Cloud Observability Platform (TCOP) console. Please note that alarm information with customized messages is not currently supported for viewing in the console.

- Basic alarms: Alarms sent by preset alarming metrics.
- Cloud probe alarms: Alarms sent by Cloud Probe.
- Customized alarms: Alarms sent by the customized monitoring feature.

Instructions

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 customize a 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 about how to manage alarm policies, see Alarm Policies.
- For more information about how to manage alarm notifications, see Creating Notification Template.

Global Replication Overview

Last updated: 2024-11-01 17:09:25

TencentDB for Redis® supports global replication for consistent data sync across regions.

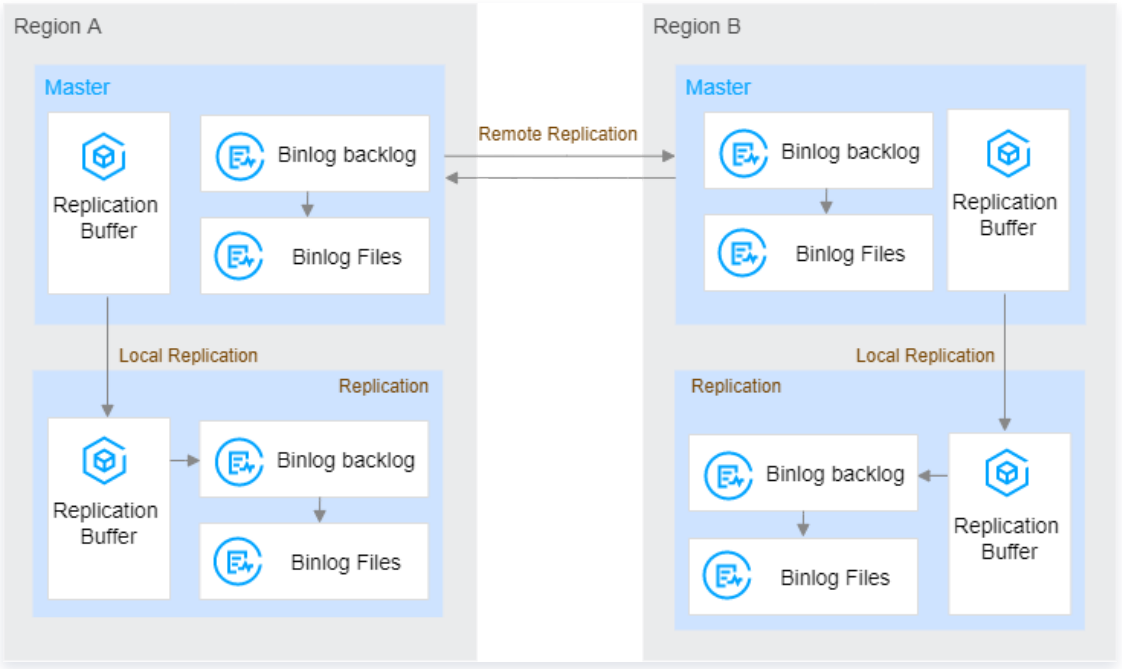
Why Global Replication

In cross-region replication scenarios with Redis native code, when the local instance has a high write volume or replication is disconnected for an extended period, the master node's backlog replication log may be insufficient for resuming incremental replication. This results in the remote instance having to perform a full replication, which can hinder access to the remote instance and severely impact its normal operations. For more information, see Redis Official Documentation on 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 feature added by Tencent Cloud to the open-source community version kernel, fully compatible with Redis 4.0 and Redis 5.0 commands. Based on the original master-slave replication scheme, a new log file is added for remote replication, ensuring eventual data consistency for instances in any region within 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.

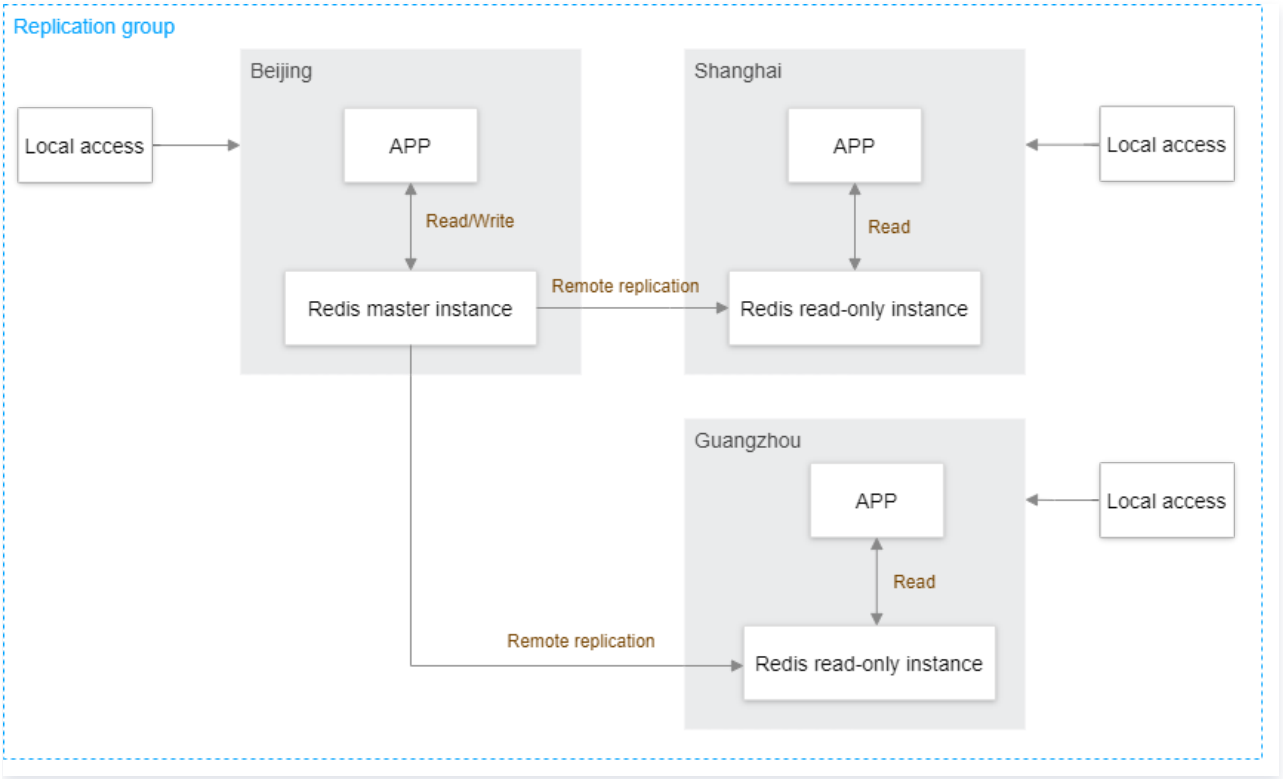


Scenarios

Read-only instance and disaster recovery

Redis Global Replication scheme configures a primary instance within a replication group and deploys read-only instances across multiple regions. Read-only instances in different regions replicate data from the primary instance. The primary

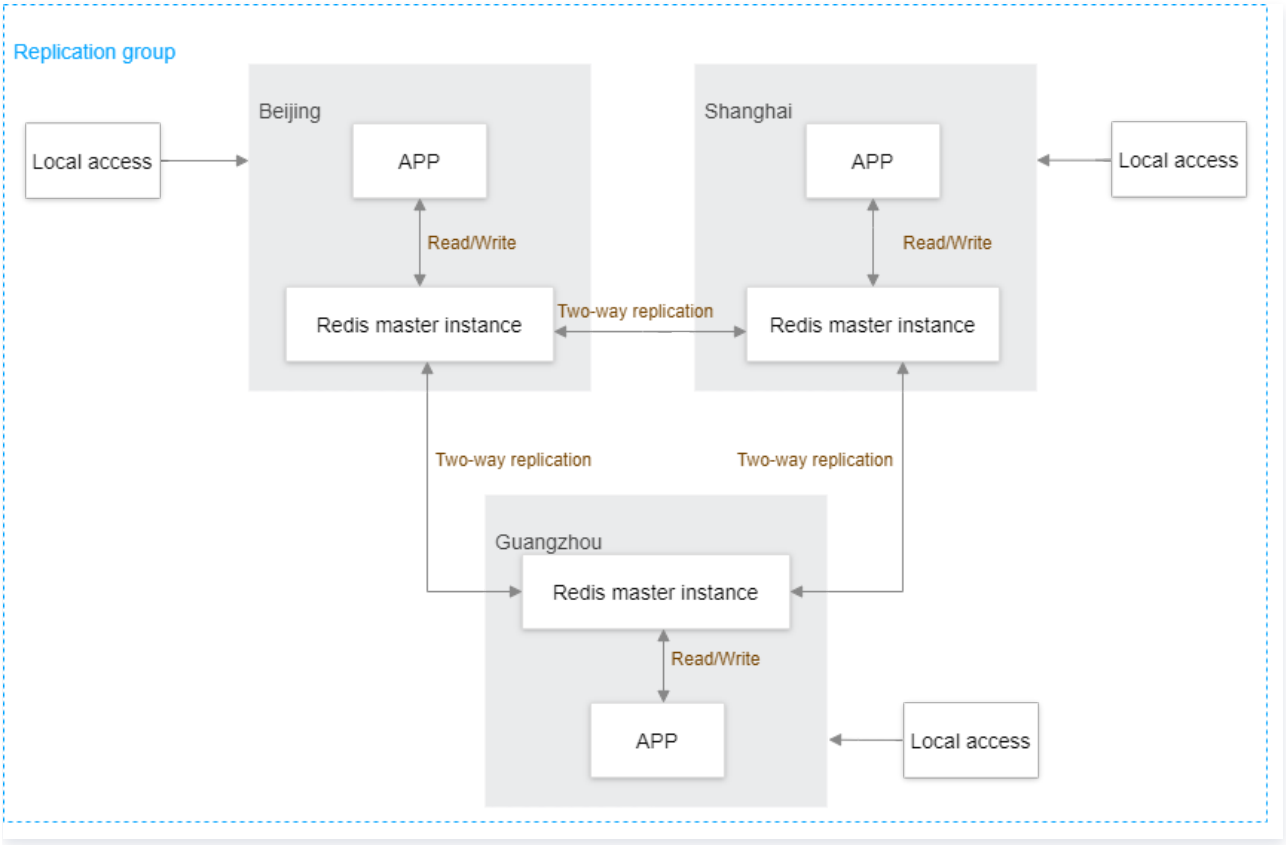
instance determines the data version, and the consistency level is eventual consistency. Users can access data locally, improving response speed, enhancing user experience, increasing data availability, and strengthening data security.



Active-Active Solution

In scenarios such as cross-region user roaming and multi-region data merging, you need to distribute a single dataset across multiple regions, allowing data to be read and updated in any region, or merge data files from multiple regions into one. In this case, the database needs to have multi-region write capabilities. A multi-master architecture can be configured in the global replication group, allowing data to be written to any master instance, and the other master instances in the

group will synchronize the data.



Note:
In Redis, the master instance executes data written by the application and other master instances in the replication group without version detection or write-time checks, following the order in which commands are received. If the same data is updated in different master instances at the same time, global replication may result in data inconsistency, and complete data consistency cannot be guaranteed. Therefore, in multi-master scenarios, it is essential to avoid updating the same data in different master instances simultaneously. Multi-master architecture carries data consistency risks, so please carefully evaluate whether this feature applies to your business scenario.

Limits

Last updated: 2024-11-01 17:09:34

Limits on Region and AZ

Global replication allows data replication between any Tencent Cloud regions, without restricting the deployment location of instances within the replication group. You can replicate data within the same AZ or across multiple AZs. Currently, only the following regions and AZs support global replication, but you cannot adjust the AZ of instances within the replication group.

Regions	Availability Zones
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 global 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 global 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 global 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 global 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 specification, all instances in a global replication group must have the same specification; otherwise, performance or capacity problems may occur.

Limits on the parameter settings

The `maxmemory-policy` parameter of instances in a replication group must be set to `noeviction`.

Limits on Command Sync

- The **FLUSHDB** or **FLUSHALL** commands will be synchronized to all instances within the replication group. Please proceed with caution.
- The **Pub** and **Sub** command families will not be synchronized. To achieve cross-region notification message replication, it is recommended to use the Stream data structure.
- When the **RESTORE** command is synced, if the target instance has the same key, it will not be executed.

Limits on 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: 2024-11-01 17:09:43

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:** Each instance has the ability to automatically switch between the primary and replica nodes in case of a failure, but it will not perform automatic failover between the master and read-only instances.

Release notes

- 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 both single-AZ and multi-AZ instances.

Billing description

When the instance syncs data from other instances in the replication group, no additional fees will be incurred.

Creating Global Replication Group

Preparations

- You have created a TencentDB for Redis® instance. For more information, see Creating TencentDB for Redis® Instance.
- The instance is in the **Running** status.

