

云数据库 Redis

使用教程

产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

使用教程

实例创建

实例初始化

连接指导

代码示例

读写分离

使用教程

实例创建

最近更新时间：2017-11-22 17:38:04

- 1) 腾讯云Redis可通过[官网](#)直接购买，在购买页可以选择地域，规格，大小，数量
- 2) 用户在腾讯云购买云存储Redis后，可以通过管理中心进入控制台，在控制台实现对购买的Redis实例包括修改名称、分配至项目以及初始化等各种必要操作。



云存储Redis 云存储Redis-实例列表 全部项目 广州 深圳金融 上海 上海金融 北京 香港 新加坡 北京

更名公告：
尊敬的用户，即日起“腾讯云redis单机版”正式更名为“腾讯云redis主从版”，更名后用户不需要做任何处理，产品功能特性保持不变，100%兼容Redis协议，主从双节点架构，自动容灾。

[+ 新建](#)
[分配至项目](#)
[修改密码](#)
[换密](#)
[设置自动续费](#)
[取消自动续费](#)
[告警设置](#)
可通过实例ID,实例名,完整IP搜索

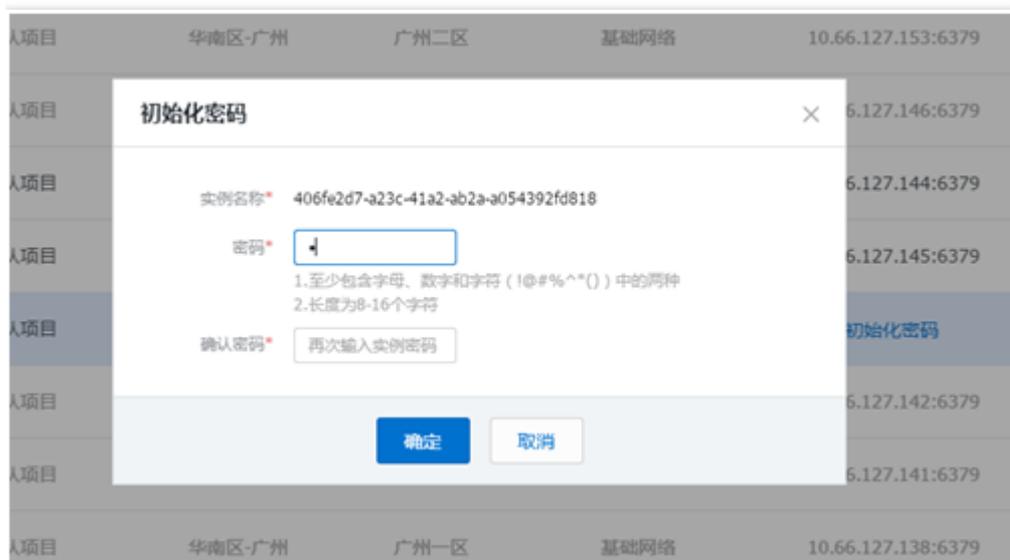
实例名称	实例状态	所属项目	可用区	网络	内网地址	规格	已使用/总容量	到期时间	管理
crs-1znib6aw	实例运行中	默认项目	广州三区	基础网络	10.66.187.77:6379	主从版	1.19G/8G	2016-12-15 19:24:50	扩容
crs-37jtt39	实例运行中	默认项目	广州二区	基础网络	10.66.130.25:6379	主从版	1.79MB/2G	2016-12-06 17:08:12	扩容

已选0项 共2项 每页显示行 10

实例初始化

最近更新时间：2017-05-05 17:54:21

新购买的实例的初始状态是实例待初始化，需要在控制台设置密码进行初始化。初始化需要输入Redis实例访问密码，如下图所示：



当初始化Redis密码完成后，Redis实例进入运行状态，如下图所示：

实例名称	实例状态	所属项目	可用区	网络	内网地址	规格	已使用/总容量	到期时间	管理
crs-1znib6aw	实例运行中	默认项目	广州三区	基础网络	10.66.187.77:6379	主从版	1.19G/8G	2016-12-15 19:24:50	扩容
crs-37ljt39	实例运行中	默认项目	广州二区	基础网络	10.66.130.25:6379	主从版	1.79MB/2G	2016-12-06 17:08:12	扩容

之后，就能登录云存储Redis、进入管理界面，以及进行分配至项目、续费升级等一系列操作。

连接指导

最近更新时间：2017-12-21 17:50:59

实例初始化完成后，连接腾讯云Redis时，需要输入设置的密码。主从版和集群版的连接示例如下

主从版连接示例

主从版支持2种格式

- 格式1，“实例id:密码”的格式类型，例如您的实例id是crs-bkuza6i3，设置的密码是abcd1234，则连接命令如下
`redis-cli -h IP地址 -p 端口 -a crs-bkuza6i3:abcd1234`

