

分布式缓存服务

# 产品简介

文档版本 01  
发布日期 2024-10-17



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 目录

<b>1 图解华为云 Redis 与开源服务的差异</b>	<b>1</b>
<b>2 什么是分布式缓存服务</b>	<b>2</b>
<b>3 典型应用场景</b>	<b>6</b>
<b>4 安全</b>	<b>8</b>
4.1 责任共担	8
4.2 身份认证与访问控制	9
4.3 数据保护技术	9
4.4 审计与日志	10
4.5 服务韧性	11
4.6 监控安全风险	11
4.7 认证证书	11
4.8 安全白皮书	13
<b>5 DCS 实例类型</b>	<b>15</b>
5.1 Redis 单机实例	15
5.2 Redis 主备实例	17
5.3 Redis Proxy 集群实例	20
5.4 Redis Cluster 集群实例	25
5.5 Redis 读写分离实例	27
5.6 Redis 实例类型差异	29
5.7 Memcached 单机实例（已停售）	31
5.8 Memcached 主备实例（已停售）	33
<b>6 DCS 实例规格</b>	<b>35</b>
6.1 Redis 4.0/5.0 实例	35
6.2 Redis 6.0 实例	50
6.3 Redis 3.0 实例（已停售）	59
6.4 Memcached 实例（已停售）	62
<b>7 开源命令兼容性</b>	<b>64</b>
7.1 Redis 4.0 支持及禁用的命令	64
7.2 Redis 5.0 支持及禁用的命令	74
7.3 Redis 6.0 支持及禁用的命令	84
7.4 Web CLI 中支持和禁用命令	99

7.5 实例受限使用命令.....	103
7.6 部分命令使用限制.....	108
7.7 Redis 3.0 支持及禁用的命令（已停售）.....	109
7.8 Memcached 支持和禁用的命令（已停售）.....	113
<b>8 容灾和多活策略.....</b>	<b>117</b>
<b>9 产品规格差异.....</b>	<b>122</b>
9.1 Redis 版本差异.....	122
9.2 Redis 企业版与基础版差异.....	123
<b>10 与开源服务的差异.....</b>	<b>125</b>
<b>11 约束与限制.....</b>	<b>128</b>
<b>12 计费说明.....</b>	<b>131</b>
<b>13 权限管理.....</b>	<b>133</b>
<b>14 基本概念.....</b>	<b>138</b>
<b>15 与其他服务的关系.....</b>	<b>140</b>

# 1 图解华为云 Redis 与开源服务的差异



# 2 什么是分布式缓存服务

分布式缓存服务（Distributed Cache Service，简称DCS）是华为云提供的一款兼容Redis的高速内存数据处理引擎，为您提供即开即用、安全可靠、弹性扩容、便捷管理的在线分布式缓存能力，满足用户高并发及数据快速访问的业务诉求。

- **即开即用**  
DCS提供单机、主备、Proxy集群、Cluster集群、读写分离类型的缓存实例，拥有从128M到2048G的丰富内存规格。您可以通过控制台直接创建，无需单独准备服务器资源。  
其中Redis 4.0、Redis 5.0和Redis 6.0版本采用容器化部署，秒级完成创建。
- **安全可靠**  
借助华为云统一身份认证、虚拟私有云、云监控与云审计等安全管理服务，全方位保护实例数据的存储与访问。  
灵活的容灾策略，主备/集群实例从单AZ（可用区）内部署，到支持跨AZ部署。
- **弹性伸缩**  
DCS提供对实例内存规格的在线扩容与缩容服务，帮助您实现基于实际业务量的成本控制，达到按需使用的目标。
- **便捷管理**  
可视化Web管理界面，在线完成实例重启、参数修改、数据备份恢复等操作。DCS还提供基于RESTful的管理API，方便您进一步实现实例自动化管理。
- **在线迁移**  
提供可视化Web界面迁移功能，支持备份文件导入和在线迁移两种方式，您可以通过控制台直接创建迁移任务，提高迁移效率。

## DCS Redis

DCS Redis当前支持的Redis版本为Redis 4.0、Redis 5.0、Redis 6.0。

### 📖 说明

- DCS Redis 3.0已暂停售卖，建议使用Redis 4.0及以上版本。
- Redis 6.0实例目前仅在部分Region上线，具体请以控制台显示为准。

Redis是一种支持Key-Value等多种数据结构的存储系统。可用于缓存、事件发布或订阅、高速队列等**典型应用场景**。Redis使用ANSI C语言编写，提供字符串（**String**）、

哈希（**Hash**）、列表（**List**）、集合结构（**Set**、**Sorted\_Set**）、流（**Stream**）等数据类型的直接存取。数据读写基于内存，同时可持久化到磁盘。

DCS Redis拥有灵活的实例配置供您选择：

- 华为云DCS Redis 4.0/5.0/6.0基础版。

表 2-1 DCS Redis 4.0/5.0/6.0 基础版

实例类型	<p>提供单机、主备、Proxy集群、Cluster集群、读写分离类型，分别适配不同的业务场景。</p> <ul style="list-style-type: none"> <li>● 单机：适用于应用对可靠性要求不高、仅需要缓存临时数据的业务场景。单机实例支持读写高并发，但不做持久化，实例重启后原有缓存数据不会加载。</li> <li>● 主备：包含一个主节点，一个或多个备节点，主备节点的数据通过实时复制保持一致，当主节点故障后，备节点自动升级为主节点。同时用户可通过读写分离技术，在主节点上写，从备节点读，从而提升缓存的整体读写性能。</li> <li>● Proxy集群：在Cluster集群的基础上，增加挂载Proxy节点和ELB节点，通过ELB节点实现负载均衡，将不同请求分发到Proxy节点，实现客户端高并发请求。每个Cluster集群分片是一个双副本的主备实例，当主节点故障后，同一分片中的备节点会升级为主节点来继续提供服务。</li> <li>● Cluster集群：通过分片化分区来增加缓存的容量和并发连接数，每个分片是一个主节点和0到多个备节点，分片本身对外不可见。分片中主节点故障后，同一分片中备节点会升级为主节点来继续提供服务。用户可通过读写分离技术，在主节点上写，从备节点读，从而提升缓存的整体读写能力。</li> <li>● 读写分离：在主备实例的基础上，增加挂载Proxy节点和ELB节点，通过ELB节点实现负载均衡，将不同请求分发到Proxy节点，Proxy节点识别用户读写请求，将请求发送到主节点或备节点，从而实现读写分离。</li> </ul>
规格	Redis提供128MB~1024G的多种规格。
兼容开源Redis版本	DCS提供不同的实例版本，分别兼容开源Redis的4.0、5.0、6.0。
底层架构	在物理机上容器化部署，Redis 4.0、5.0单节点QPS达10万/秒，Redis 6.0基础版单机、主备实例QPS分别达15万/秒和17万/秒。
高可用与容灾	除单机实例外，其他类型的实例都提供Region内的跨可用区部署，实现实例内部节点间的电力、网络层面物理隔离。

有关开源Redis技术细节，您可以访问Redis官方网站<https://redis.io/>了解。

- 华为云DCS Redis 6.0企业版。  
华为云DCS 企业版，为华为云全自研的版本，由传统的master-worker线程模型升级为master-N\*worker多线程模型，对比传统缓存软件模型，整体性能N倍提升。DCS 企业版100%兼容Redis协议、模块和脚本，如脚本和事务的原子性等，在相同的硬件上，企业版的QPS约是Redis的两倍，且延迟可降低约60%。

多线程高性能版，适用于对单节点性能有超高要求的用户，比如互联网热点事件、大咖直播等高并发访问场景。Redis 6.0之前版本，通常受到慢请求单线程处理的制约，任何一个慢请求的出现都会带来其它用户请求的延时。Redis多线程高性能版对从IO到后端事件处理的全流程，进行了多线程并行优化；通过公平自旋锁实现多线程高效访问缓存数据；通过优化Key逐出算法，提升逐出效率1倍以上；通过支持SubKey过期提高了大Key读写性能。

表 2-2 DCS Redis 6.0 企业版

实例类型	Redis 6.0企业版分为高性能型和企业版存储型，目前只支持主备实例类型。（企业版存储型由内存+SSD磁盘组成。） 主备实例包含一个主节点，一个备节点，主备节点的数据通过实时复制保持一致，当主节点故障后，备节点自动升级为主节点。同时用户可通过读写分离技术，在主节点上写，从备节点读，从而提升缓存的整体读写性能。
规格	Redis提供8GB、16GB、32GB、64GB多种规格。
兼容开源版本	DCS企业版完全兼容开源Redis 6。
底层架构	基于虚拟机部署，单节点QPS达40万/秒。
高可用与容灾	除单机实例外，其他类型的实例都提供Region内的跨可用区部署，实现实例内部节点间的电力、网络层面物理隔离。

## DCS Memcached（已停售）

### 📖 说明

DCS Memcached已停售，建议使用Redis实例。

Memcached是一种内存Key-Value缓存系统，它支持简单字符串数据的存取，通常作为后端数据库内容缓存，以提升web的应用性能，降低对后端数据库的性能依赖，具体了解请参考[Memcached（已停售）典型应用场景](#)。

DCS全面兼容Memcached协议并增强实现了双机热备和数据持久化。

表 2-3 DCS Memcached 灵活的实例配置

实例类型	提供单机、主备两种类型，分别适配不同的业务场景。 单机：适用于应用对可靠性要求不高、仅需要缓存临时数据的业务场景。单机实例支持读写高并发，但不做持久化，实例重启后原有缓存数据不会加载。 主备：包含一个主节点和一个备节点，主备节点的数据通过实时复制保持一致，备节点对用户不可见且不能直接读写数据，当主节点故障后，备节点自动升级为主节点。
内存规格	单机和主备实例均提供2G、4G、8G、16G、32G、64G共6种内存规格。



高可用与容灾	主备实例提供Region内的跨可用区部署，实现实例内部节点间的电力、网络层面物理隔离。
--------	---

有关开源Memcached技术细节，您可以访问Memcached官方网站<https://memcached.org/>。

# 3 典型应用场景

## Redis 应用场景

很多大型电商网站、视频直播和游戏应用等，存在大规模数据访问，对数据查询效率要求高，且数据结构简单，不涉及太多关联查询。这种场景使用Redis，在速度上对传统磁盘数据库有很大优势，能够有效减少数据库磁盘IO，提高数据查询效率，减轻管理维护工作量，降低数据库存储成本。Redis对传统磁盘数据库是一个重要的补充，成为了互联网应用，尤其是支持高并发访问的互联网应用必不可少的基础服务之一。