Instructions

1. Log in to the TencentDB for Redis® console.
2. Choose **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**.

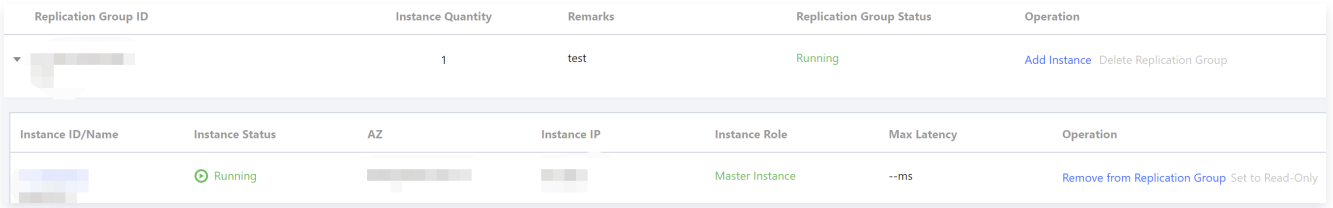
Category	Description	Required	Sample
Name	Name of the replication group to be created. Enter a name as prompted.	Supported	test
Remarks	Brief description of the replication group. You can enter any characters to distinguish the group from others.	Not required	Replication group creation test
Master Instance Region	Select the region of the master instance in the replication group.	Supported	Guangzhou
Select Master	Select the master instance in the replication group. After selecting the master instance, the version, architecture, and	Supported	test-XXX

Instance	memory capacity of the instance will be displayed. Please confirm whether the specifications meet your requirements.		

> **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.



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.

Supports and Limits

- 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.

Preparations

- 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.
> - In an existing replication group, the first instance that is re-added after removing all instances.

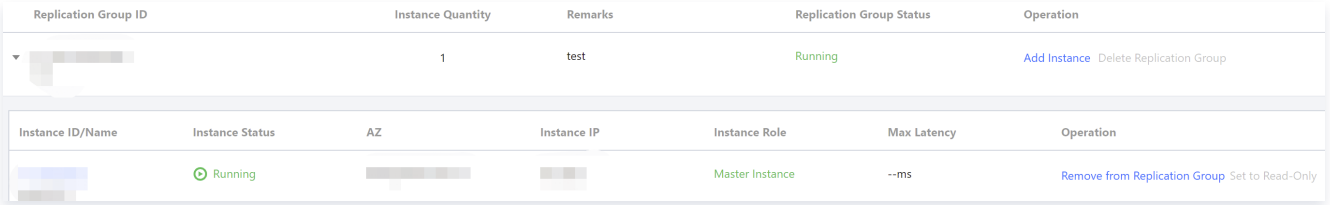
Instructions

1. On the **Redis – Global Replication** page, select the replication group to which you want to add instances from the instance list.
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 opt to add a read-only instance. If you need to add a master instance, please submit a ticket to contact Tencent Cloud engineers for activation.

4. Return to the **Redis - Global Replication** page, and click ▶ before the name of the replication group in the replication group list to show its instance list, where you can view the newly added instance.



After adding an instance, you can perform data synchronization within the replication group. You can add multiple instances for data replication according to your actual needs.

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 cannot 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

Regardless of whether a replication group has one or multiple master instances, when replication is interrupted, the system does not set the master instances as read-only instances or perform other operations. Instead, it automatically resumes the replay of incremental logs after the instances recover. We recommend setting alarms for replication exceptions and configuring the master instances as read-only instances during replication anomalies (such as replication disconnections) to ensure data consistency.

Related APIs

Related APIs	API Meaning
CreateReplicationGroup	Create replication group API
DeleteReplicationInstance	Removes a member from a replication group
AddReplicationInstance	Adds a member to a replication group
DescribeReplicationGroup	Queries a replication group
ChangeMasterInstance	This API is used to switch with master instance in a replication group.
ChangeInstanceRole	This API is used to modify the role of an instance in a replication group.

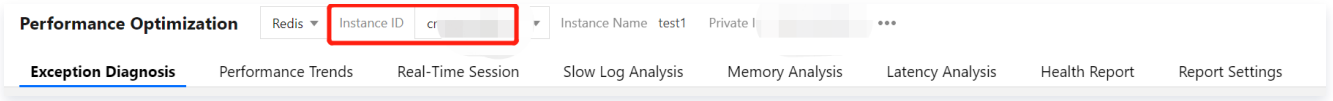
Performance Optimization Overview

Last updated: 2024-11-01 17:10:03

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, choose **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. View and analyze the diagnosis data of the instance.

Monitoring Category	Monitoring Explanation
Exception Diagnosis	Performs real-time performance monitoring and health inspections on the database and gives diagnosis prompts and optimization suggestions for failures.
Performance Trend	Monitors the performance metrics by instance, Redis node, and proxy node.
Real-time Sessions	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.
Health Report	Provides a health score for the current instance based on various monitoring metrics and statistical data.

Related APIs

Related APIs	API Function
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 of an instance

Exception diagnosis

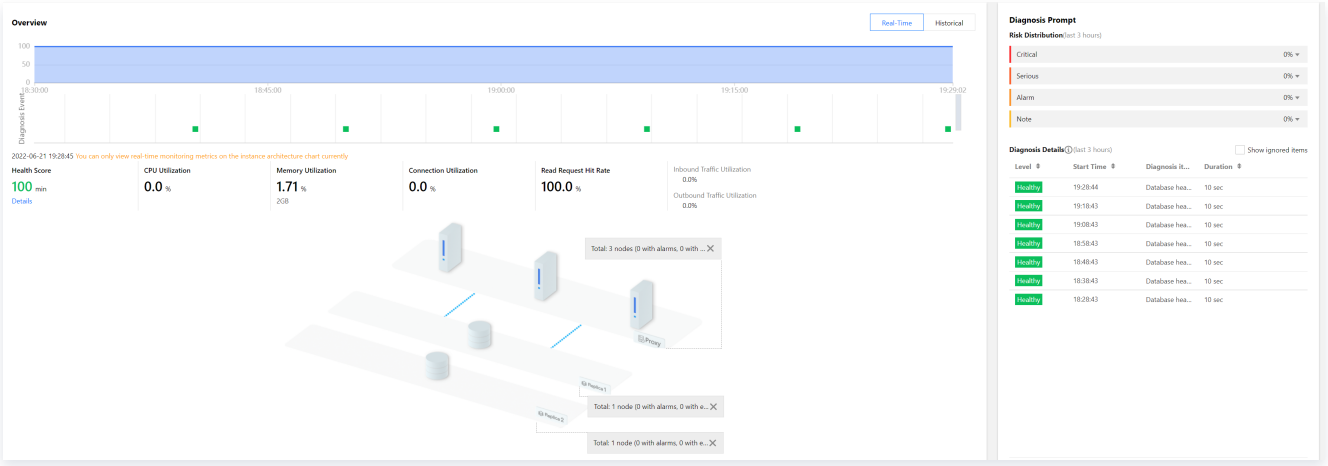
Last updated: 2024-11-01 17:10:14

Feature Overview

Exception diagnosis, based on data collected through intelligent monitoring, provides 7x24-hour real-time database exception analysis. For performance issues of database instances, it utilizes SQL optimization engine, performance analysis engine, and rule engine to conduct comprehensive analysis from dimensions such as configuration, SQL analysis, business logic, and usage rationality. This provides both current and historical health assessment results, including health scores and risk levels, to help operations personnel intuitively, clearly, and quickly grasp the running status of database instances.

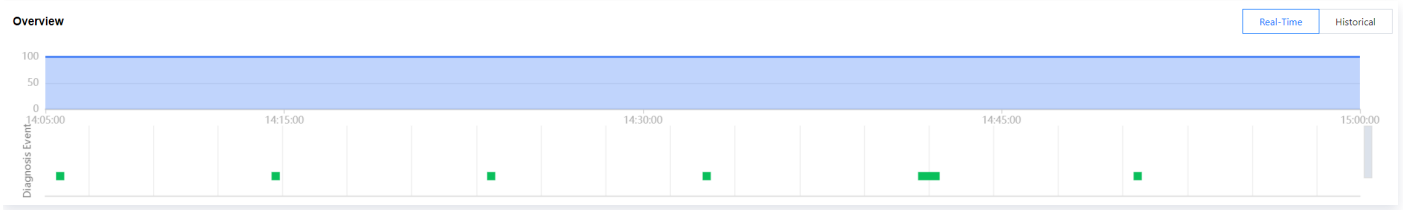
Viewing Performance Optimization

1. Log in to the TencentDB for Redis® console and choose **Performance Optimization** on the left sidebar.
2. On the **Performance Optimization** page of **DBbrain**, choose **Exception Diagnosis** to view the exception diagnosis details.



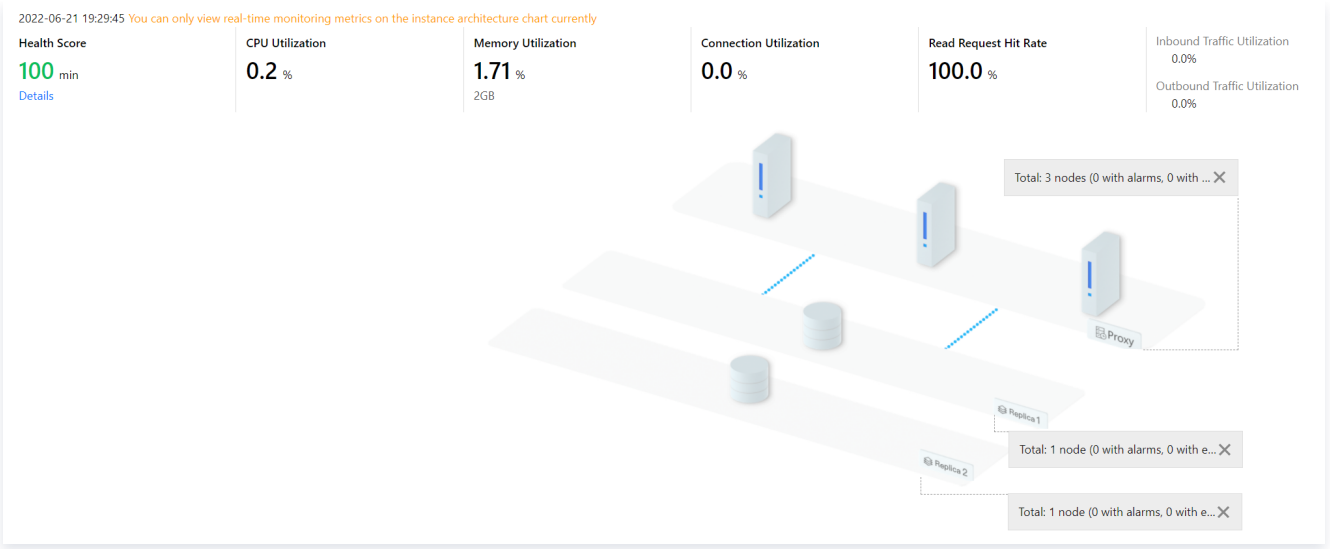
Overview

In the **Overview** section, the default view displays the trend of the Redis database instance's health score over the past hour, as well as the risk level of the database health inspection diagnosis. Click **History** in the top right corner to select a time range and view the health score and inspection diagnosis results for that historical period.



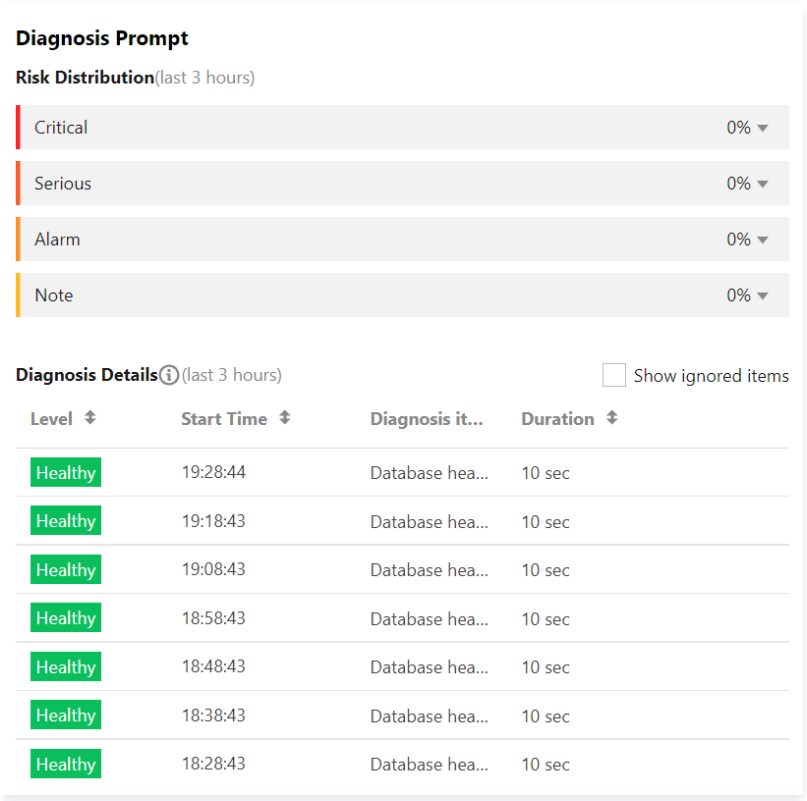
Below the **Overview** section, the current health score of the database and the alarms and exceptions for each instance node are displayed.