- 格式2，开源的格式类型，例如您设置的密码是abcd1234，则连接命令如下
`redis-cli -h IP地址 -p 端口 -a abcd1234`

(注：在2017.06.28后新购的主从版实例才支持格式2的访问方式)

集群版连接示例

集群版仅支持以下密码格式

- 实例id:密码”的格式类型，例如您的实例id是crs-bkuza6i3，设置的密码是abcd1234，则连接命令如下
`redis-cli -h IP地址 -p 端口 -a crs-bkuza6i3:abcd1234`

注意事项

- 腾讯云Redis不支持无密码访问

代码示例

最近更新时间：2017-06-28 17:27:58

以C链接代码示例

运行前必备: 下载并安装 hiredis <https://github.com/redis/hiredis>

示例代码：

```
#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: 192.168.0.195 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:%s", instance_id, 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;
}
```

读写分离

最近更新时间：2018-09-06 09:14:35

简介

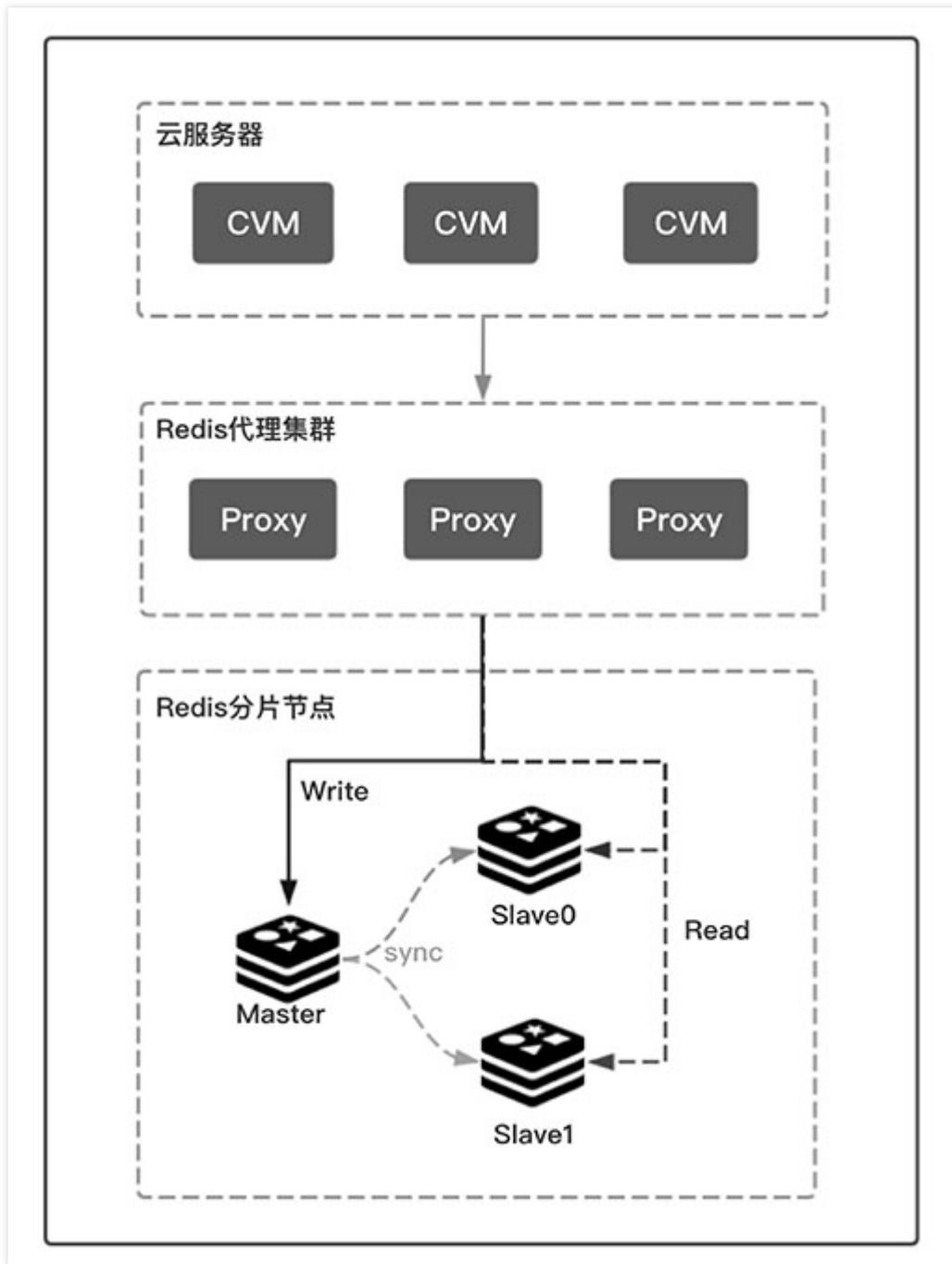
云数据库Redis支持开启和关闭读写分离功能，针对读多写少的业务场景，解决热点数据集中的读需求，最大支持1主5从模式，提供最大5倍的读性能扩展能力。