以下举几个典型样例：

### 1. （电商网站）秒杀抢购

电商网站的商品类目、推荐系统以及秒杀抢购活动，适宜使用Redis缓存数据库。

例如秒杀抢购活动，并发高，对于传统关系型数据库来说访问压力大，需要较高的硬件配置（如磁盘IO）支撑。Redis数据库，单节点QPS支撑能达到10万，轻松应对秒杀并发。实现秒杀和数据加锁的命令简单，使用SET、GET、DEL、RPUSH等命令即可。

加锁部分，可参考最佳实践：[使用DCS实现分布式锁](#)

### 2. （视频直播）消息弹幕

直播间的在线用户列表，礼物排行榜，弹幕消息等信息，都适合使用Redis中的SortedSet结构进行存储。

例如弹幕消息，可使用ZREVRANGEBYSCORE排序返回，在Redis 5.0中，新增了zpopmax, zpopmin命令，更加方便消息处理。

### 3. （游戏应用）游戏排行榜

在线游戏一般涉及排行榜实时展现，比如列出当前得分最高的10个用户。使用Redis的有序集合存储用户排行榜非常合适，有序集合使用非常简单，提供多达20个操作集合的命令。

可参考最佳实践：[使用DCS实现排行榜功能](#)

### 4. （社交APP）返回最新评论/回复

在web类应用中，常有“最新评论”之类的查询，如果使用关系型数据库，往往涉及到按评论时间逆排序，随着评论越来越多，排序效率越来越低，且并发频繁。

使用Redis的List（链表），例如存储最新1000条评论，当请求的评论数在这个范围，就不需要访问磁盘数据库，直接从缓存中返回，减少数据库压力的同时，提升APP的响应速度。

## Memcached（已停售）典型应用场景

Memcached主要存储字符串类的简单key-value数据。

### 1. 静态页面缓存。

Web页面的内容片段，包括HTML，CSS和图片等静态数据，内容修改操作少，读取频繁，可以缓存到DCS Memcached实例，提高网站的访问性能。

### 2. 数据库前端缓存。

在动态系统中存在对大量数据读多写少的场景，如社交、博客网站大量查询用户信息、好友信息、文章信息等。为了减少磁盘数据库负载，提升性能，这些经常需要从数据库读取的数据，可以缓存在Memcached中。

适宜缓存的数据主要有：

- 经常被读取，实时性要求不高，且可以自动过期的数据。

例如网站首页最新文章列表、某某排行等数据，虽然新数据不断产生，但对用户体验影响比较小。这类数据可使用典型的缓存策略，设置合理的过期时间，当数据过期以后再从数据库中读取。为了让编辑或者其它人员能马上看到效果，可以再定一个缓存清除/刷新策略。

- 经常被读取并且实时性要求强的数据。

比如用户的好友列表，用户文章列表，用户阅读记录等。这类数据首先被载入到Memcached中，当发生更改（添加、修改、删除）时就刷新缓存数据。

### 3. 秒杀功能。

商品下单操作，牵涉数据库读取，写入订单，更改库存，及事务一致性要求，对于使用传统型数据库的平台来说，秒杀活动阶段要避免订单创建后库存缺货，同时提供流畅的用户体验，压力巨大。

可以利用Memcached的incr/decr功能，在内存中存储商品的库存量，秒杀的抢单过程主要在内存中完成，速度非常快，抢单成功即得一个订单号，这时再去支付页面完成订单的后续操作。

## 📖 说明

不适用Memcached的应用场景：

- 单个缓存对象大于1M

Memcached单个缓存对象的value值不能超过1M。超过1M的场景，建议使用Redis。

- Key的长度大于250字符

如确需使用Memcached，可将key先进行md5，得到散列值，然后存储key对应的散列值。

- 业务需要保证数据高可靠

开源Memcached不支持持久化数据，无法存副本、备份以及数据迁移。

注意：DCS的Memcached主备版本实现了数据持久化，具体可联系技术支持。

- 对数据结构和处理有高级要求

Memcached只支持key-value简单结构，不支持链表、集合等高级数据结构以及排序等一系列复杂操作。

# 4 安全

## 4.1 责任共担

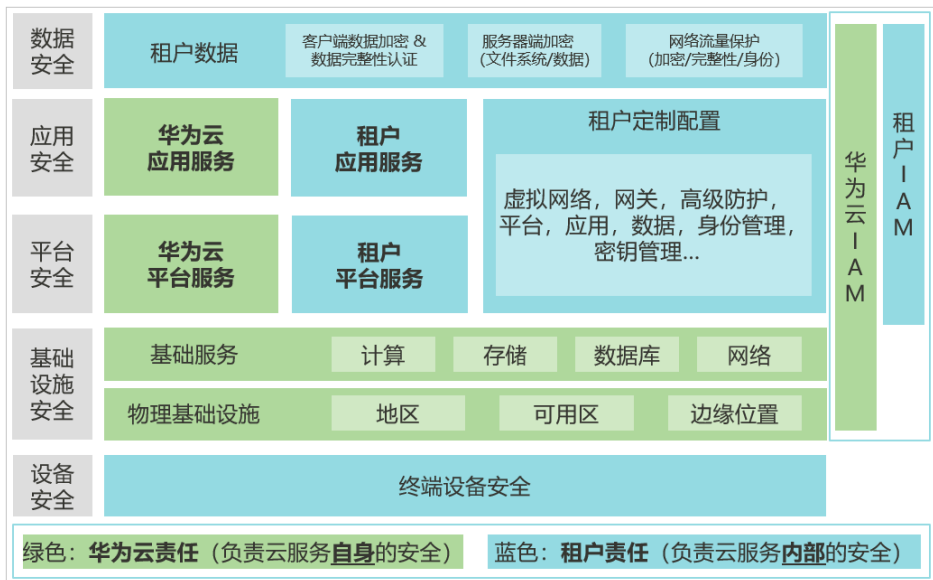
华为云秉承“将公司对网络和业务安全性保障的责任置于公司的商业利益之上”。针对层出不穷的云安全挑战和无孔不入的云安全威胁与攻击，华为云在遵从法律法规业界标准的基础上，以安全生态圈为护城河，依托华为独有的软硬件优势，构建面向不同区域和行业的完善云服务安全保障体系。

安全性是华为云与您的共同责任，如[图4-1](#)所示。

- **华为云**：负责云服务**自身**的安全，提供安全的云。华为云的安全责任在于保障其所提供的IaaS、PaaS和SaaS类云服务自身的安全，涵盖华为云数据中心的物理环境设施和运行其上的基础服务、平台服务、应用服务等。这不仅包括华为云基础设施和各项云服务技术的安全功能和性能本身，也包括运维运营安全，以及更广义的安全合规遵从。
- **租户**：负责云服务**内部**的安全，安全地使用云。华为云租户的安全责任在于对使用的IaaS、PaaS和SaaS类云服务内部的安全以及对租户定制配置进行安全有效的管理，包括但不限于虚拟网络、虚拟主机和访客虚拟机的操作系统，虚拟防火墙、API网关和高级安全服务，各项云服务，租户数据，以及身份账号和密钥管理等方面的安全配置。

《[华为云安全白皮书](#)》详细介绍华为云安全性的构建思路与措施，包括云安全战略、责任共担模型、合规与隐私、安全组织与人员、基础设施安全、租户服务与租户安全、工程安全、运维运营安全、生态安全。

图 4-1 华为云安全责任共担模型



## 4.2 身份认证与访问控制

### 身份认证

无论用户通过DCS控制台还是API访问DCS，都会要求访问请求方出示身份凭证，并进行身份合法性校验，同时提供登录保护和登录验证策略加固身份认证安全。DCS服务基于统一身份认证服务（Identity and Access Management, IAM），支持三种身份认证方式：[用户名密码](#)、[访问密钥](#)、[临时访问密钥](#)。同时还提供[登录保护](#)及[登录验证策略](#)。

### 访问控制

对企业中的员工设置不同的DCS访问权限，以达到不同员工之间的权限隔离，使用统一身份认证服务（IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制华为云资源的访问。详细介绍：[DCS权限策略](#)。

## 4.3 数据保护技术

DCS通过多种数据保护手段和特性，保障DCS的数据安全可靠。

表 4-1 DCS 的数据保护手段和特性

数据保护手段	简要说明	详细介绍
容灾和多活	根据对数据与服务不同可靠性要求，您可以选择将缓存实例部署在单可用区内（单机房），或者跨可用区（同城灾备）。	<a href="#">容灾多活策略</a>

数据保护手段	简要说明	详细介绍
副本冗余	副本通过增量数据同步的方式保持缓存数据一致，当网络发生异常或节点故障时，通过冗余副本自动故障切换，并且故障恢复后进行一次全量同步，保持数据一致性。	<a href="#">数据复制</a>
数据持久化	业务系统日常运行中可能出现一些小概率的异常事件。部分可靠性要求非常高的业务系统，除了要求缓存实例高可用，还要求缓存数据安全、可恢复，甚至永久保存，以便在缓存实例发生异常后能够使用备份数据进行恢复，保障业务正常运行。	<a href="#">备份与恢复</a>

## 4.4 审计与日志

### 云审计

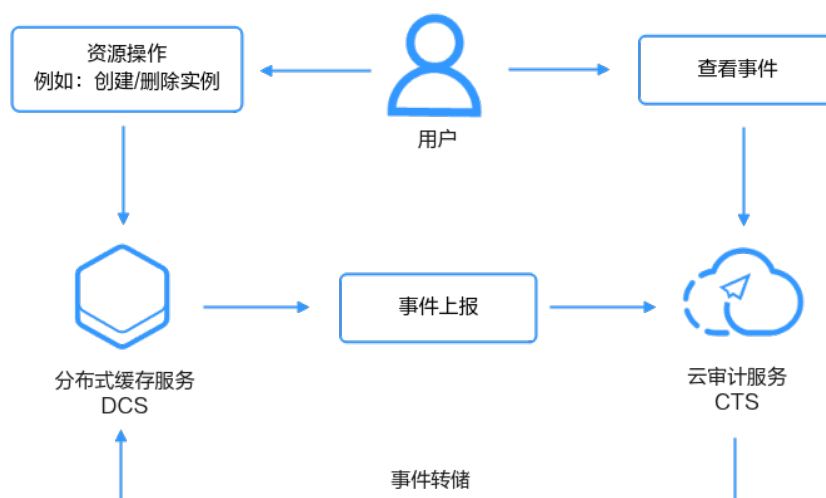
云审计服务（Cloud Trace Service, CTS），是华为云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。