- It displays the current health score of the database instance and the statistical data of current resource monitoring indicators, including CPU usage, memory usage, connection usage, input/output traffic usage, and read request hit rate. This helps you quickly grasp the usage of database resources. Click **Details** below the health score to navigate to the **Health Report** tab, where you can view detailed information about the health score. For more information, see Health Report Management.
- On the other hand, the system architecture diagram of the database is displayed, allowing you to intuitively see the Proxy nodes, primary nodes, and replica nodes of the database architecture. You can also directly view the number of alarms and exception events for each node. Moreover, by hovering your mouse over a node in the architecture diagram, you can view the monitoring data of its key metrics. For monitoring metrics, please refer to Performance Trends.



Diagnosis Prompt

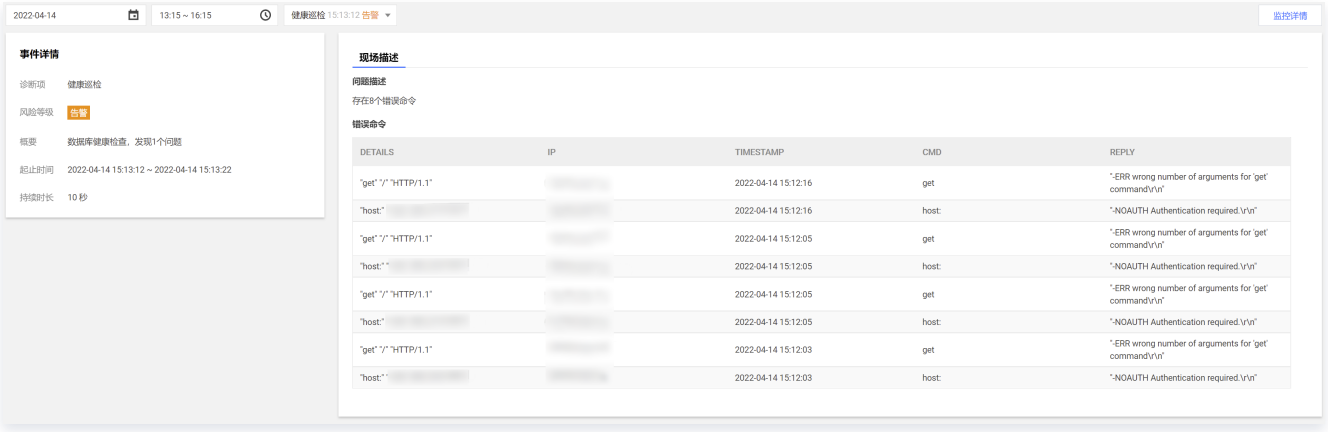
In the **Diagnosis Prompt** section, the diagnosis data of the database health inspection in the last 3 hours is displayed.



- **Risk Distribution**: Displays the risk level distribution proportion of the database health inspection diagnosis in the last 3 hours. Risk levels in descending order of severity, include: Fatal, Serious, Warning, and Notice. If all risk level proportions are 0%, the database status is **Healthy**.
- **Diagnosis Details**: Displays the details of the database risk level collected every 10 minutes. You can hover over any exception alarm to view, ignore, or cancel ignoring alarms of the same type.

Category	Note
Grade	The risk levels of diagnosis, in descending order of severity, include: Fatal, Critical, Warning, Notice, and Healthy.
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 see the detailed information about the exceptions discovered during the health inspection, as shown below.
Ignore	Click on **Ignore**, and in the confirmation dialog box, click on **Confirm**. Alerts generated from the same root cause will be ignored.
Ignore Cancellation	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 **Ignore Cancellation**, and all exception alarms triggered with the same cause will not be ignored any more.



Performance Trends

Last updated: 2024-11-01 17:10:32

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.

Monitored metrics

Supported monitoring metrics are displayed in three dimensions: instance, Redis node, and proxy node.

Instance

Category	Meaning	Metric	Unit	Description
CPU	CPU utilization	cpu_util	%	Avg CPU usage
	Max CPU utilization of a node	cpu_max_util	%	The maximum CPU utilization of a node (shard or replica) in an instance
Memory Info	MEM Usage	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	mem_util	%	The ratio of the memory actually used to the total memory requested
	Max memory utilization of a node	mem_max_util	%	The maximum memory utilization of a node (shard or replica) in an instance
Key information	Keys	keys	Connections	Total number of keys stored in an instance (first-level keys)
	Expired keys	expired	Connections	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	Connections	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	Connections	Number of TCP connections to an instance
	Inbound Traffic	in_flow	Mb/s	Private Inbound Traffic
	TrafficOut	out_flow	Mb/s	Private Outbound Traffic
Network utilization	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
Requests	Total requests	commands	Queries/second	QPS, that is, the number of command executions per second
	Read Requests	cmd_read	Queries/second	The number of read command executions per second
	Write Request	cmd_write	Queries/second	The number of write command executions per second
	Other Requests	cmd_other	Queries/second	The number of command (excluding write and read commands) executions per second.
	Big value requests	cmd_big_value	Queries/second	The number of executions of commands larger than 32 KB per second
	Key requests	cmd_key_count	Queries/second	The number of keys requested per second
	Mget requests	cmd_cmget	Queries/second	The number of requests made through MGET per second
Response	Slow query	cmd_slow	Threads	The number of command executions with a latency greater than the `slowlog-log-slower-than` configuration
	Read request hits	cmd_hits	Threads	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	Threads	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 Errors	Execution Errors	cmd_err	Threads	Number of command execution errors, such as when a command does not exist or a parameter is incorrect
Latency	Average execution latency	latency_avg	ms	Average execution latency from Proxy to Redis server
	Max execution latency	latency_max	ms	Maximum execution latency from Proxy to 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 read command categories, see Command Categories.
	Average write latency	latency_write	ms	The average execution latency of write commands between the proxy and the Redis server. For write command categories, see Command Categories.
	Average latency of other commands	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server

Redis node

Category	Meaning	Metric	Unit	Description
CPU	CPU utilization	cpu_util	%	Avg CPU usage
Network usage	Number of connections	connections	Connections	The number of connections from Proxy to a node
	Connection utilization	connections_util	%	The connection usage of a node
Memory Info	MEM Usage	mem_used	MB	Memory capacity actually used, including data and cache
	Memory utilization	mem_util	%	The ratio of the memory actually used to the total memory requested
Key information	Keys	keys	Connections	Total number of keys stored in an instance (first-level keys)
	Expired keys	expired	Connections	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	Connections	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 secondary node and the primary node
Requests	Total requests	commands	Queries/second	QPS, that is, the number of command executions per second
	Read Requests	cmd_read	Queries/second	The number of read command executions. For read command categories, see Command types.
	Write Request	cmd_write	Queries/second	The number of write command executions. For write command categories, see Command types.
	Other Requests	cmd_other	Queries/second	Executions of commands other than read and write commands
Response	Slow query	cmd_slow	Threads	The number of command executions with a latency greater than the

				`slowlog-log-slower-than` configuration
	Read request hits	cmd_hits	Threads	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	Threads	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

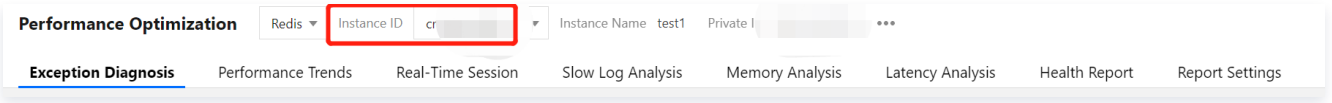
Category	Meaning	Metric	Unit	Description
CPU	CPU utilization	cpu_util	%	Proxy CPU utilization
Requests	Total requests	proxy_commands	Queries/second	The number of Proxy command executions per second
	Key requests	cmd_key_count	keys/second	The number of keys accessed by a command per second
	Mget requests	cmd_mget	Queries/second	The number of Mget command executions per second
	Execution Errors	cmd_err	Queries/second	The number of Proxy command execution errors per second, for example, when a command does not exist, parameters are incorrect, etc.
	Big value requests	cmd_big_value	Queries/second	The number of executions of commands larger than 32 KB per second
Traffic	Inbound Traffic	in_flow	Mb/s	Private Inbound Traffic
	TrafficOut	out_flow	Mb/s	Private Outbound Traffic
Network usage	Number of connections	connections	Connections	Number of TCP connections to an instance
	Connections per sec	client_connections_received_per_second	Connections	The number of TCP connections established per second
	Disconnections per sec	client_connections_closed_per_second	Connections	The number of TCP connections closed per second
	Abnormal disconnections per second	client_connections_aborted_per_second	Connections	The number of TCP connections aborted per second
Network utilization	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	Threads	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	Threads	The number of times outbound traffic triggers a traffic limit
Latency	Average execution latency	latency_avg	ms	Average execution latency from Proxy to Redis server
	Max execution latency	latency_max	ms	Maximum execution latency from Proxy to 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 read command categories, see Command Categories.
	Average write latency	latency_write	ms	The average execution latency of write commands between the proxy and the Redis server. For write command categories, see Command Categories.
	Average latency of other commands	latency_other	ms	The average execution latency of commands other than read and write commands from Proxy to Redis server

Viewing monitoring data

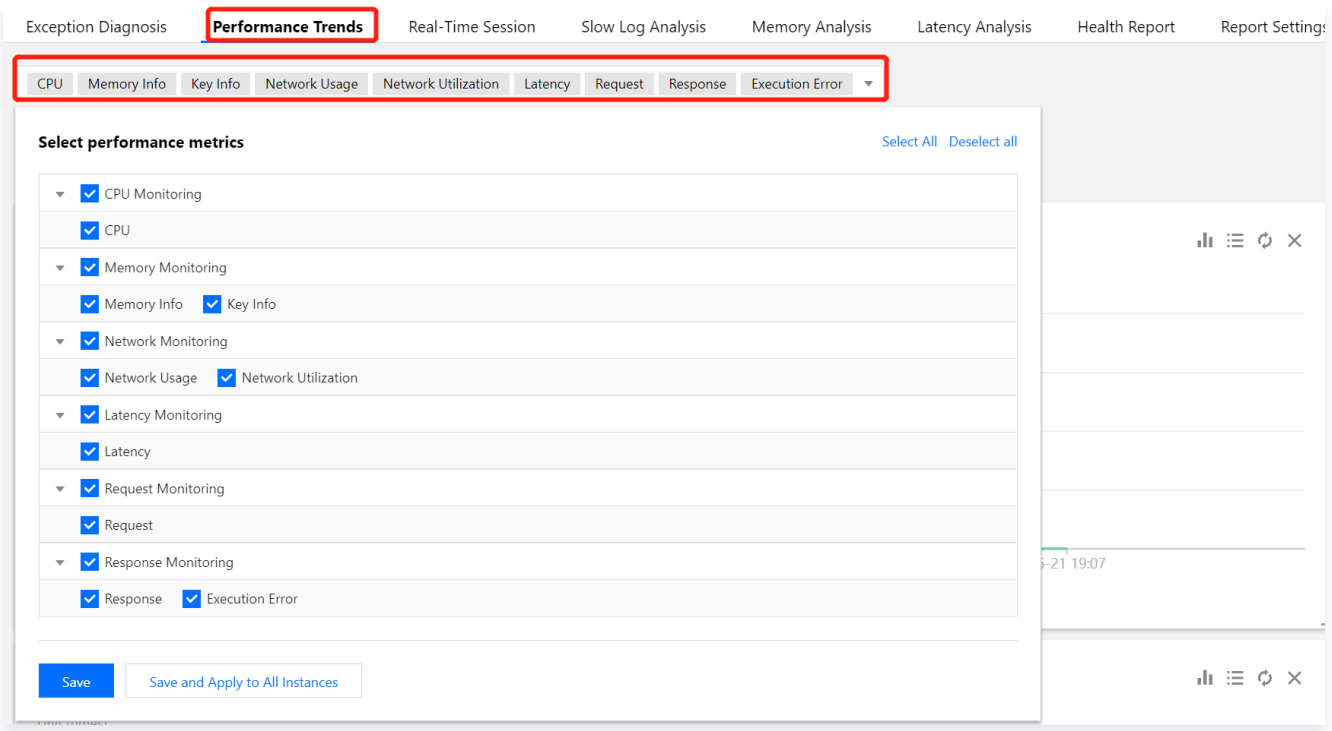
Step 1. Select monitoring metrics

1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



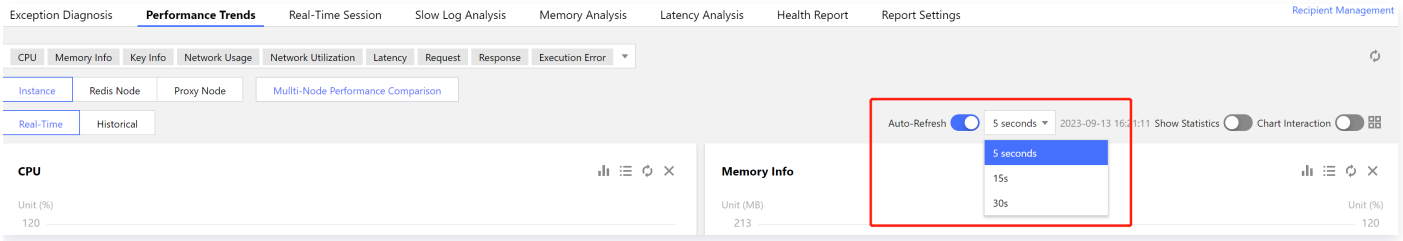
4. Click the **Performance Trends** tab, select the desired performance metrics in the drop-down list of metric categories, and click **Save**.