读写分离架构

社区版Redis4.0集群版原理

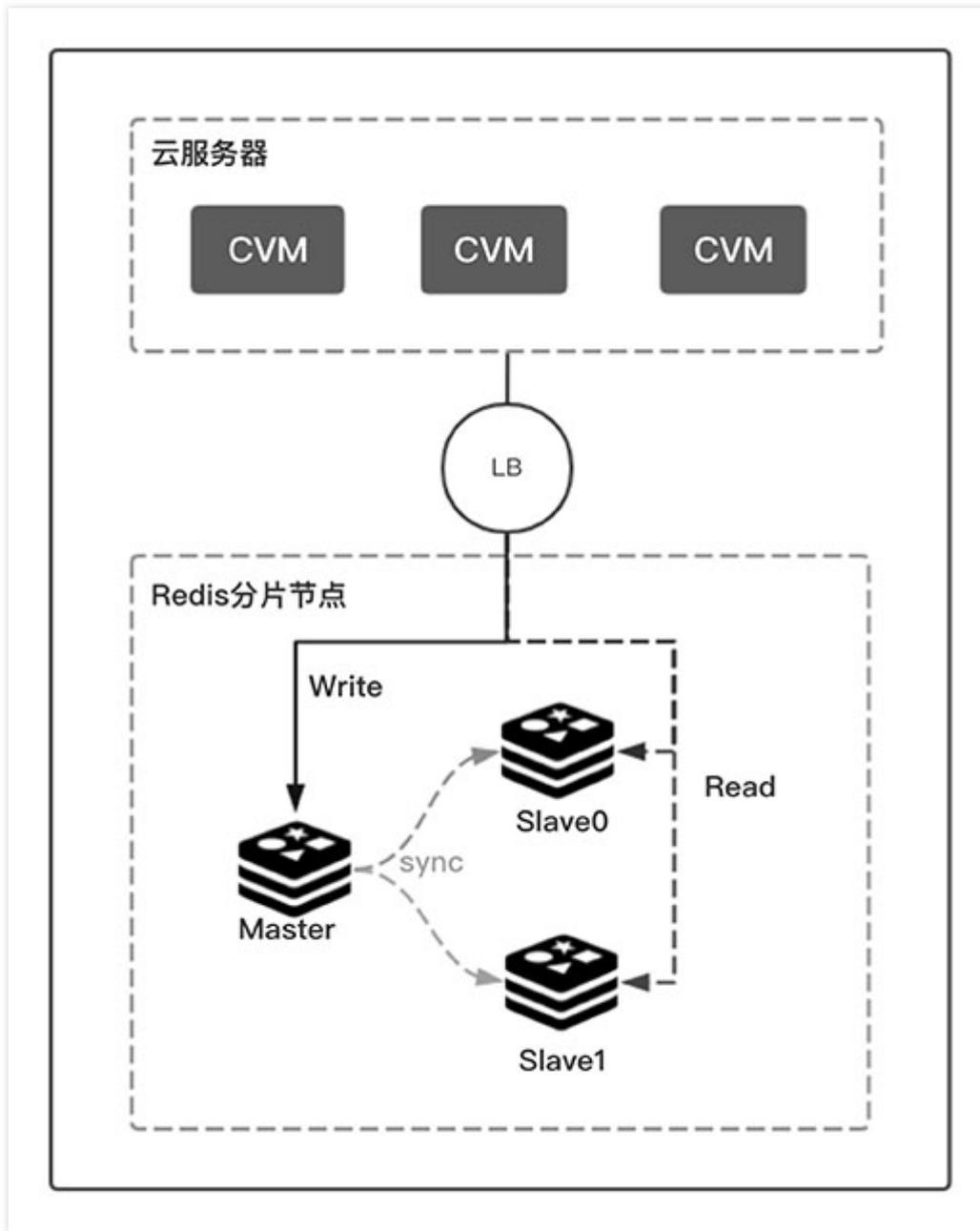
- **读写分离原理:** 云数据库Redis4.0集群版通过Proxy层实现自动读写分离;

- **读写分离权重:** 开启读写分离Proxy将按照主节点只写，从节点平均分配读请求的方式来提供访问；



CKV集群版读写分离原理

- **读写分离原理:** 云数据库Redis-CKV版系统天生支持读写分离架构，所有请求通过LB（负载均衡网关）分配到各个集群的各个节点，每个节点都有全局的Slot路由信息。开启读写分离后，如果读取的Key落在该节点将直接读取数据返回，否则将会根据路由信息将请求转发到对应的节点，由对应节点读取数据后返回该节点，然后返回客户端。
- **读写分离权重:** 因为CKV版本的流量全靠LB来路由，所以这里的读写权重是按照TCP连接的四元组（源IP、源端口、目的IP、目的端口）来平均分配；



读写分离计费

- 开启读写分离，将对只读副本收取一定的费用，具体费用详情请参见文档 [计费说明](#)