用户开通云审计服务并创建和配置追踪器后，CTS可记录DCS的管理事件和数据事件用于审计。

CTS的详细介绍和开通配置方法，请参见[CTS快速入门](#)。

CTS支持追踪的DCS管理事件和数据事件列表，请参见[云审计服务支持的DCS操作列表](#)。

图 4-2 云审计服务



## 4.5 服务韧性

DCS服务提供了3级可靠性架构，通过双AZ容灾、AZ内实例容灾、实例数据多副本技术方案，保障服务的持久性和可靠性。

表 4-2 DCS 服务可靠性架构

可靠性方案	简要说明
双AZ容灾	DCS提供了主备、Cluster集群、Proxy集群多种类型实例，支持跨AZ容灾，当一个AZ异常时，不影响DCS实例持续提供服务。
AZ内实例容灾	AZ内，DCS主备、Cluster集群、Proxy集群实例通过节点冗余方式实现实例容灾，当检测到主节点故障后，快速切换到备节点，保障DCS实例持续提供服务。
数据容灾	通过支持数据多副本方式实现数据容灾。

## 4.6 监控安全风险

DCS提供基于云监控服务CES的资源 and 操作监控能力，帮助用户对每个DCS实例进行自动实时监控、告警和通知操作。用户可以实时掌握实例的各类业务请求、资源占用、带宽、并发操作数和流控次数等关键信息。

关于DCS支持的监控指标，以及如何创建监控告警规则等内容，请参见[监控](#)。

## 4.7 认证证书

### 合规证书

华为云服务及平台通过了多项国内外权威机构（ISO/SOC/PCI等）的安全合规认证，用户可自行[申请下载](#)合规资质证书。

图 4-3 合规证书下载

## 资源中心

华为云还提供以下资源来帮助用户满足合规性要求，具体请查看[资源中心](#)。

图 4-4 资源中心



## 4.8 安全白皮书

分布式缓存服务（Distributed Cache Service，简称DCS）是华为云提供的一款安全、可信的内存数据库服务。

DCS秉承华为云对租户的安全承诺，尊重租户数据主权，坚持中立、客观的立场，恪守业务边界，不碰租户数据，不会利用租户数据谋取商业价值。DCS允许租户快速发放不同类型的缓存实例，并可根据业务需要，对缓存资源进行弹性扩容。DCS提供自动备份、数据快照、数据恢复等功能，以防止数据丢失。提供参数修改功能，则允许租户根据业务需要进行实例调优。

DCS还提供多个特性来保障租户数据的可靠性和安全性，例如VPC、安全组、白名单、公网访问SSL加密连接、自动备份、数据快照和跨可用区部署等。

### 📖 说明

如果您需要了解DCS安全方面的详细操作指导，请参考[DCS安全最佳实践](#)。

## 网络隔离

VPC允许租户通过配置VPC入站IP范围，来控制连接缓存实例的IP地址段。DCS实例运行在租户独立的VPC内。租户可以创建一个跨可用区的子网组，之后可以根据业务需要，将部署DCS的高可用实例选择此子网完成，DCS在创建完实例后会为租户分配此子网的IP地址，用于缓存实例。DCS实例部署在租户VPC后，租户可通过VPN使其它VPC能够访问实例所在VPC，也可以在VPC内部创建ECS，通过私有IP连接缓存实例。租户可以综合运用子网、安全组、白名单等配置，来完成DCS实例的隔离，提升DCS实例的安全性。

## 访问控制

租户创建DCS实例时，可以选择配置安全组（仅Redis 3.0/Redis 6.0企业版/Memcached实例支持）或者白名单（仅Redis 4.0、Redis 5.0和Redis 6.0基础版支持）。

租户可以通过VPC，对DCS实例所在的安全组或白名单入站、出站规则进行限制，从而控制可以连接实例的网络范围。

配置安全组/白名单不需要重启实例。

租户创建DCS实例时，建议将访问方式设置为密码访问，并为实例设置访问密码，防止未经认证的客户端误操作实例，以达到对客户端进行认证访问的目的，提高实例使用的安全性。

## 传输和存储加密

Redis的传输协议RESP在Redis 6.0之前的版本仅支持明文传输。Redis 6.0基础版实例支持RESP3传输协议，支持SSL传输加密。

当前DCS实例在开启公网访问Redis 3.0时，支持基于Stunnel的客户端与服务端TLS加密传输。在开启公网访问时，指定的CA会为每个实例生成唯一的服务证书。客户端可以使用从服务控制台上下载的CA根证书，并在连接实例时提供该证书，对实例服务端进行认证并达到加密传输的目的。

由于需要兼容开源RESP协议，在未开启公网访问时，如果客户有传输加密诉求，建议在数据存储前使用加密算法进行加密后传输（例如AES 256等），并且控制在信任域内进行访问，加密后的数据在落盘保存时同样也会加密保存。

## 自动备份和手工备份

DCS提供自动备份和手工备份两种备份操作。自动备份默认未开启，需要租户选择是否开启，备份存储期限最多7天，同时开启自动备份后，允许对实例执行备份文件的恢复。DCS自动备份会进行全量数据备份，在备份期间会影响备节点的性能。手工备份是租户手动触发的实例全量备份，这些备份数据存储在华为OBS桶中，当租户删除实例时，会同步删除OBS桶中的快照。

## 数据复制

DCS的主备和集群实例支持部署高可用实例。租户可选择在单可用区或多可用区中部署实例。当租户选择跨AZ实例时，DCS实例会主动建立和维护Redis同步复制。在实例主节点故障的情况下，缓存实例会自动将备实例升为主节点，从而达到高可用的目的。如果租户使用缓存实例时，业务中读取数据比例大，可以选择4.0以上版本的读写分离实例或者集群多副本实例，缓存实例会自动维护主节点和多个备节点之间的数据同步复制，租户可以根据业务需要连接不同的地址进行读写分离。

## 数据删除

租户删除DCS实例时，存储在缓存实例中的数据都会被删除，任何人都无法查看及恢复数据。

# 5 DCS 实例类型

## 5.1 Redis 单机实例

Redis单机实例为单节点架构，不支持数据持久化，适用于不要求数据可靠性的缓存业务场景。

### 📖 说明

- DCS Redis 6.0目前仅在北京四和华南广州等部分Region上线。
- 不支持Redis版本的升级，例如，不支持Redis 4.0单机升级为Redis 5.0单机实例。如果需要使用高版本Redis单机实例，建议重新创建高版本Redis单机实例，然后将原有Redis实例的数据迁移到高版本实例上。
- 单机实例无法保证数据持久性，且不支持数据自动或手动备份，选用前请务必确认风险。

### 单机实例特点

1. 系统资源消耗低，支持高QPS  
单机实例不涉及数据同步、数据持久化所需消耗的系统开销，因此能够支撑更高的并发。Redis单机实例QPS达到10万以上。
2. 进程监控，故障后自动恢复  
DCS部署了业务高可用探测，单机实例故障后，30秒内会重启一个新的进程，恢复业务。
3. 即开即用，数据不做持久化  
单机实例开启后不涉及数据加载，即开即用。如果服务QPS较高，可以考虑进行数据预热，避免给后端数据库产生较大的并发冲击。
4. 低成本，适用于开发测试  
单机实例各种规格的成本相对主备减少40%以上。适用于开发、测试环境搭建。

总体说来，单机实例支持读写高并发，但不做持久化，实例重启时不保存原有数据。单机实例主要服务于数据不需要由缓存实例做持久化的业务场景，如数据库前端缓存，用以提升数据读取效率，减轻后端并发压力。当缓存中查询不到数据，可穿透至磁盘数据库中获取，同时，重启服务/缓存实例时，可从磁盘数据库中获取数据进行预热，降低后端服务在启动初期的压力。

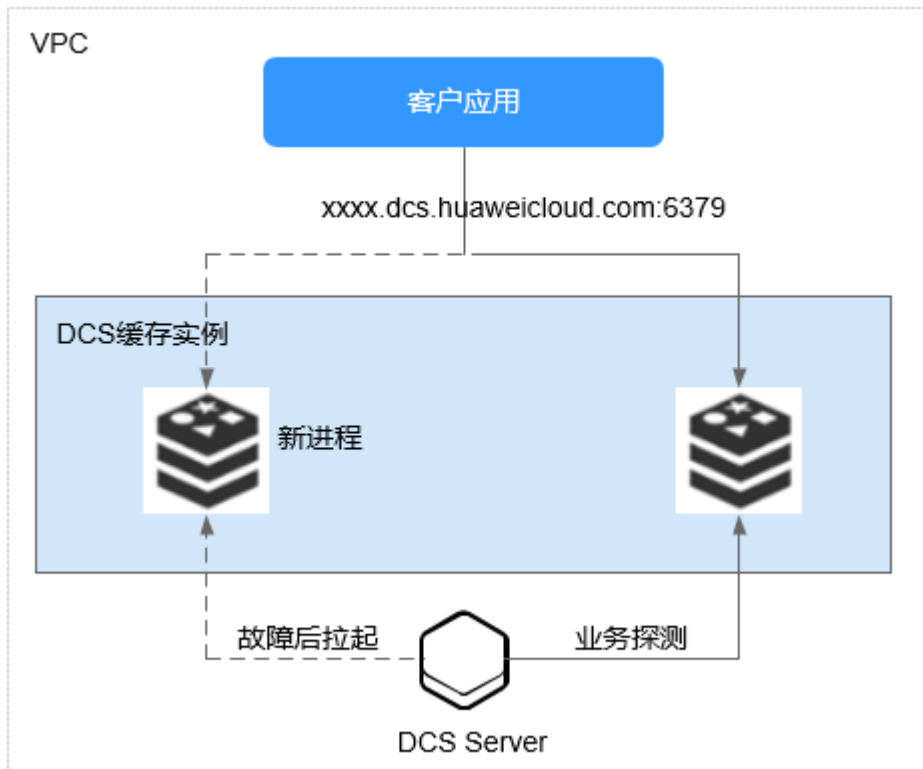
## 实例架构设计

DCS的Redis单机实例架构，如图5-1所示。

### 说明

Redis 3.0和Redis 6.0企业版实例不支持定义端口，端口固定为6379，Redis 4.0/5.0/6.0基础版实例支持自定义端口，如果不自定义端口，则使用默认端口6379。以下图中以默认端口6379为例，如果已自定义端口，请根据实际情况替换。