   To apply the selected performance metrics to all TencentDB for Redis® instances under the Tencent Cloud account, click **Save and Apply to All Instances**, as shown below.



Step 2. Set the collection granularity

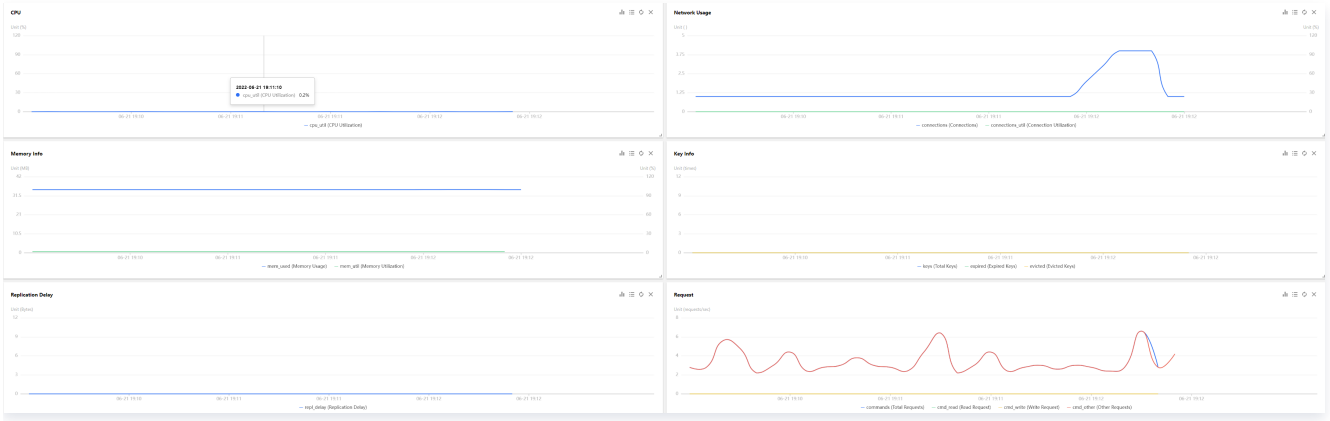
In the drop-down list to the right of **Auto Refresh** at the top right corner of the **Performance Trends** tab, select the granularity for collecting monitoring data. Options include 5 seconds, 15 seconds, and 30 seconds, as shown below.



Step 3. View the change trends of the monitoring metrics

Viewing monitoring metrics in different dimensions

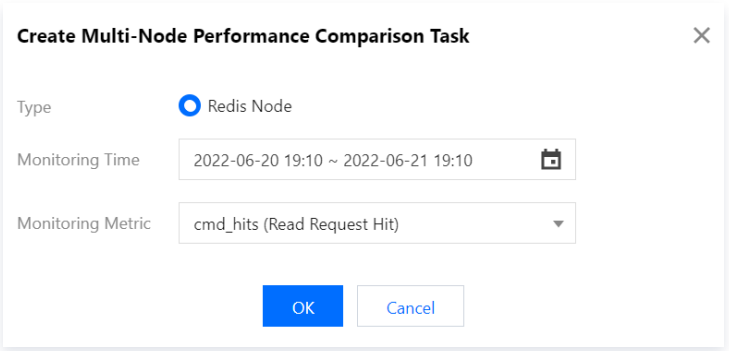
Under the **Performance Trends** tab, you can view monitoring metrics data for instance, Redis node, and proxy node dimensions according to your business operations and maintenance requirements.



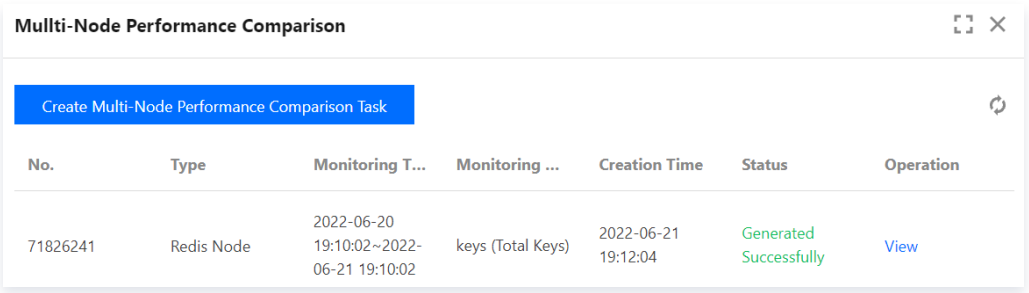
Comparing the performance metrics of multiple nodes

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 then click **OK**.



4. In the **Multi-Node Performance Comparison** panel's task list, wait for the **Status** to show **Generation Successful**.

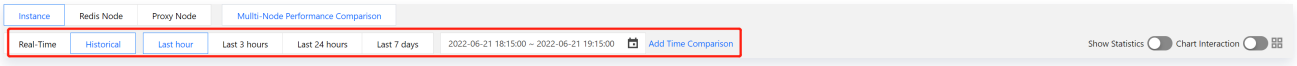


5. Click **View** in the **Operation** column to view the comparison data of all Redis nodes. The connections metric is used as an example as shown below:

Switching Between Real-Time/Historical Views

On the **Performance Trends** tab, the real-time monitoring data is displayed by default.

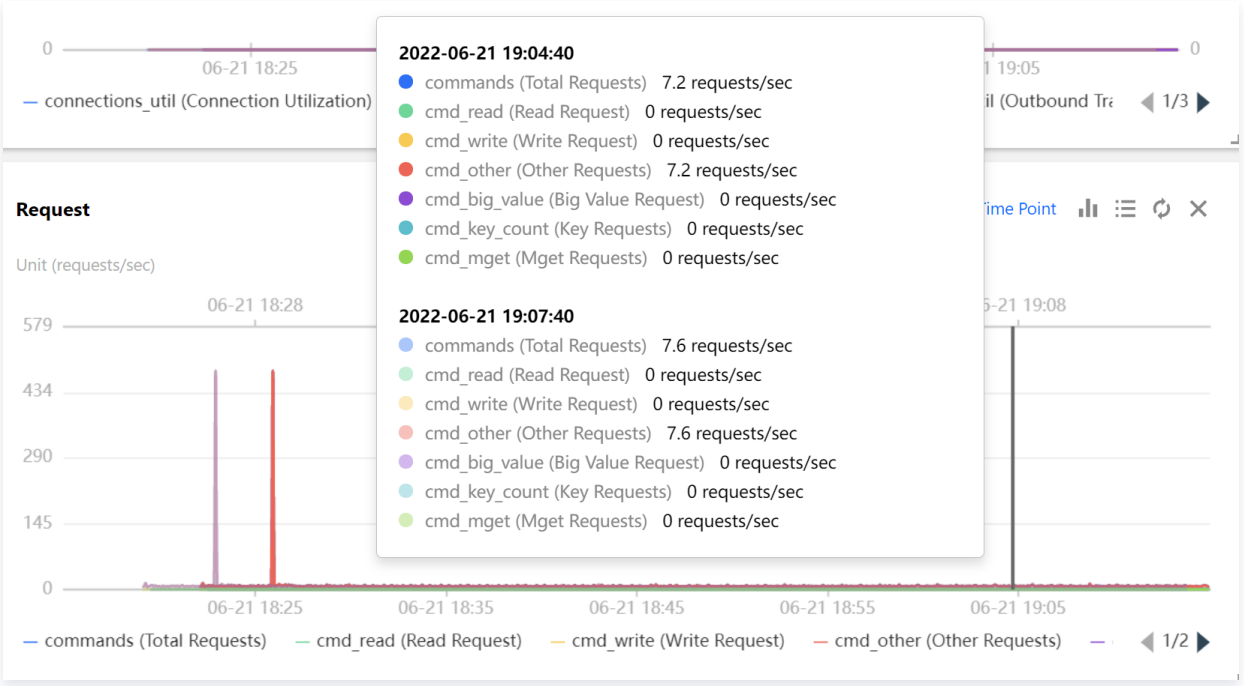
- In routine Ops monitoring, database instance metrics can be monitored in real time.
- When you need to locate exceptions, click **Historical** to analyze the monitoring data in a past time period.
  - The monitoring data in the last 1 hour, 3 hours, and 7 days can be viewed.
  - Click to view monitoring data for any time period within the last 30 days.



Comparing performance metrics in different time periods

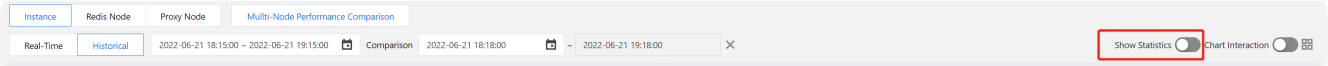
1. On the **Performance Trends** tab, click **History**, and then click **Add Time Comparison**.
2. In the time selection box, select two time periods for comparison.

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.

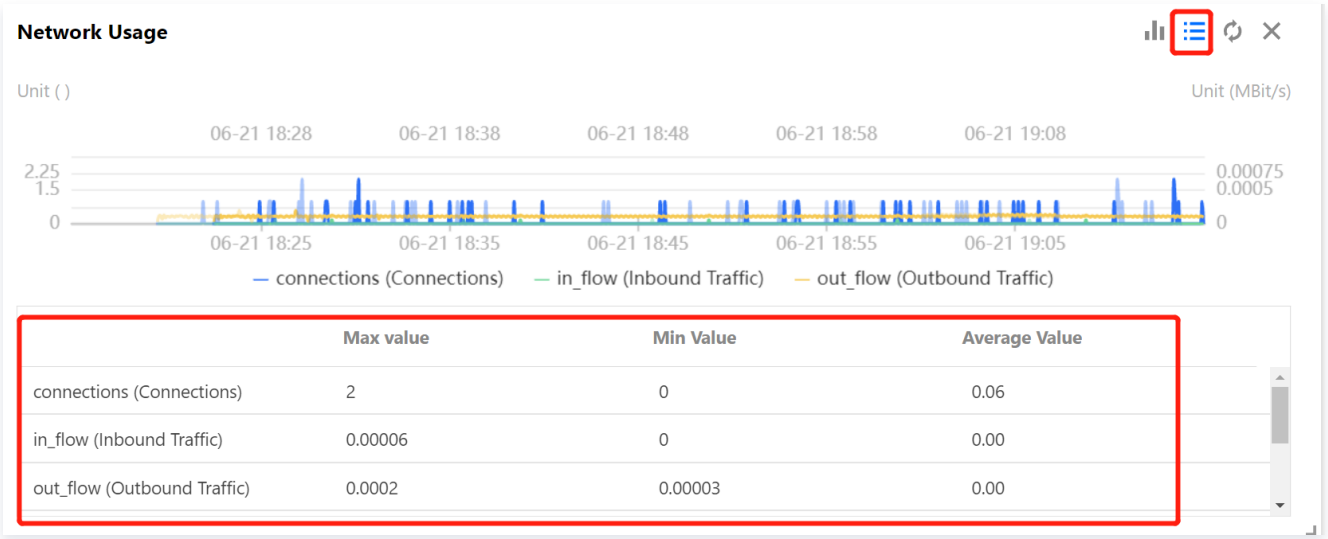


Displaying monitoring metric data in chart

- Click next to as shown below to display the max, min, and average values of each monitoring metric in a table.



- 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:

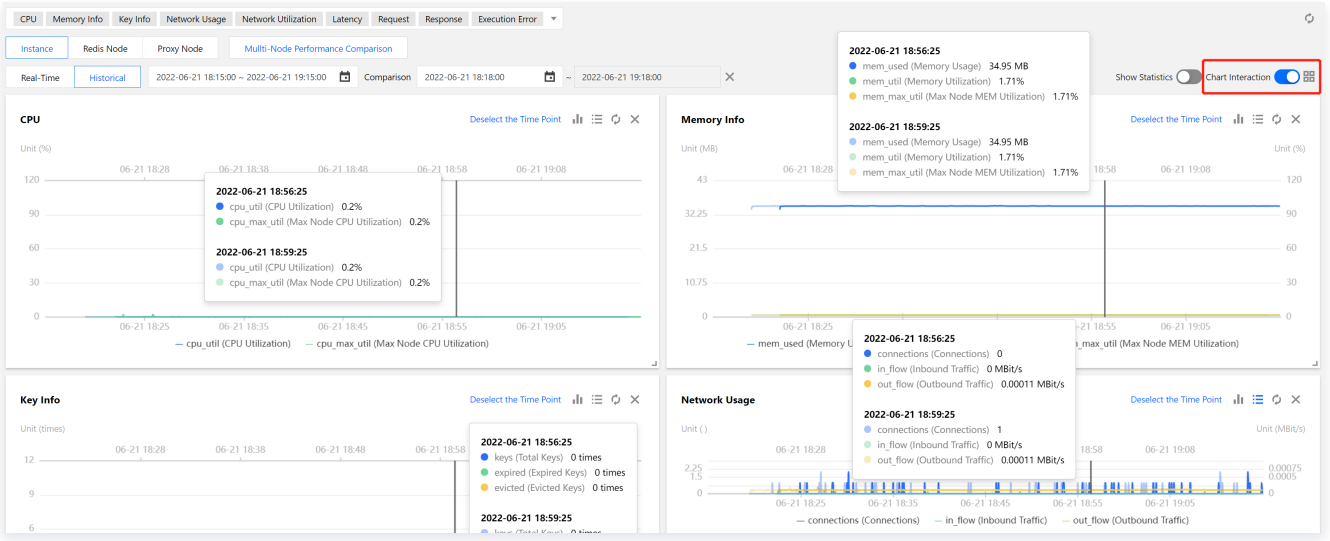


Viewing monitoring data through chart interaction

The **Chart Interaction** feature is suitable for analyzing the data of a monitoring view and its associated monitoring views.

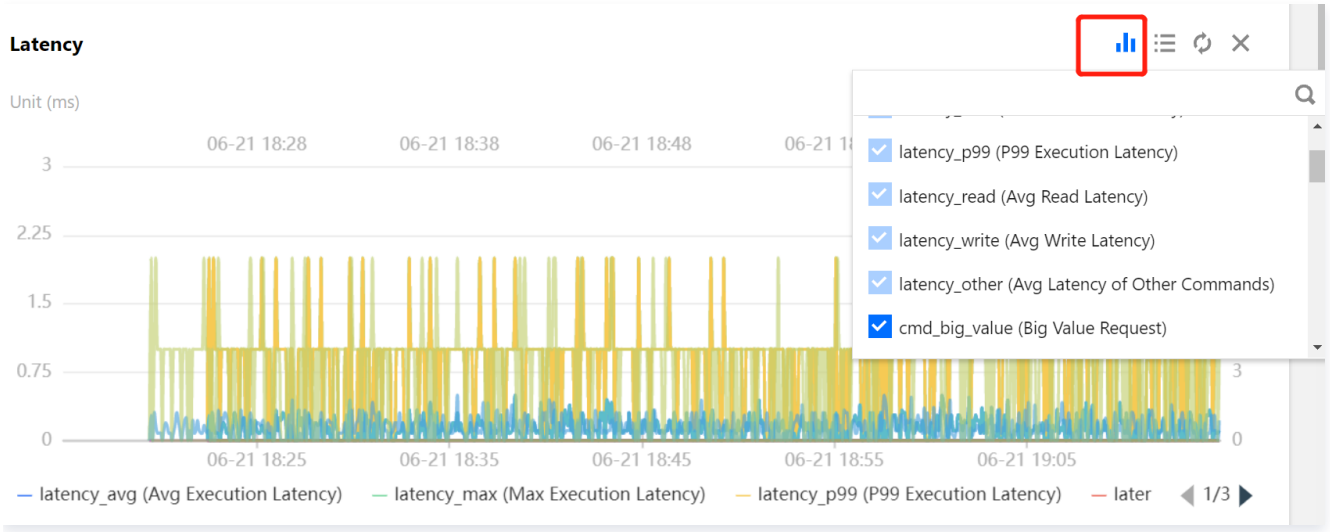
1. In the top-right corner of the **Performance Trends** tab, click the icon next to **Chart Linkage**.

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. Click **Undo Pin** in the top-right corner of the monitoring view to unpin.



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

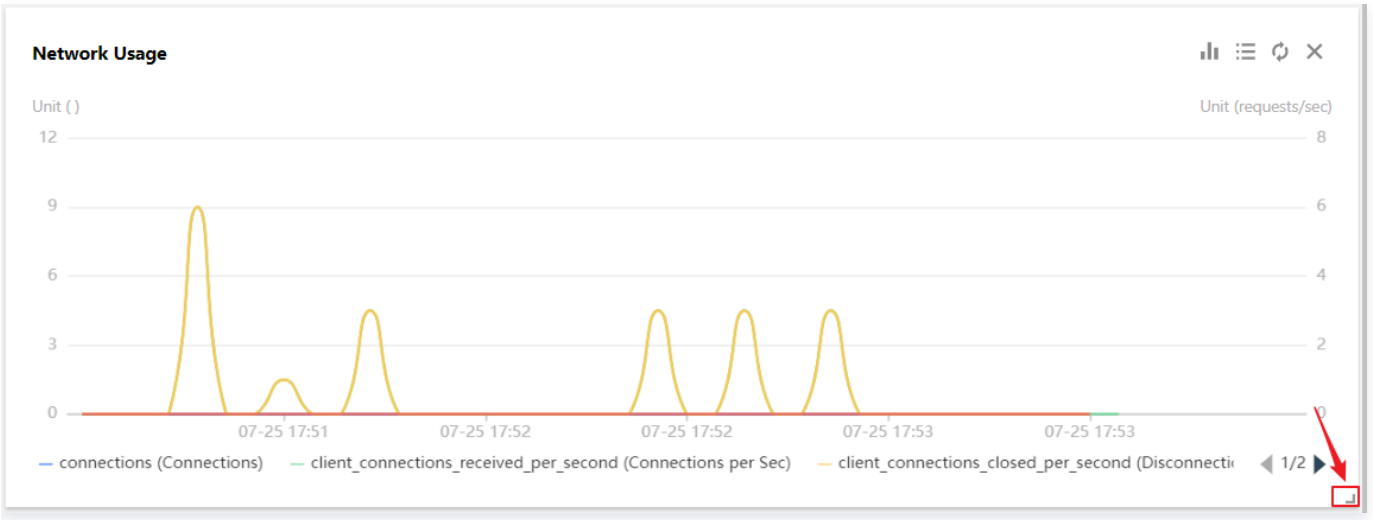
Click on the right side of the chart linkage in the top-right corner to switch between one-column and two-column display modes. Refer to the following image for the one-column mode.

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.

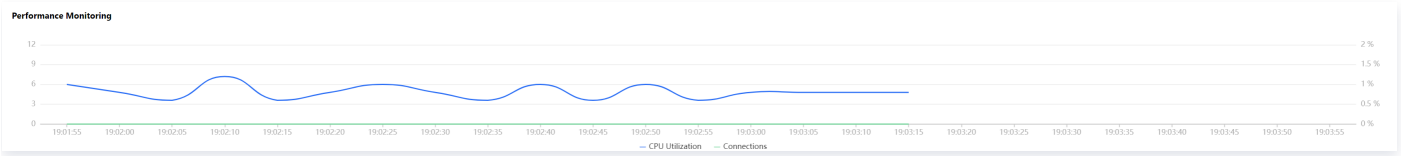


Real-Time Sessions

Last updated: 2024-11-01 17:10:41

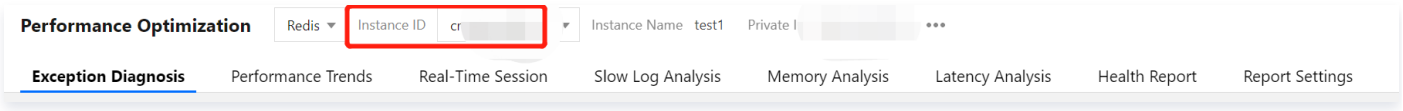
Feature Overview

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.

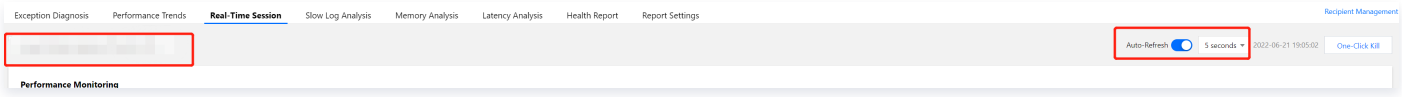


Viewing Real-Time Session Statistics

1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.

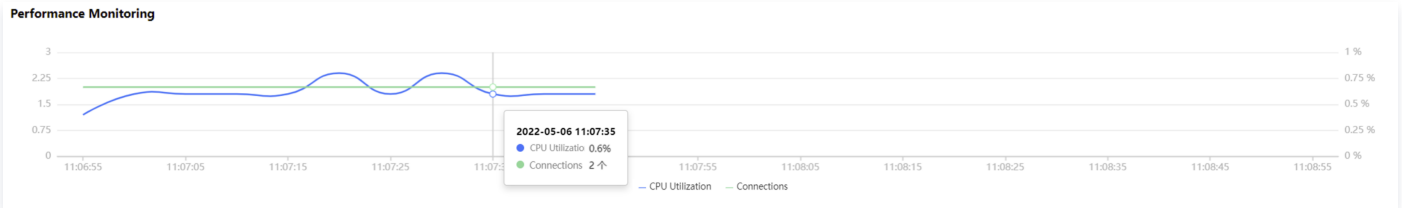


4. Click the **Real-Time Session** tab. In the drop-down list at the top-left corner of the **Performance Monitoring** view, you can select the **Proxy ID** to be analyzed based on the trend chart of **CPU utilization** or **connection quantity**.
In the drop-down list next to **Auto-Refresh** at the top-right corner of the **Performance Monitoring** view, select the collection granularity of the monitoring data. The default is **5s**, and 5s, 15s, and 30s are supported.

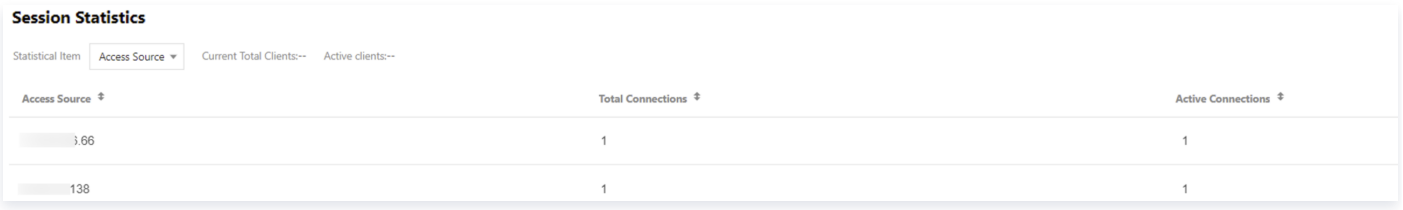


5. View the real-time session details.

- 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: 2024-11-01 17:10:52

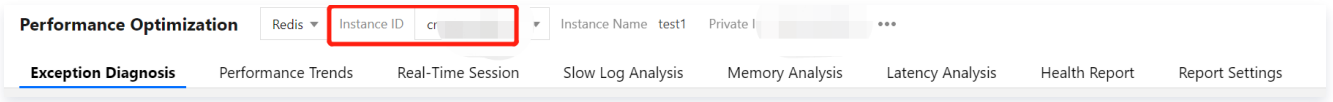
Feature Overview

A slow query is defined as a query statement that exceeds the specified duration, and the corresponding statement is referred to as a slow query statement. Slow log analysis primarily focuses on the statistical analysis of slow query counts for both **instance** and **proxy** dimensions within the database, providing expert-level optimization recommendations to assist in enhancing database performance.

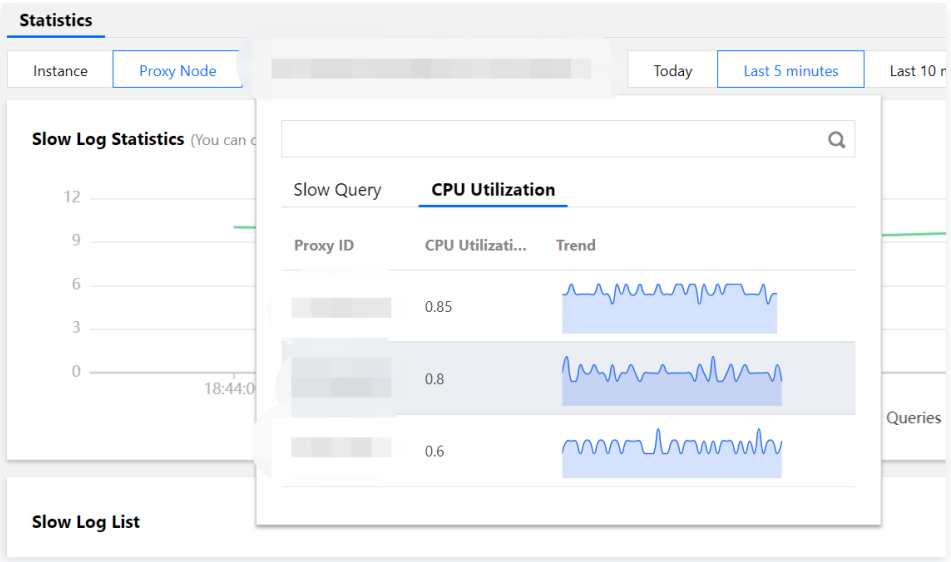
- In the instance (Redis database instance) dimension, you can view the CPU utilization, number of slow logs, consumed time statistics by log segment, and information of the entire slow log list.
- In the proxy (middleware cluster node) dimension, you can view the proxy's slow log statistics, consumed time statistics by segment, and details of the slow log list.