图 5-1 Redis 单机实例示意图



示意图说明：

- VPC**  
 虚拟私有云。实例的内部所有服务器节点，都运行在相同VPC中。
- 客户应用**  
 运行在ECS上的客户应用程序，即实例的客户端。  
 Redis实例兼容开源协议，可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。
- DCS缓存实例**  
 DCS实例，单机实例只有1个节点，1个Redis进程。  
 DCS实时探测实例可用性，当Redis进程故障后，DCS为实例重新拉起一个新的Redis进程，且该操作秒级完成，恢复业务。

## 5.2 Redis 主备实例

本章节主要介绍Redis缓存类型的主备实例。

### 📖 说明

- DCS Redis 6.0目前仅在北京四和华南广州等部分Region上线。
- 不支持Redis版本的升级，例如，不支持Redis 4.0主备升级为Redis 5.0主备实例。如果需要使用高版本Redis主备实例，建议重新创建高版本Redis主备实例，然后将原有Redis实例的数据迁移到高版本实例上。

### 主备实例特点

DCS的主备实例在单机实例基础上，增强服务高可用以及数据高可靠性。

主备实例具有以下特性：

#### 1. 持久化，确保数据高可靠

实例包含一个主节点和一个备节点，都默认开启数据持久化。

Redis 3.0主备实例的备节点对用户不可见，不支持客户端直接读写数据。

Redis 4.0、Redis 5.0和Redis 6.0基础版主备实例的备节点对用户可见，用户可以通过只读地址连接到备节点上读取数据。

#### 2. 数据同步

主备节点通过增量数据同步的方式保持缓存数据一致。

### 📖 说明

当网络发生异常或有节点故障时，主备实例会在故障恢复后进行一次全量同步，保持数据一致性。

#### 3. 故障后自动切换主节点，服务高可用

当主节点故障后，连接会有秒级中断、不可用，备节点在30秒内自动完成主备切换，切换完成后恢复正常访问，无需用户操作，保证业务平稳运行。

#### 4. 多种容灾策略

跨AZ部署（可用区）：DCS支持将主备实例的主备副本部署在不同的AZ内，节点间电力与网络均物理隔离。您可以将应用程序也进行跨AZ部署，从而达到数据与应用全部高可用。

#### 5. 读写分离

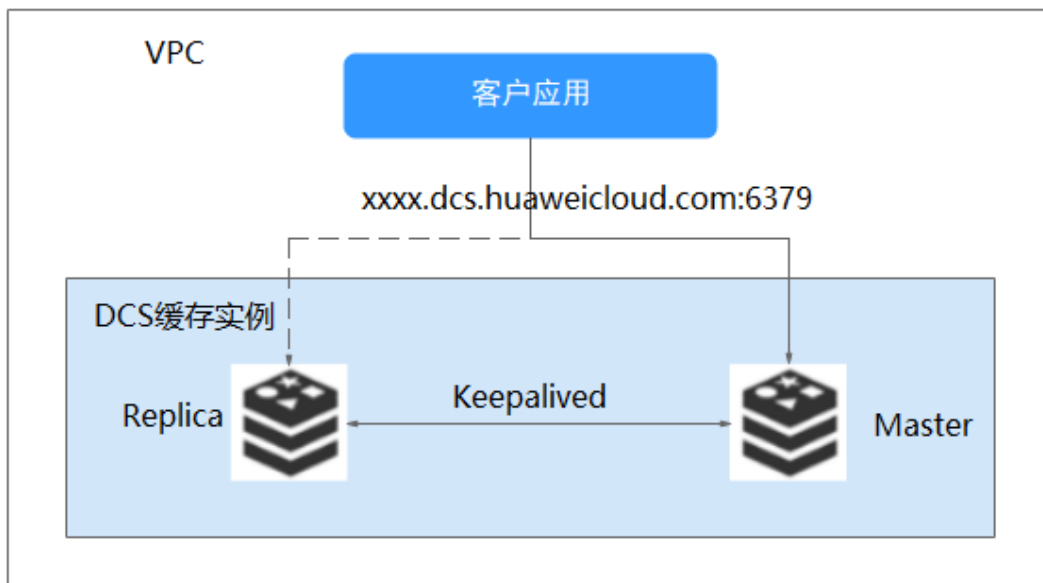
Redis 4.0、Redis 5.0和Redis 6.0基础版主备实例支持客户端读写分离，分别提供可读写的连接地址（主节点）和只读地址（备节点），客户端连接时，可分别选择主节点或备节点。

主备实例实现客户端读写分离，需要在客户端做配置。如需使用读写分离功能，推荐使用[读写分离实例](#)。

## Redis 3.0 主备实例架构设计

DCS的Redis 3.0主备实例架构，如[图5-2](#)所示。

图 5-2 Redis 3.0 主备实例示意图



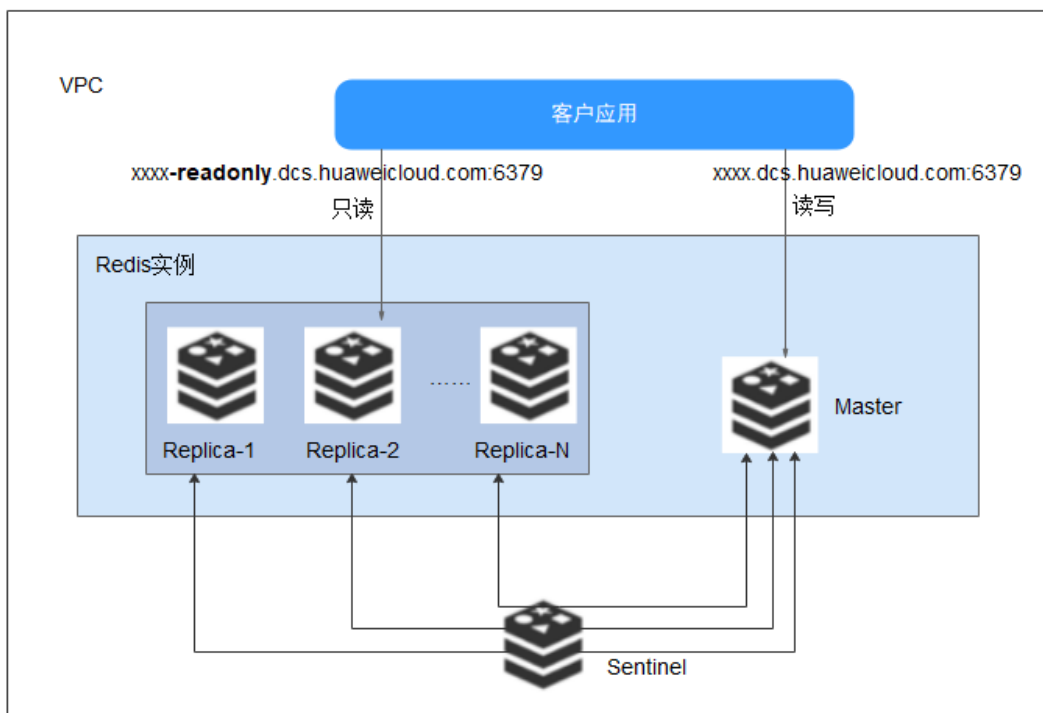
示意图说明：

- **VPC**  
虚拟私有云。实例的内部所有服务器节点，都运行在相同VPC中。
- **客户应用**  
运行在ECS上的客户应用程序，即Redis的客户端。  
Redis实例兼容开源协议，可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。
- **DCS缓存实例**  
DCS实例。主备实例包含了Master和Replica两个节点。默认开启数据持久化功能，同时保持节点间数据同步。  
DCS实时探测实例可用性，当主节点故障后，备节点升级为主节点，恢复业务。  
Redis的访问端口默认为6379。

## Redis 4.0/5.0/6.0 基础版主备实例架构设计

Redis 4.0/5.0/6.0基础版主备实例的架构设计，如下图所示。

图 5-3 Redis 4.0/5.0/6.0 基础版主备实例示意图



图说明如下：

1. Redis 4.0/5.0/6.0基础版主备实例，分别提供可读写的域名连接地址（用于连接主节点）和只读地址（用于连接备节点）。  
连接地址和只读地址，可通过控制台的实例详情页面获取。
2. Redis 4.0/5.0/6.0基础版主备实例使用哨兵模式（Sentinel）进行管理，Sentinel 会一直监控主备节点是否正常运行，当主节点出现故障时，进行主备倒换。  
Sentinel对用户不可见，仅在服务内部中使用。Sentinel的详细介绍可参考[什么是哨兵](#)。
3. 只读节点和读写节点规格一致，用户创建主备实例时，默认包含一个主节点和一个备节点。