Viewing Slow Log Analysis Data

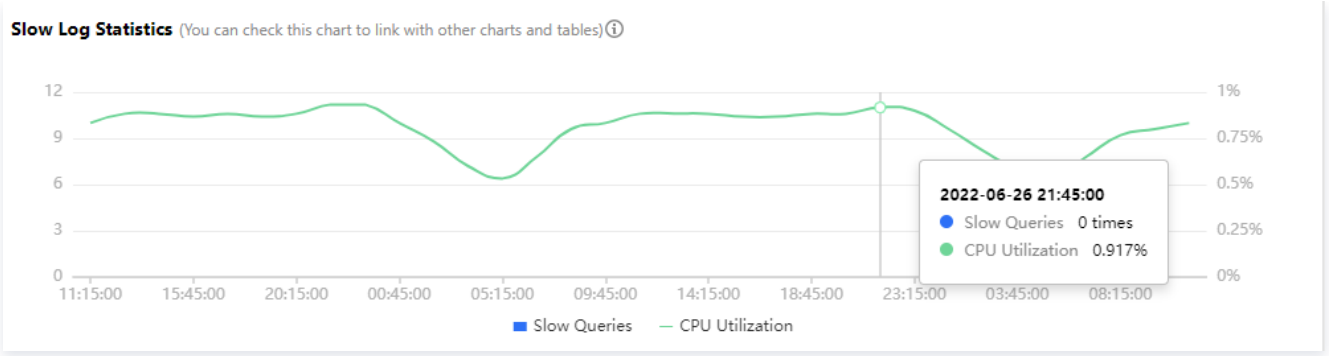
1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



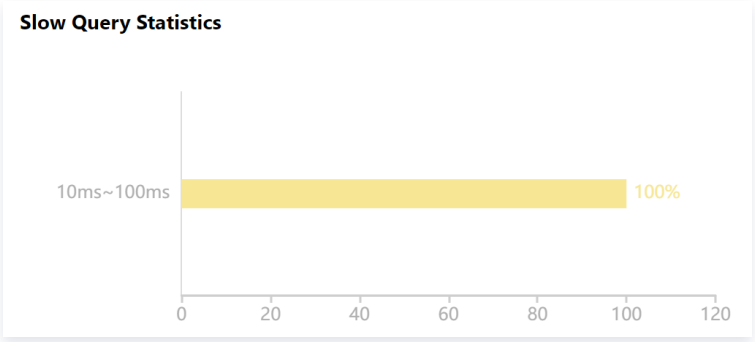
4. Click the **Slow Log Analysis** tab. Then, select the dimension for viewing slow logs and set the query time period in the **Statistics** section.
   - Click **Instance** to view the instance's slow log statistics trend chart.
   - Click **Proxy Node** and select the desired Proxy ID from the dropdown list that follows. You can choose the **Proxy ID** to analyze based on the trend chart of **CPU utilization** or the trend chart of **slow query** count changes.
   - In the time box, click to select a time period of up to 4 days to view slow logs.



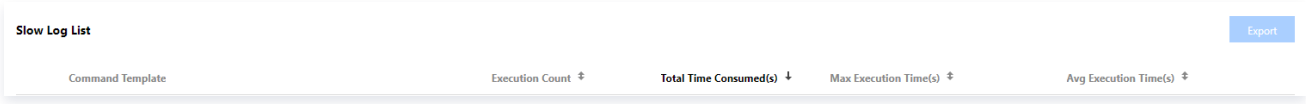
5. View the slow log statistics change trend, slow query statistics, and slow log list.
   - **Slow Log Statistics Change Trend**
     Slow Log Statistics focuses on the number of slow queries and CPU utilization, enabling rapid identification of CPU usage when the number of slow queries remains high within the selected time period. This helps prevent computer lags or unresponsiveness caused by excessive CPU utilization due to a large number of slow queries.



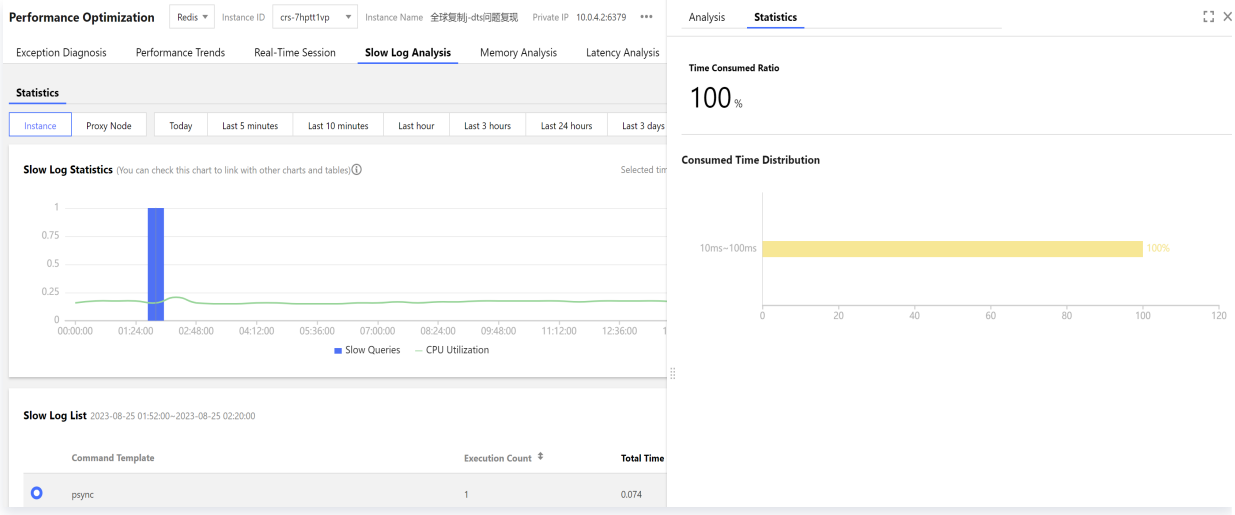
- **Slow Query Duration Statistics**: The slow query duration statistics display the proportion of the proxy node's slow queries by duration range, where the vertical axis represents the duration range, and the horizontal axis represents the distribution proportion. As illustrated in the following diagram, 100% of slow queries fall within the duration range of 10–15 seconds.



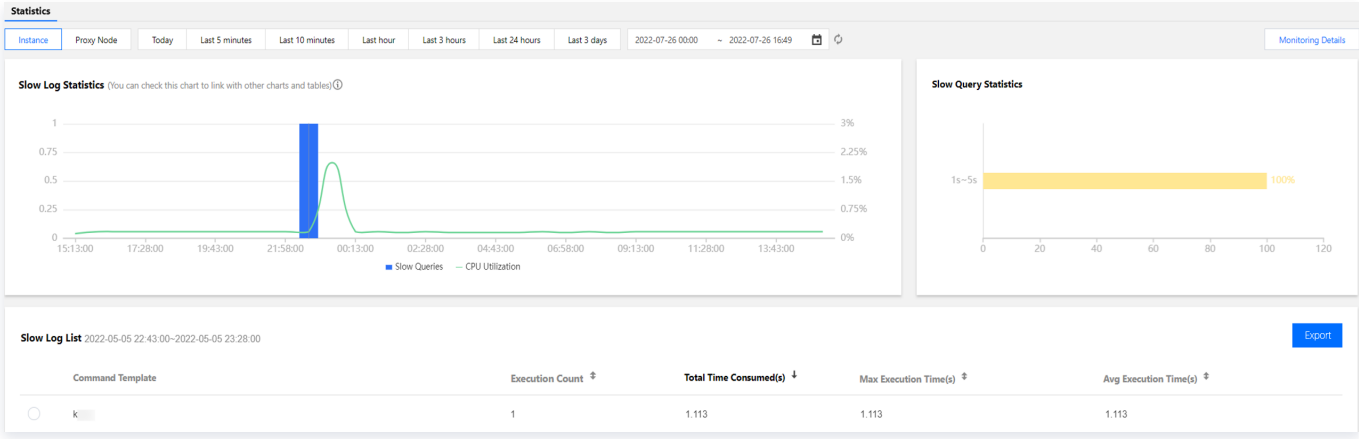
- **Slow Log List**
  - The slow log list displays the execution counts and durations of slow query commands. Click **Export** to download the data for local viewing and analysis.



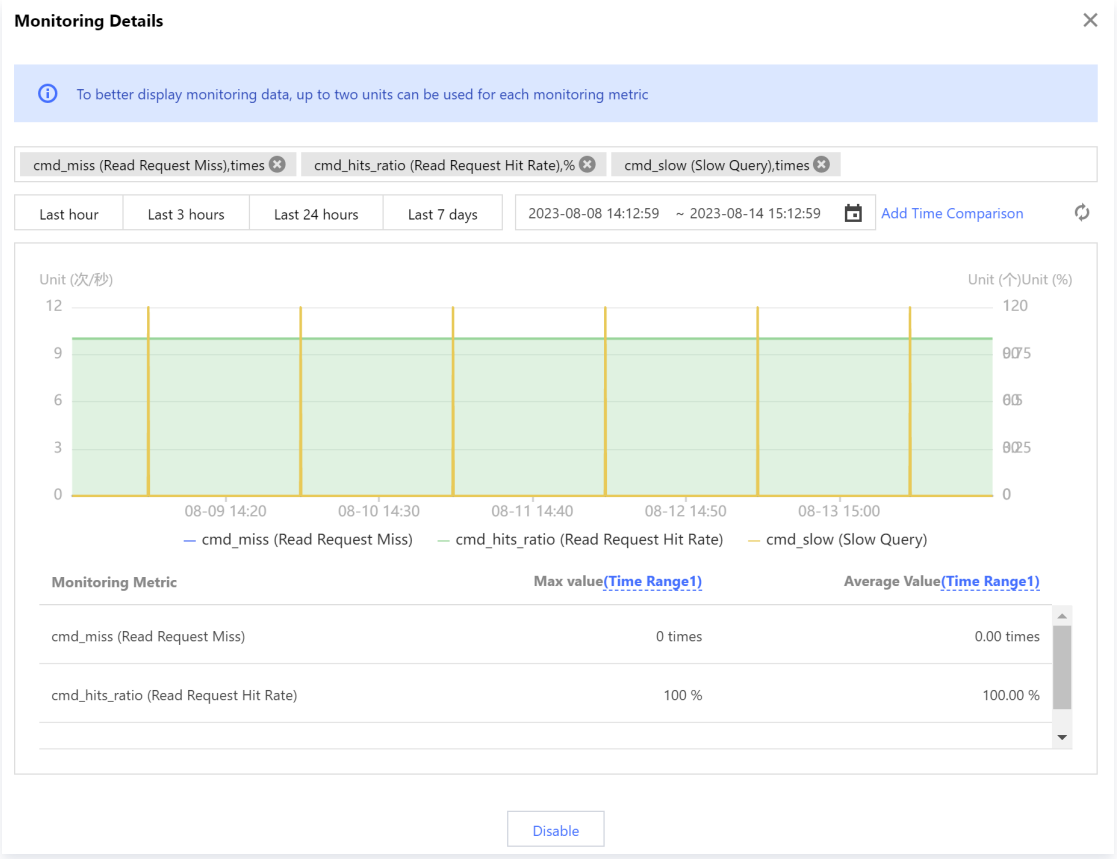
  - Click a command template to display the specific analysis, optimization suggestion, and statistics on the right.
    - The **Analysis** tab displays the command template, command sample, optimization suggestion, and description.
    - The **Statistics** tab displays the execution duration distribution 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

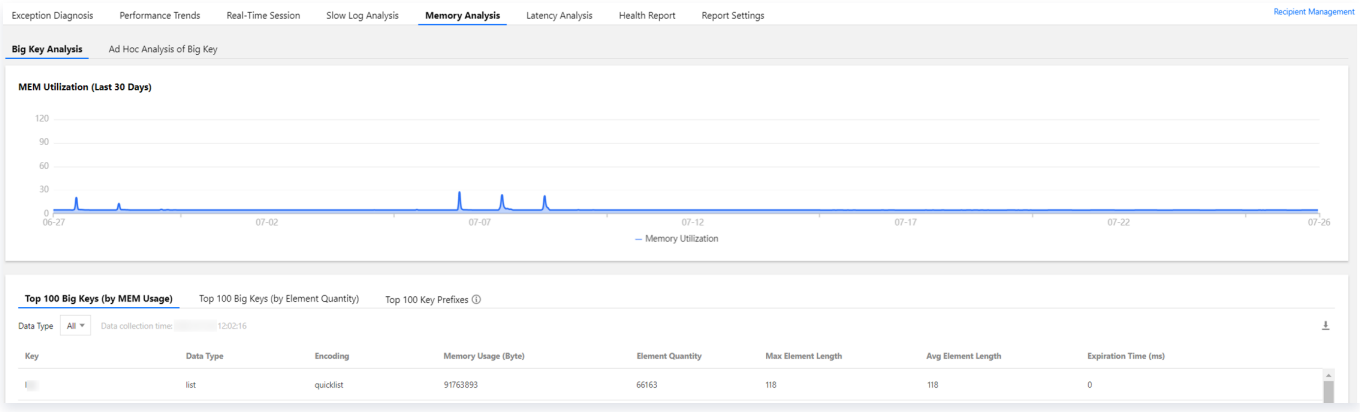
Last updated: 2024-11-01 17:11:00

Feature 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.

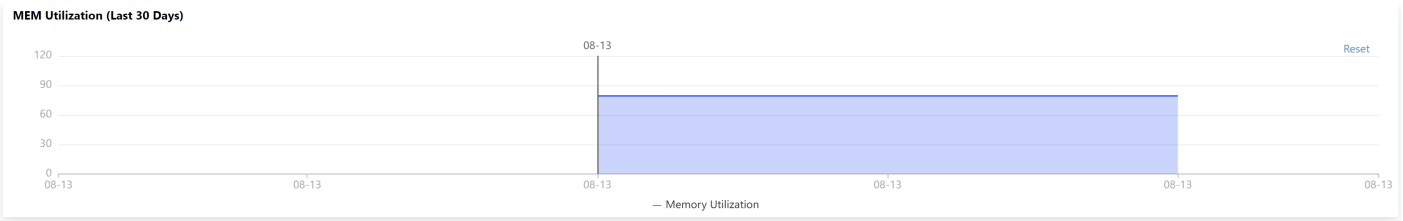
Big Key Analysis Data Interpretation

1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **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. Choose 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 within that time range.



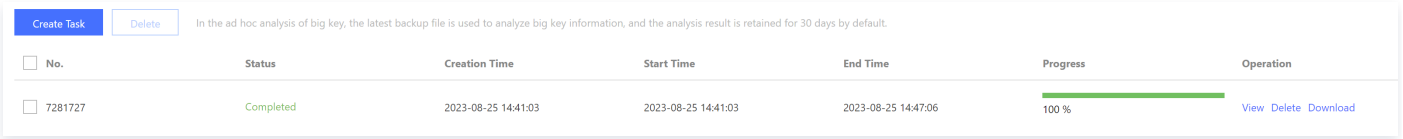
   - **Top 100 Big Keys**
     Select a data storage type in the **Data Type** drop-down list to view the information of the top 100 big keys, including memory usage, element quantity, max element length, average element length, and expiration time.
     - **Top 100 Big Keys (by Memory Usage)**: Shows the top 100 big Keys ranked by memory usage in descending order.
     - **Top 100 Big Keys (by Element Quantity)**: Shows the top 100 big Keys ranked by the number of elements, from the most to the least.
     - **Top 100 Key Prefixes**: Top 100 big keys by prefix.

- **Quickly Locate Big Keys**
  In the memory utilization monitoring view, the change trend of the instance's memory utilization over the past 30 days is displayed by default. If you notice a day with high memory usage, click the day on the date axis to fix the time column. Simultaneously, the information of big Keys for that day is displayed in the list of the top 100 big Keys below, allowing you to quickly identify big Key data during that time.



Creating Ad Hoc Analysis of Big Key

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 latest backup of the database for automatic analysis. You can view the analysis progress through 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 **Instant Big Key Analysis Task Result** panel on the right.

Latency Analysis

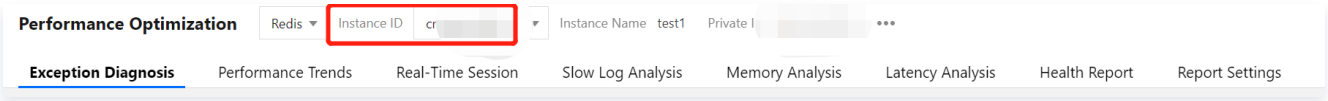
Last updated: 2024-11-01 17:11:09

Feature Overview

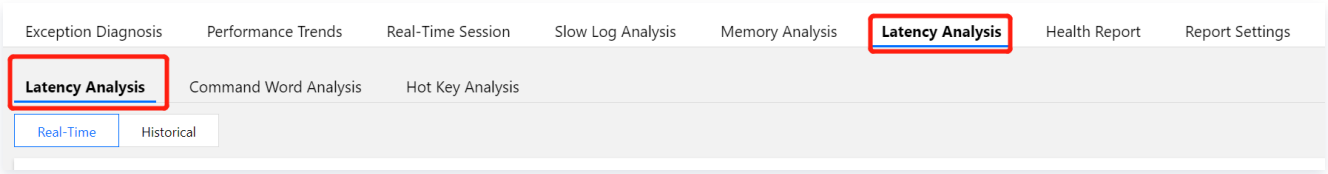
DBbrain's latency analysis feature collects the execution latency statistics of all database requests and monitors the latency accurate down to the millisecond. This helps you troubleshoot TencentDB for Redis® instance failures and performance problems.

Viewing Latency Analysis

1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.

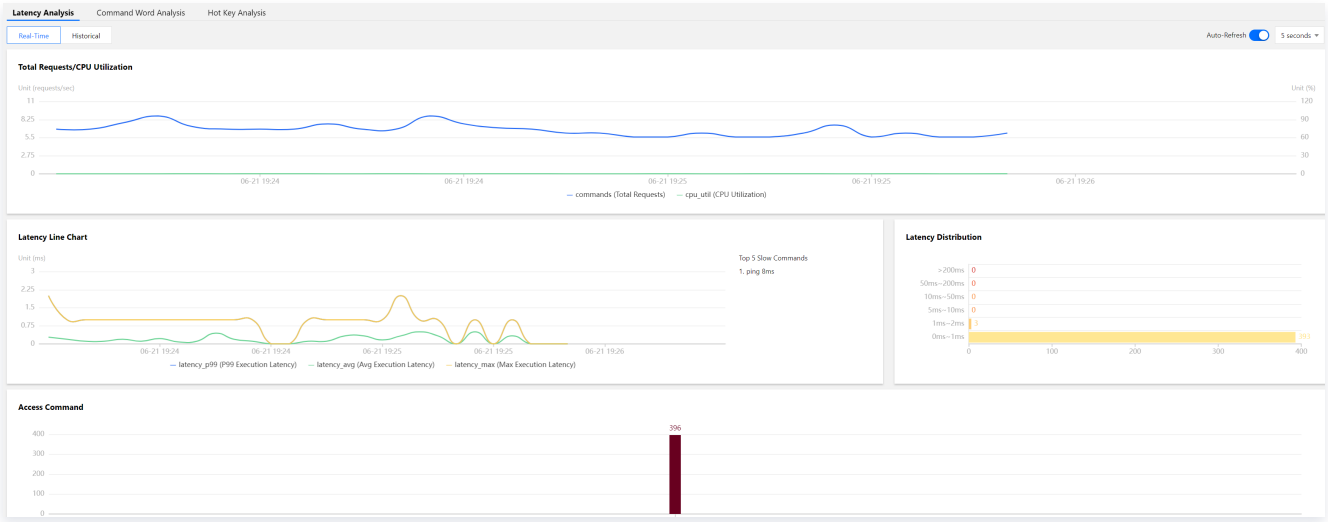


4. Select the **Latency Analysis** tab, and in the dropdown list next to **Auto Refresh** at the top right, set the collection granularity. Options include 5 seconds, 15 seconds, and 30 seconds.

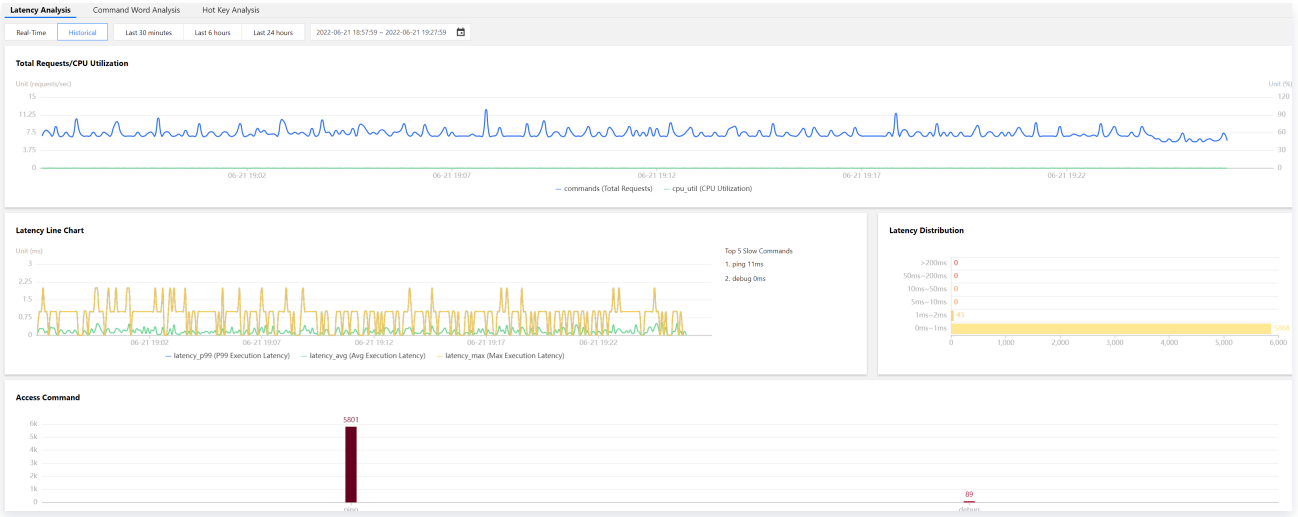


5. View the monitoring data of latency analysis.

- **Real-time Monitoring**
  By default, the change in real-time monitoring data is displayed as a curve.



- **Historical data**
  Click **Historical** to view the monitoring data in a past time period.
  - Select **Last 30 minutes**, **Last 6 hours**, or **Last 24 hours** to view the monitoring data in the corresponding time period.
  - 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, which can help you quickly identify the CPU utilization value when the number of requests is high.

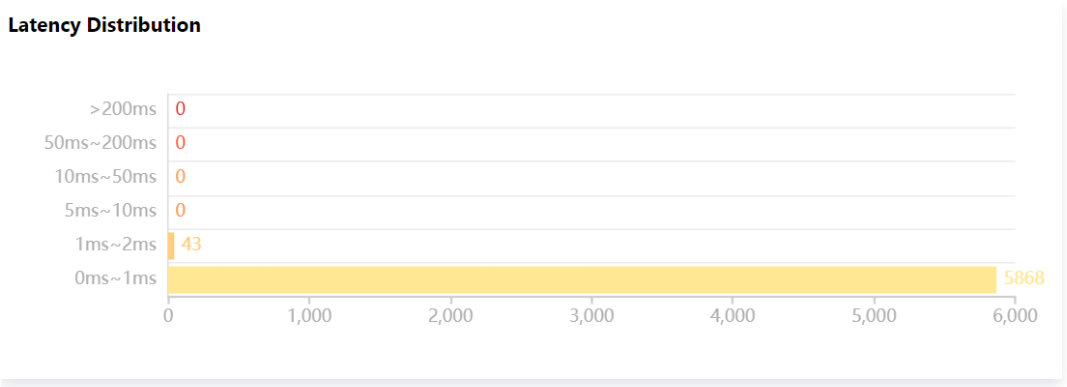
Latency line chart

The change trends of three key metrics for database request execution latency and the top five slow commands are displayed.

- **P99 Execution Latency**: Shows the change trend of the 99% request execution latency.
- **Average Execution Latency**: Shows the change trend of the average request execution latency.
- **Max Execution Latency**: Shows 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 database access commands are statistically represented in a bar chart format. The chart below shows the access count for the 'auth' and 'get' commands. Click on the 'Command Analysis' tab to view the statistical data for each access command. For more information, please refer to Command Analysis.