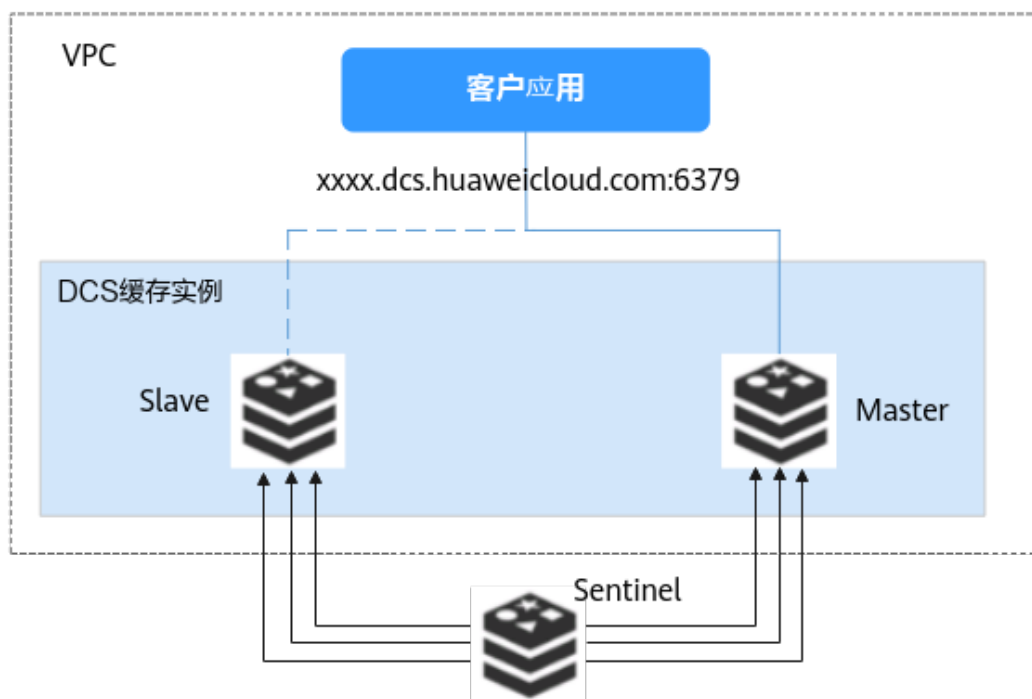
#### 📖 说明

- Redis 4.0/5.0/6.0基础版实例支持定义端口，如果不自定义端口，则使用默认端口 6379。图中以默认端口6379为例，如果已自定义端口，请根据实际情况替换。
- Redis 4.0/5.0/6.0基础版主备实例在只读域名不带负载均衡，对于可靠性和时延敏感的应用场景，建议使用集群或读写分离类型的实例。
- Redis 4.0/5.0/6.0基础版主备实例的只读域名在从节点发生故障场景下会出现请求失败的情况，对于可靠性和时延敏感的应用场景，建议使用读写分离类型的实例。

## Redis 6.0 企业版主备实例架构设计

Redis 6.0企业版主备实例架构设计如[图5-4](#)所示。

图 5-4 Redis 6.0 企业版主备实例示意图



图说明如下：

- **VPC**

虚拟私有云。实例的内部所有服务器节点，都运行在相同VPC中。

- 📖 **说明**

VPC内访问，客户端需要与Redis 6.0企业版主备实例处于相同VPC，并且配置安全组访问规则。[如何选择和配置Redis实例以及客户端的安全组。](#)

- **客户应用**

运行在ECS上的客户应用程序，即Redis的客户端。

Redis实例兼容开源协议，可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。

- **DCS缓存实例**

DCS实例。主备实例包含了Master和Replica两个节点。

Redis 6.0企业版主备实例使用哨兵模式（Sentinel）进行管理，Sentinel会一直监控主备节点是否正常运行，当主节点出现故障时，进行主备倒换。

Sentinel对用户不可见，仅在服务内部中使用。Sentinel的详细介绍可参考[什么是哨兵](#)。

Redis的访问端口默认为6379。

## 5.3 Redis Proxy 集群实例

DCS Redis Proxy集群实例，是基于LVS+Proxy的高可用集群版本。Redis Proxy集群实例特点：

- 客户端与云服务解耦。



- 性能与Cluster集群一样，支持百万并发。
- 提供灵活的内存规格档位，适配不同场景。

**说明**

- 不支持Redis版本的升级，例如，不支持Redis 4.0 Proxy集群升级为Redis 5.0 Proxy集群实例。如果需要使用高版本Redis Proxy集群实例，建议重新创建高版本Redis Proxy集群实例，然后将原有Redis实例的数据迁移到高版本实例上。
- 在连接Proxy集群实例时，客户端不需要做特殊配置，使用方式与单机、主备实例相同，使用实例IP地址或域名连接即可，不需要知晓和使用Proxy节点或分片地址。

## Redis 3.0 Proxy 集群实例

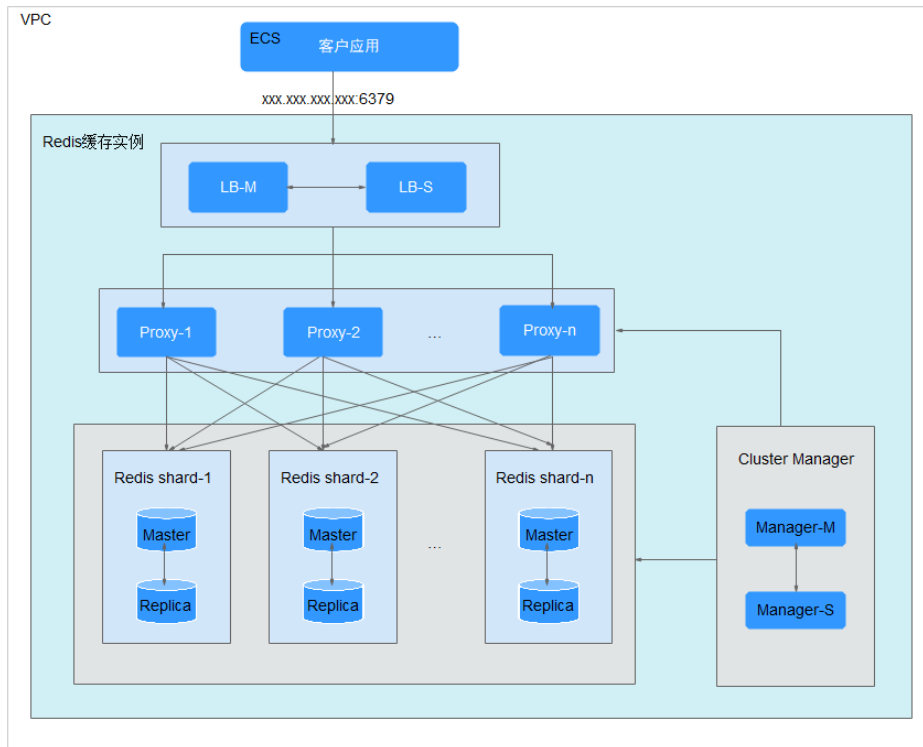
DCS Redis 3.0 Proxy集群实例基于开源Redis 3.0版本构建，兼容[开源codis](#)，提供64G~1024G多种大容量规格版本，支持x86的CPU架构，用于满足**百万级以上并发与大容量数据缓存**的需要。Redis集群的数据分布式存储和读取，由DCS内部实现，用户无需投入开发与运维成本。

Redis集群实例由“负载均衡器”、“Proxy服务器”、“集群配置管理器”、“**集群分片**”共4个部分组成。

**表 5-1** Redis 3.0 集群实例规格和 Proxy 节点数、分片数的对应关系

集群版规格	Proxy节点数	分片数 ( Shard )
64G	3	8
128G	6	16
256G	8	32
512G	16	64
1024G	32	128

图 5-5 Redis 3.0 Proxy 集群实例示意图



示意图说明：

- **VPC**

虚拟私有云。集群实例的内部所有服务器节点，都运行在相同VPC中。

- **说明**

如果Proxy集群实例没有开通公网访问，则客户端需要与集群实例处于相同VPC，并且配置安全组访问规则。

如果Proxy集群实例开通了公网访问，则客户端可以部署在外部网络，并通过集群实例绑定的弹性公网ip地址访问Redis Proxy集群实例。

相关参考：[公网连接Redis3.0 Proxy实例](#)，[如何选择和配置Redis实例以及客户端的安全组](#)。

- **客户应用程序**

客户应用程序，即Redis集群客户端。

Redis可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。

- **LB-M/LB-S**

负载均衡服务器，采用主备高可用方式。Redis集群实例提供访问的IP地址和域名，即为负载均衡服务器地址。

- **Proxy**

Redis集群代理服务器。用于实现Redis集群内部的高可用，以及承接客户端的高并发请求。

支持使用Proxy节点的IP连接集群实例。

- **Redis shard**

Redis集群的分片。

每个分片也是一个Redis主备实例，分片上的主实例故障时，系统会自动进行主备切换，集群正常提供服务。

某个分片的主备实例都故障，集群可正常提供服务，但该分片上的数据不能读取。

- **Cluster manager**

集群配置管理器，用于存储集群的配置信息与分区策略。用户不能修改配置管理器的信息。

## Redis 4.0/5.0/6.0 基础版 Proxy 集群实例

### 📖 说明

Redis 4.0及以上版本的Proxy集群实例，当前仅部分区域支持，请以控制台实际上线区域为准。

DCS Redis 4.0/5.0/6.0 Proxy集群实例基于开源Redis的4.0/5.0/6.0版本构建，兼容[开源codis](#)，提供4G~1024G多种大容量规格版本。