Command Word Analysis

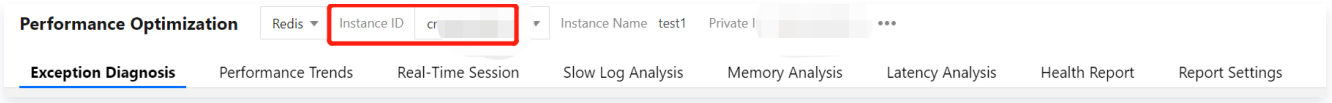
Last updated: 2024-11-01 17:11:18

Feature Overview

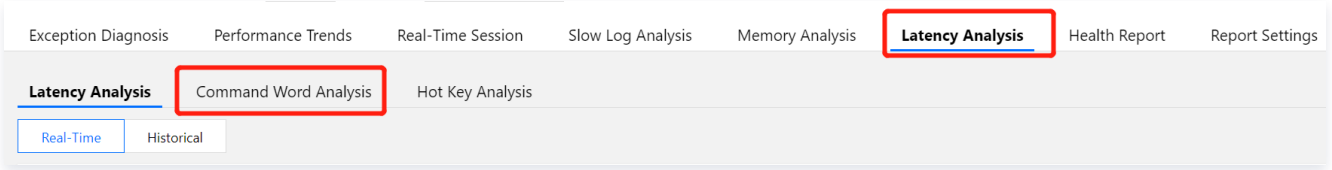
Command word analysis is designed to statistically analyze the quantity and latency of database access commands. Latency analysis swiftly identifies frequently accessed commands, while command word analysis further locates the specific time when the command is frequently executed and calculates the corresponding latency, assisting in troubleshooting, pinpointing issues, and optimizing performance.

Viewing Command Word Analysis

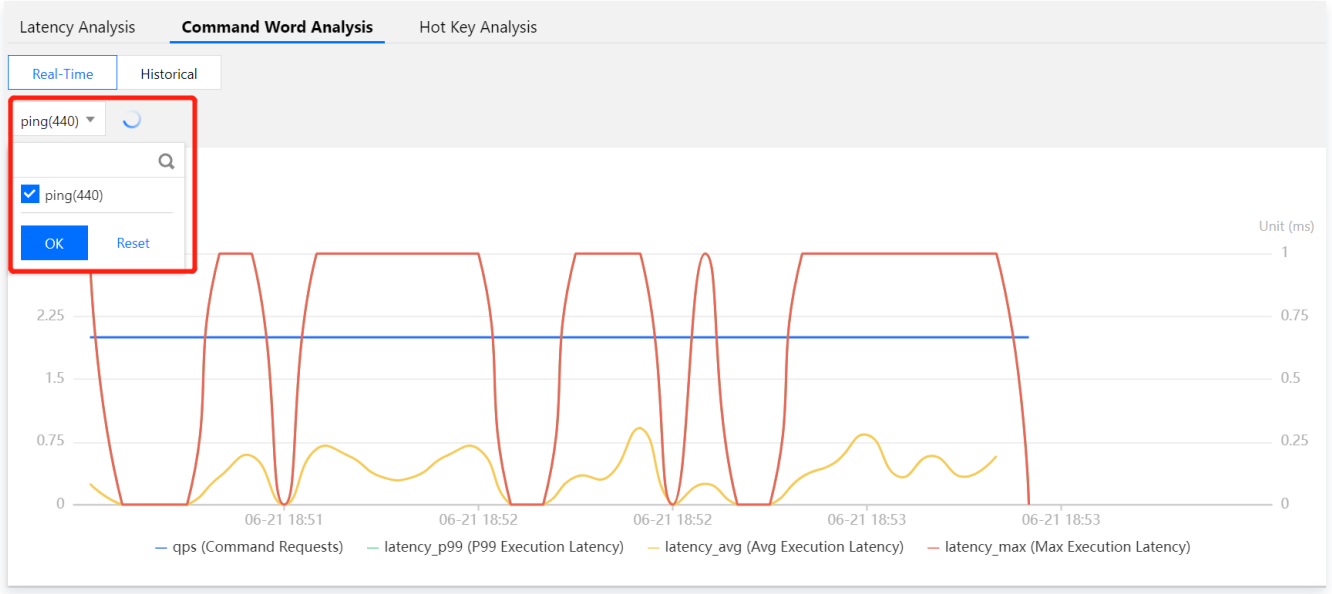
1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **Performance Optimization**.
3. At the top of the **Performance Optimization** page of **DBbrain**, select the target instance in the **Instance ID** drop-down list.



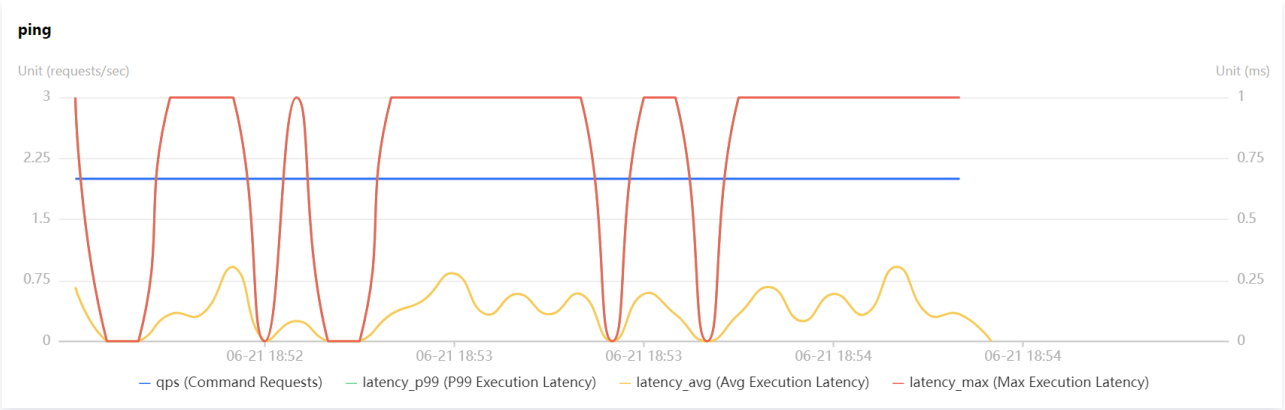
4. Select the **Latency Analysis** > **Command Word Analysis** tab, and set the collection granularity in the dropdown list next to **Auto Refresh** in the upper right corner. Options include 5 seconds, 15 seconds, and 30 seconds.



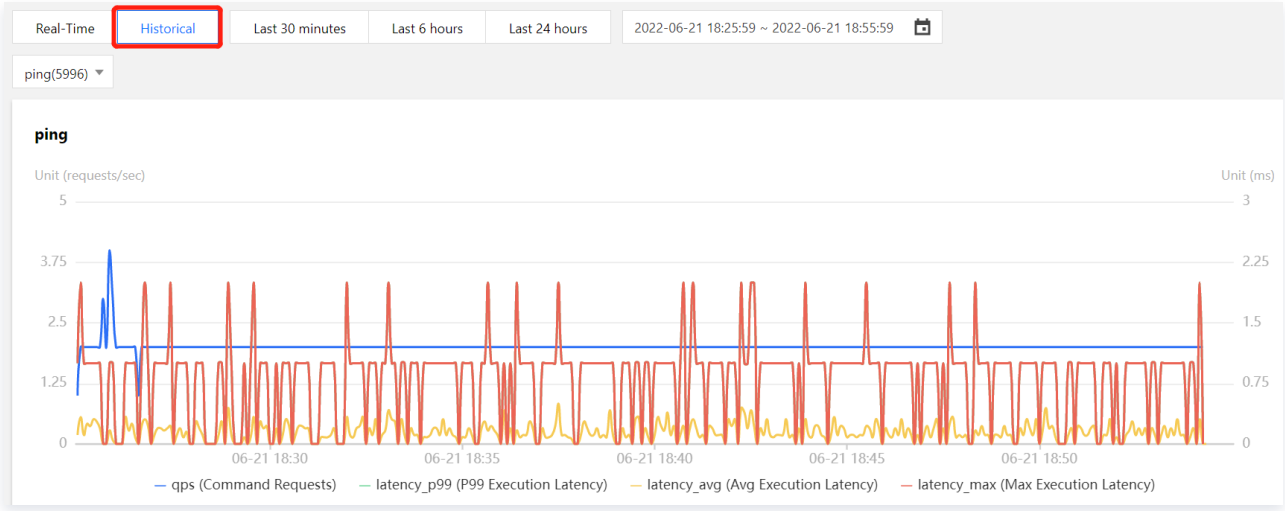
5. (Optional) Quickly filter target command words in the drop-down list in the top-left corner.



6. Analyze the trend data of the command words.
   - **Real-time Statistics**
     By default, real-time statistics include the change trends of command request count, P99 execution latency, average execution latency, and maximum execution latency. For more information on metrics, see Performance Trends performance indicators.



- **Historical Data**
  Click **History** to directly view statistics for the past 30 minutes, 6 hours, or 24 hours. In the time selection box, you can click to view statistics for the past 2 days.



Hot Key Analysis

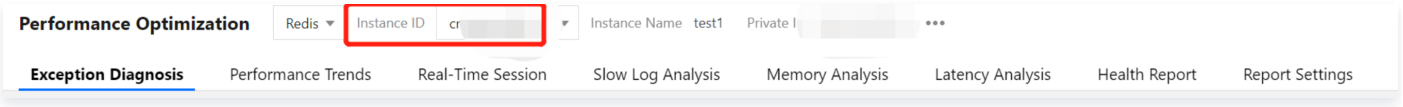
Last updated: 2024-11-01 17:11:29

Feature Overview

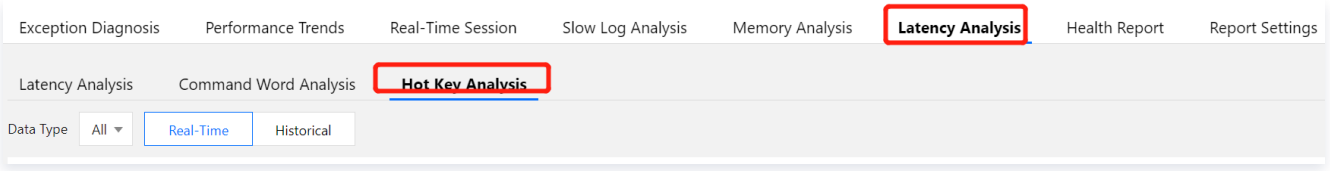
In Redis databases, we refer to Keys with high access frequency as hot Keys. When there are a large number of requests to the Redis database, most of these requests will access a specific Key in Redis. This can lead to excessively concentrated traffic, reaching the physical NIC limit, and potentially causing issues or even downtime for the Redis service.
By utilizing DBbrain's Hot Key Analysis feature, you can quickly identify hot Keys, providing crucial insights for database performance optimization.

Viewing the hot key analysis data

1. Log in to the TencentDB for Redis® console.
2. On the left sidebar, choose **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. On the **Latency Analysis** > **Hot Key Analysis** page, set the collection granularity in the drop-down list next to **Auto Refresh** in the upper right corner. The granularity can be 5 seconds, 15 seconds, or 30 seconds.



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 **History** to directly view statistics for the last 1 hour, 3 hours, 24 hours, or 7 days. In the time selection box, click to choose statistics within a continuous 7-day period in the last month, allowing you to query data within the last month, with a maximum continuous 7-day data analysis each time.

Sentinel Mode

Last updated: 2024-11-01 17:11:39

Scenario

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-l                                    SENTINEL sentinels test
1)  1) "ip"
    2) "10.
    3) "port"
    4) "6379"
    5) "name"
    6) "master"
    7) "runid"
    8) "4cfacd71
    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                                  SENTINEL get-master-addr-by-name master
1) "10.     .35"
2) "6379"
```

Connection Sample for the Sentinel Mode

Preparations

- The Redis instance version is 4.0 or 5.0.
- The database instance is in the **Running** status.
- In the TencentDB for Redis® console, navigate to the Instance Details page and locate the Network Information section to obtain the internal IPv4 address and port for connecting to the database. For more information, refer to the View Instance Details section.
- Obtain the account and password for database access. For detailed directions, see Managing Account.
- You have downloaded and installed 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 port.
        sentinels.add("XX.XX.XX.XX:6379");
        GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
        String dbPassword = "root:xxx";// Replace this with your database access password
        String sentinelPassword = "root:xxx";// Replace this with your database access password

        JedisSentinelPool jedisSentinelPool =
            new JedisSentinelPool(masterName, sentinels, poolConfig,
                2000, 2000, dbPassword,
                0, null, 2000, 2000,
                sentinelPassword, null);
        System.out.println("jedisSentinelPool.getResource().ping() = " +
jedisSentinelPool.getResource().ping());
        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 address and port
of your database instance
        sentinelConfig.setPassword(RedisPassword.of("xxx"));// Replace this with your database
access password
        sentinelConfig.setSentinelPassword(RedisPassword.of("xxx"));// Replace this with your
database access password
        JedisPoolConfig poolConfig = new JedisPoolConfig();
        JedisConnectionFactory connectionFactory = new JedisConnectionFactory(sentinelConfig,
poolConfig);
        connectionFactory.afterPropertiesSet();
        return connectionFactory;
    }

    @Bean
    @ConditionalOnBean(JedisConnectionFactory.class)
    public RedisTemplate<String, String> redisTemplate(@Qualifier("jedisConnectionFactory")
JedisConnectionFactory factory) {
        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.opsForValue().get("test"));
        return template;
    }

}
```