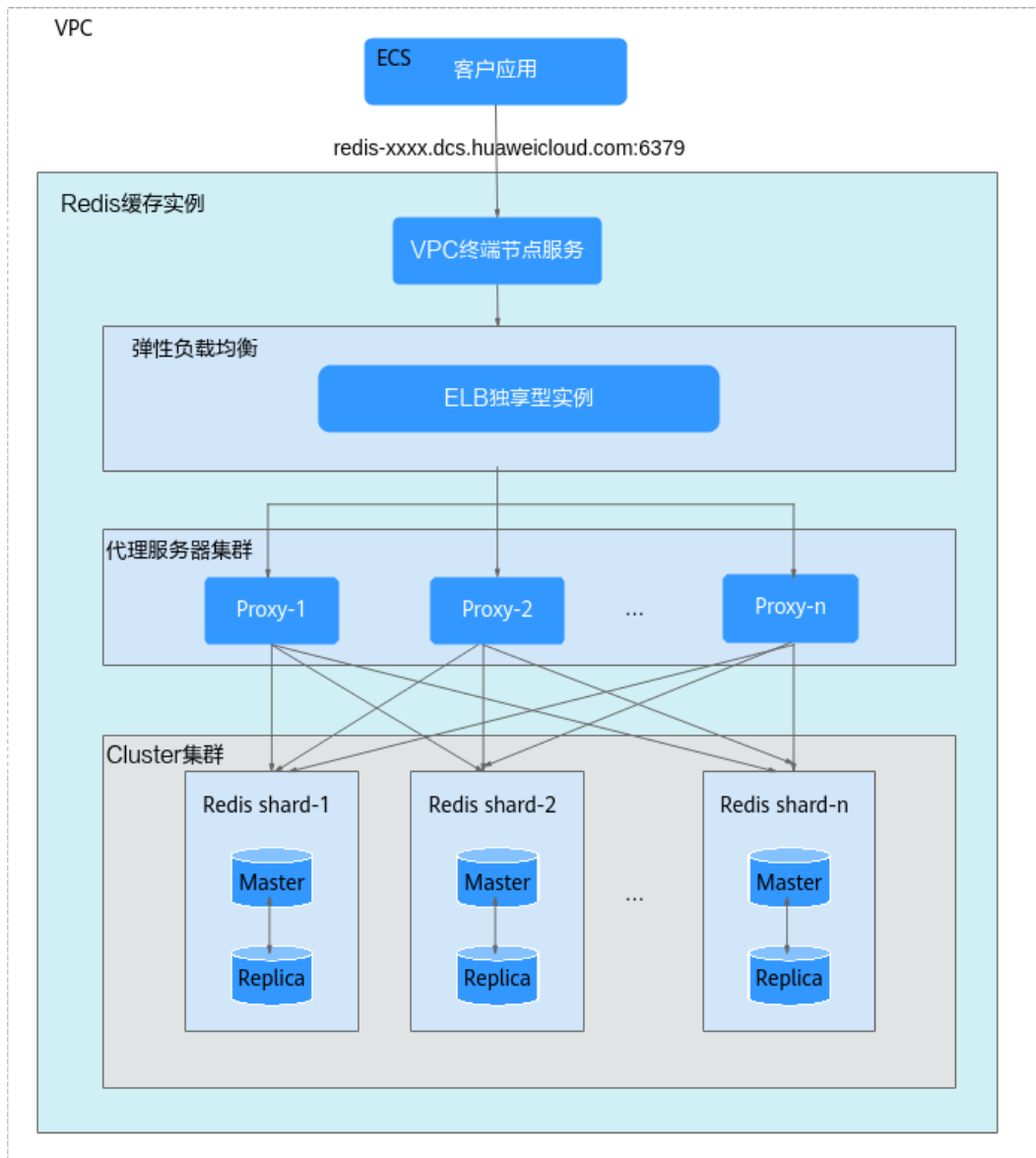
Proxy集群每种实例规格对应的分片数，如[表5-2](#)所示，在创建实例时，支持自定义分片大小。当前暂时不支持自定义副本数，默认每个分片为双副本架构。

**每个分片内存=实例规格/分片数**，例如，集群规格为48GB的实例，分片数为6，则每个集群分片的大小为48G/6=8G。

**表 5-2** Redis 4.0/5.0/6.0 基础版 Proxy 集群实例规格和 Proxy 节点数、分片数的对应关系

集群版规格	Proxy节点数	分片数	每个分片内存 ( GB )
4GB	3	3	1.33
8GB	3	3	2.67
16GB	3	3	5.33
24GB	3	3	8
32GB	3	3	10.67
48GB	6	6	8
64GB	8	8	8
96GB	12	12	8
128GB	16	16	8
192GB	24	24	8
256GB	32	32	8
384GB	48	48	8
512GB	64	64	8
768GB	96	96	8
1024GB	128	128	8

图 5-6 Redis 4.0/5.0/6.0 基础版 Proxy 集群实例示意图



实例示意图说明：

- **VPC**

虚拟私有云。集群实例的内部所有服务器节点，都运行在相同VPC中。

- **说明**

客户端需要与集群实例处于相同VPC，并且实例白名单允许客户端的IP地址访问。

- **客户应用程序**

客户应用程序，即Redis集群客户端。

Redis可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。

- **VPC终端节点服务**

VPC终端节点服务，主要是将Redis缓存实例配置为VPC终端节点支持的服务，用户可以直接通过VPC终端节点服务的地址访问。

Redis Proxy集群实例提供的IP地址和域名，即为VPC终端节点服务的地址。

- **ELB**

弹性负载均衡服务器，采用集群高可用方式，支持多可用区部署。

- **Proxy**

Redis集群代理服务器。用于实现Redis集群内部的高可用，以及承接客户端的高并发请求。

暂不支持使用Proxy节点的IP连接集群实例。

- **Cluster集群**

Redis集群的分片。

每个分片也是一个双副本的Redis主备实例，分片上的主实例故障时，系统会自动进行主备切换，集群正常提供服务。

某个分片的主备实例都故障，集群可正常提供服务，但该分片上的数据不能读取。

## 5.4 Redis Cluster 集群实例

DCS Redis Cluster集群实例，是原生Cluster的集群版本。Redis Cluster集群实例的特点：

- 兼容Redis原生Cluster集群。
- 继承smart client的设计方案。
- 相比主备，数倍性能提升。

Redis Cluster集群实例支持从客户端实现读写分离，相关操作请参考[Cluster集群实例读写分离](#)。

### 📖 说明

- 不支持Redis版本的升级，例如，不支持Redis 4.0 Cluster集群升级为Redis 5.0 Cluster集群实例。如果需要使用高版本Redis Cluster集群实例，建议重新创建高版本Redis Cluster集群实例，然后将原有Redis实例的数据迁移到高版本实例上。
- 客户端连接Redis Cluster集群实例与连接单机、主备、Proxy集群实例的方式不同，连接示例请参考[连接Redis缓存实例](#)。
- 当前Redis 6.0基础版Cluster集群仅在北京四和华南广州等部分Region支持。

## Cluster 集群实例

Cluster版Redis集群兼容[开源Redis的Cluster](#)，基于smart client和无中心的设计方案，对服务器进行分片。

Cluster版Redis集群每种实例规格对应的分片数，如[表5-3](#)所示。

在创建DCS Cluster集群实例时，可以自定义分片大小。如果不自定义分片大小，使用系统默认分片，**每个分片的大小=实例规格/分片数**，例如，集群规格为48GB的实例，分片数为6，则每个集群分片的大小为48GB/6=8GB。

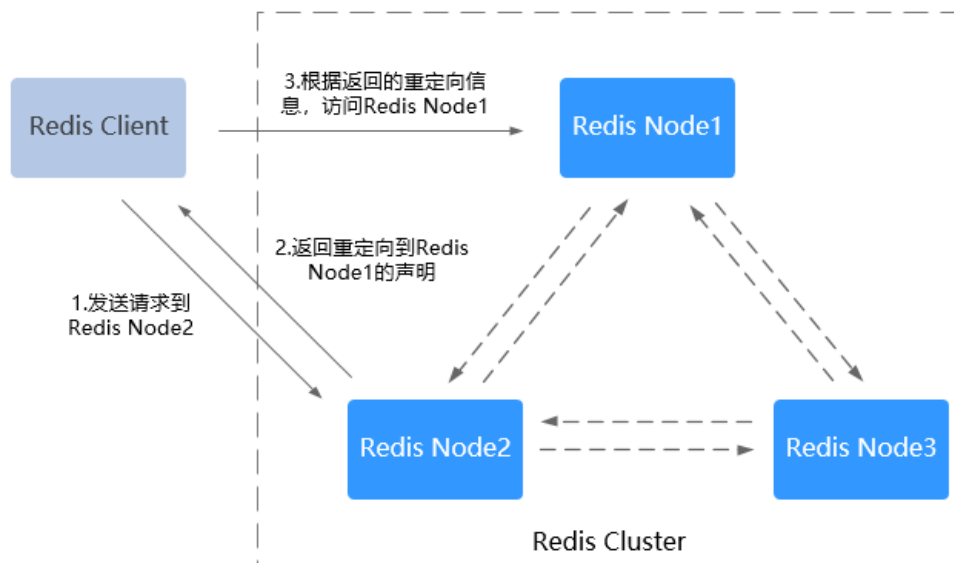
表 5-3 Cluster 集群实例规格和分片数的对应关系

集群版规格	分片数
4GB/8GB/16GB/24GB/32GB	3
48GB	6
64GB	8
96GB	12
128GB	16
192GB	24
256GB	32
384GB	48
512GB	64
768GB	96
1024GB	128
2048GB	128

- 无中心架构

Redis Cluster的任意节点都可以接收请求，但节点会将请求发送到正确的节点上执行，同时，每一个节点也是主从结构，默认包含一个主节点和一个从节点，由Redis Cluster根据选举算法决定节点主从属性。

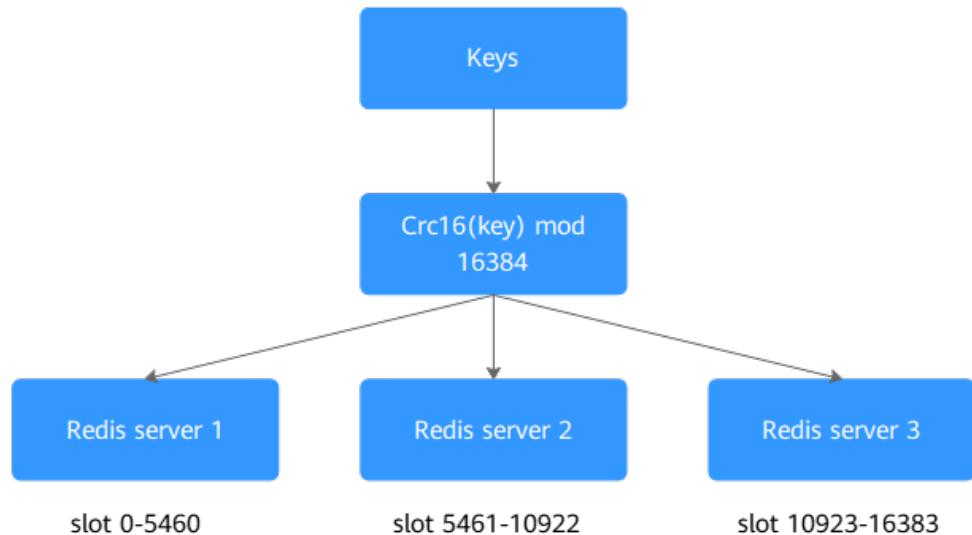
图 5-7 Redis Cluster 无中心架构



- 数据预分片

Redis Cluster会预先分配16384个slot，每个Redis的server存储所有slot与redis server的映射关系。key存储在哪个slot中，由 $Crc16(key) \bmod 16384$ 的值决定。如下图所示：

图 5-8 Redis Cluster 预分片示意图



## 5.5 Redis 读写分离实例

本章节主要介绍DCS服务的读写分离实例，读写分离主要适用于读高并发、写请求较少的业务场景，解决高并发的性能问题，节约运维成本。

### 说明

- 读写分离实例类型，仅在部分Region支持，具体请以控制台显示为准。
- Redis 4.0/5.0/6.0基础版本支持读写分离实例，其他版本暂不支持。

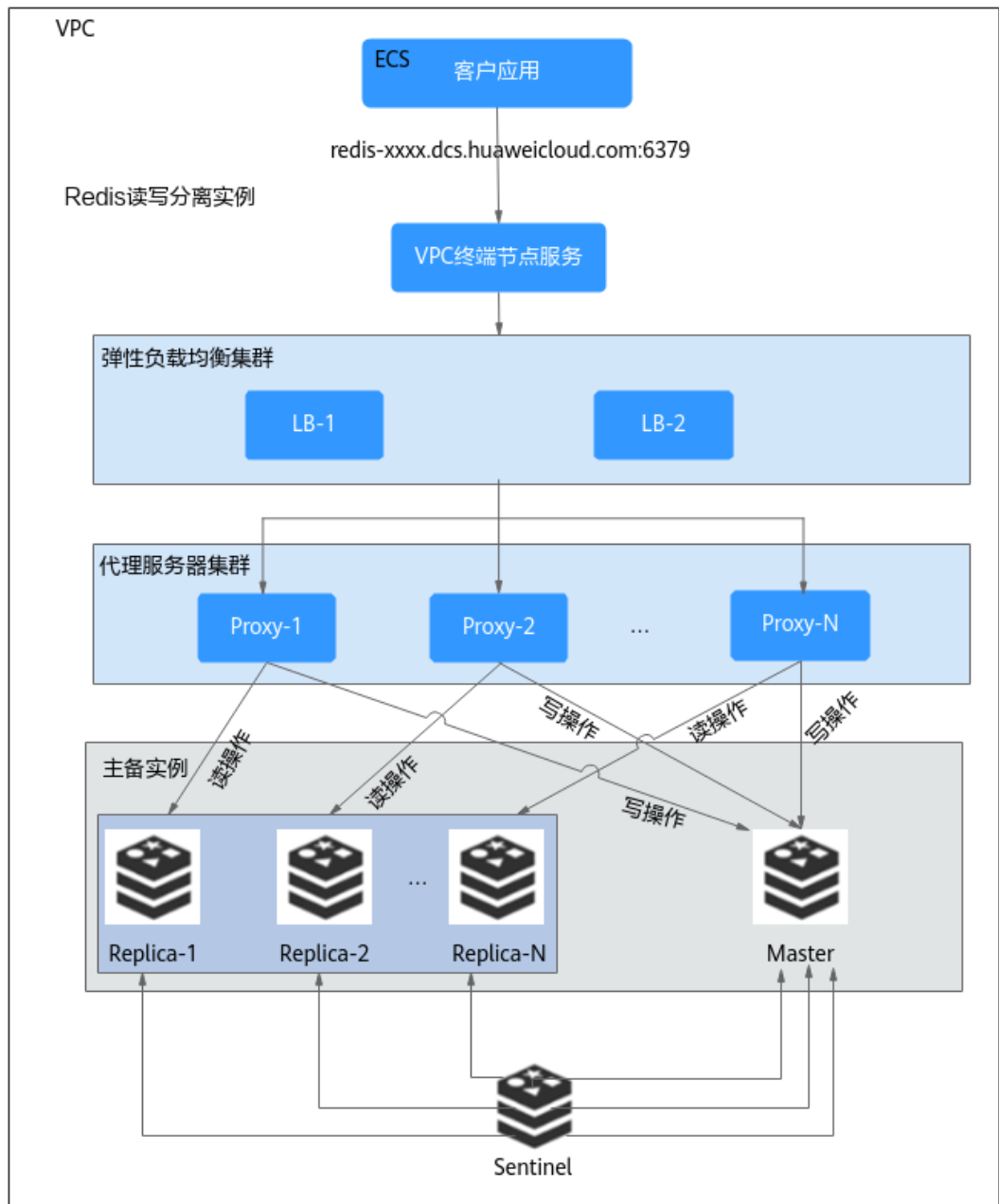
读写分离实例，默认为从服务端侧实现的读写分离，通过Proxy节点识别用户读写请求，如果是写请求，则转发给主节点，如果是读请求，则转发给备节点，不需要用户在客户端做任何配置。

主备读写分离实例使用限制：

1. 主备读写分离实例的读请求会发送到从节点，从节点从主节点同步数据会有一些的时延。  
请确保业务侧不依赖主从同步的时延，如果对主从同步时延有依赖的场景，不适用读写分离实例，请考虑普通主备或集群。
2. 主备读写分离实例适用于写少读多的场景，如果写流量过大，可能导致主从断连，或断连后主从同步失败，导致读请求性能下降。  
写流量大的场景请考虑普通主备或集群。
3. 从节点故障后，需要一定的时间从主节点全量同步数据，同步数据期间，从节点不对外提供服务，此时实例读请求性能会下降。  
推荐使用<32G内存规格的实例，内存规格越小，主从全量同步数据时间越少，同步断链时间越短。

## 读写分离实例

图 5-9 读写分离实例



实例示意图说明：

- **VPC终端节点服务**

VPC终端节点服务，主要是将Redis缓存实例配置为VPC终端节点支持的服务，用户可以直接通过VPC终端节点服务的地址访问。

Redis读写分离实例提供的IP地址和域名，即为VPC终端节点服务的地址。

- **ELB**

弹性负载均衡服务器，采用集群高可用方式，支持多可用区部署。



- Proxy**  
 代理服务器集群。通过Proxy节点识别用户读写请求，如果是写请求，则转发给主节点，如果是读请求，则转发给备节点，不需要用户在客户端做任何配置。
- Sentinel集群**  
 监控主备节点状态，当主节点出现故障或异常时，进行主备倒换，保证服务不中断。
- 主备实例**  
 读写分离实例，后端是一个主备实例，包含了主和备两个节点。默认开启数据持久化功能，同时保持节点间数据同步。  
 主备节点支持跨可用区部署。

## 5.6 Redis 实例类型差异

Redis单机、主备、读写分离、Proxy集群和Cluster集群实例，在特性支持、特性限制以及命令限制有部分差异，具体请查看[表5-4](#)。

### 📖 说明

DCS Redis 3.0已下线，暂停售卖，建议使用Redis 4.0及以上版本。

表 5-4 不同实例类型的差异说明

对比项	单机/主备/读写分离	Proxy集群	Cluster集群
兼容Redis版本	兼容开源Redis 3.0/4.0/5.0/6.0。可在创建实例时选择版本号。企业版当前只支持主备实例。	兼容开源Redis 3.0/4.0/5.0/6.0。可在创建实例时选择版本号。	兼容开源Redis 4.0/5.0/6.0版本。可在创建实例时选择版本号。
特性支持	<ul style="list-style-type: none"> <li>支持event notify。</li> <li>支持pipeline。</li> </ul>	<ul style="list-style-type: none"> <li>支持pipeline、mset、mget。</li> <li>支持scan、keys、slowlog。</li> <li>支持发布订阅。</li> </ul>	<ul style="list-style-type: none"> <li>支持event notify。</li> <li>支持brpop、blpop、brpoplpush。</li> <li>支持发布订阅。</li> </ul>

对比项	单机/主备/读写分离	Proxy集群	Cluster集群
特性限制	仅单机实例不支持数据持久化及备份与恢复的功能。	<ul style="list-style-type: none"> <li>lua脚本受限使用，所有的key必须在同一个slot，否则会报错，建议使用 <b>hashtag</b> 技术。</li> <li>支持多个key的命令中，部分命令要求所有key必须属于同一个slot，否则会报错，建议使用 <b>hashtag</b> 技术。具体限制多个key必须属于同一slot的命令，请参考<b>Proxy集群多Key命令说明</b>。</li> <li>不支持event notify用法。</li> </ul>	<ul style="list-style-type: none"> <li>lua脚本受限使用，所有的key必须在同一个slot，建议使用<b>hashtag</b>技术。</li> <li>需要客户端SDK支持redis cluster协议，需要能够处理"-MOVED"响应。</li> <li>使用pipeline、mset/mget模式时，所有key必须属于同一个slot，否则报错，建议使用<b>hashtag</b>技术。</li> <li>使用event notify时，需要建立与每个redis-server的连接，分别处理每个连接上的事件。</li> <li>执行scan、keys等遍历类或者全局类命令时，需要对每个redis-server分别执行该命令。</li> </ul>
客户端协议	使用传统Redis客户端即可。	使用传统Redis客户端即可，不需要支持Redis Cluster协议。	需要客户端支持Redis Cluster协议。
命令限制	不支持的Redis命令，请参考 <b>开源命令兼容性</b> 。 读写分离实例受限使用的命令请参考 <b>实例受限使用的Redis命令</b> 。	不支持的Redis命令，请参考 <b>开源命令兼容性</b> 。 Proxy集群实例受限使用的命令请参考 <b>实例受限使用的Redis命令</b> 。	不支持的Redis命令，请参考 <b>开源命令兼容性</b> 。 Cluster集群实例受限使用的命令请参考 <b>实例受限使用的Redis命令</b> 。

对比项	单机/主备/读写分离	Proxy集群	Cluster集群
副本数	<p>单机实例为单副本，只有一个节点。</p> <p>主备和读写分离实例默认为双副本，默认为一主一从的架构。</p> <p>在创建Redis主备和读写分离实例时，支持自定义副本数，形成一主多从的架构。</p> <p>目前Redis 3.0、Redis 6.0企业版实例不支持自定义副本数。</p>	<p>每个集群分片都为双副本，但不支持为分片新增副本，每个分片是一主一从的架构。</p>	<p>每个集群分片默认为双副本，支持自定义副本数，每个分片可以是一主多从的架构。在创建实例时，也可以定义为单副本，单副本表示实例只有主节点，无法保障数据高可靠。</p>

## 相关问题

- [实例是否支持多DB?](#)
- [实例的CPU规格是怎么样的?](#)
- [实例是否支持读写分离?](#)

## 5.7 Memcached 单机实例（已停售）

### 📖 说明

DCS Memcached已停售，建议使用Redis实例。

本章节主要介绍Memcached单机实例的特点和架构。

### 单机实例特点

1. 系统资源消耗低，支持高QPS  
单机实例不涉及数据同步、数据持久化所需消耗的系统开销，因此能够支撑更高的并发。Memcached的单机实例QPS达到10万以上。
2. 进程监控，故障后自动恢复  
DCS部署了业务高可用探测，单机实例故障后，30秒内会重启一个新的进程，恢复业务。
3. 即开即用，数据不做持久化  
单机实例开启后不涉及数据加载，即开即用。如果服务QPS较高，可以考虑进行数据预热，避免给后端数据库产生较大的并发冲击。
4. 低成本，适用于开发测试

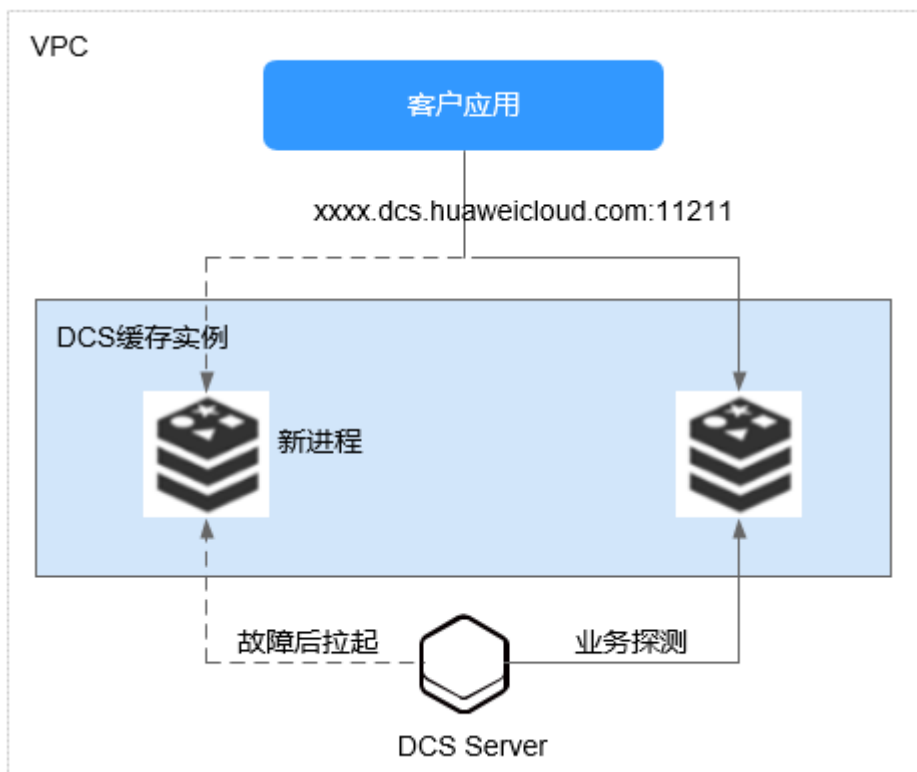
单机实例各种规格的成本相对主备减少40%以上。适用于开发、测试环境搭建。

总体说来，单机实例支持读写高并发，但不做持久化，实例重启时不保存原有数据。单机实例主要服务于数据不需要由缓存实例做持久化的业务场景，如数据库前端缓存，用以提升数据读取效率，减轻后端并发压力。当缓存中查询不到数据，可穿透至磁盘数据库中获取，同时，重启服务、缓存实例时，可从磁盘数据库中获取数据进行预热，降低后端服务在启动初期的压力。

## 实例架构设计

DCS的Memcached单机实例架构，如图5-10所示。

图 5-10 Memcached 单机实例示意图



示意图说明：

- **VPC**

虚拟私有云。实例的内部所有服务器节点，都运行在相同VPC中。

- 📖 **说明**

Memcached单机实例不支持公网访问，客户端需要与实例处于相同VPC，并且配置安全组访问规则。

相关参考：[如何选择和配置Redis实例以及客户端的安全组。](#)

- **客户应用**

运行在ECS上的客户应用程序，即实例的客户端。

Memcached实例兼容开源协议，可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。

- **DCS缓存实例**

DCS实例，单机实例只有1个节点，1个Memcached进程。

DCS实时探测实例可用性，当Memcached进程故障后，DCS为实例重新拉起一个新的Memcached进程，恢复业务。

Memcached实例访问端口为11211。

## 5.8 Memcached 主备实例（已停售）

### 📖 说明

DCS Memcached已停售，建议使用Redis实例。

本章节主要描述Memcached主备实例。

### 主备实例特点

Memcached主备实例在单机实例基础上，增强服务高可用以及数据高可靠性。

Memcached主备实例具有以下特性：

1. **持久化，确保数据高可靠**

实例包含一个主节点和一个备节点，都默认开启数据持久化。同时支持数据持久化，确保数据高可靠。

Memcached主备实例的备节点对用户不可见，不支持客户端直接读写数据。

2. **数据同步**

主备节点通过增量数据同步的方式保持缓存数据一致。

### 📖 说明

当网络发生异常或有节点故障时，主备实例会在故障恢复后进行一次全量同步，保持数据一致性。

3. **故障后自动切换主节点，服务高可用**

当主节点故障后，备节点在30秒内自动完成主备切换，无需用户操作，业务平稳运行。

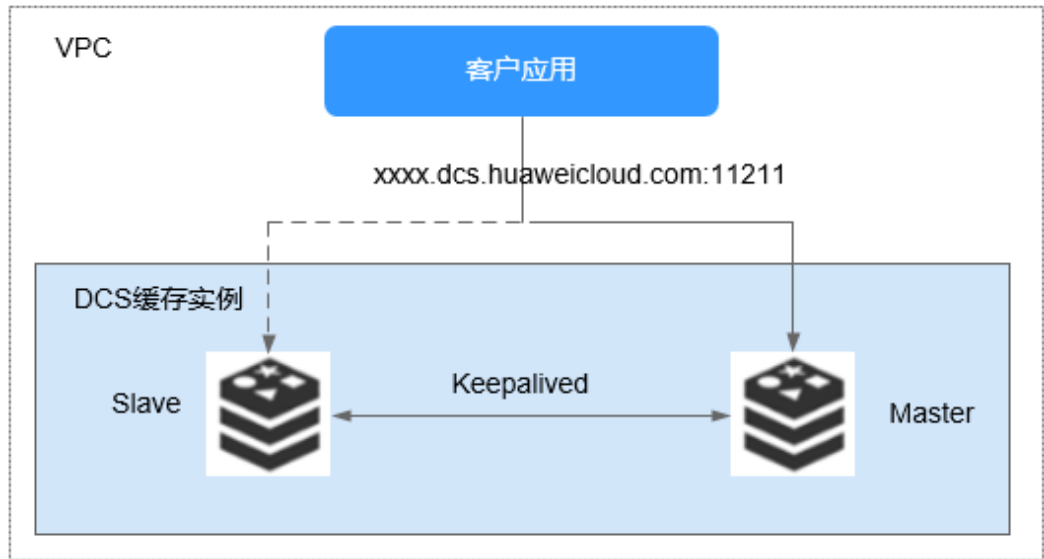
4. **多种容灾策略**

跨AZ部署（可用区）：DCS支持将主备实例部署在不同的AZ内，节点间电力与网络均物理隔离。您可以将应用程序也进行跨AZ部署，从而达到数据与应用全部高可用。

### Memcached 主备实例架构设计

DCS的Memcached主备实例架构，如[图5-11](#)所示。

图 5-11 Memcached 主备实例示意图



示意图说明：

- **VPC**

虚拟私有云。实例的内部所有服务器节点，都运行在相同VPC中。

- **说明**

Memcached主备实例不支持公网访问，客户端需要与主备实例处于相同VPC，并且配置安全组访问规则。

相关参考：[如何选择和配置Redis实例以及客户端的安全组](#)。

- **客户应用**

运行在ECS上的客户应用程序，即Memcached的客户端。

Memcached实例兼容开源协议，可直接使用开源客户端进行连接，关于多语言客户端连接示例，请参考用户指南的[连接缓存实例](#)。

- **DCS缓存实例**

DCS实例。主备实例包含了Master和Slave两个节点。默认开启数据持久化功能，同时保持节点间数据同步。

DCS实时探测实例可用性，当主节点故障后，备节点升级为主节点，恢复业务。

Memcached实例访问端口为11211。

# 6 DCS 实例规格

## 6.1 Redis 4.0/5.0 实例

本节介绍DCS Redis 4.0和Redis 5.0实例的产品规格，包括内存规格、实例可使用内存、连接数上限、最大带宽/基准带宽、参考性能（QPS）等。

实例各项指标如下：

- 实例已使用内存：您可以通过查看监控指标“内存利用率”和“已用内存”查看实例内存使用情况。
- 最大连接数：表示允许客户端同时连接的个数，即连接并发数。具体实例的连接数，可查看监控指标“活跃的客户数量”。最大连接数对应参数maxclients，实例创建后支持在控制台“实例详情>参数配置”中修改。
- QPS：即Query Per Second，表示数据库每秒执行的命令数。QPS的测试方法，可参考[性能测试方法](#)。
- 带宽：即实例的内网带宽，您可以查看监控指标“流控次数”，确认带宽是否超过限额。其中，带宽还涉及监控指标“带宽使用率”，该指标仅供参考，可能会出现超过100%的情况，具体原因，可查看[带宽使用率指标会超过100%的原因](#)。

### 📖 说明

- Redis 4.0和Redis 5.0实例支持“单机”、“主备”、“Proxy集群”、“Cluster集群”和“读写分离”类型。
- Redis 4.0和Redis 5.0实例有“x86计算”和“Arm计算”两种CPU架构。推荐使用“x86计算”类型，部分Region已停售“Arm计算”类型。
- 不同区域支持的规格可能存在差异，请以控制台可选择的实际规格为准。

### 单机实例

Redis 4.0/5.0单机实例有x86和Arm两种CPU类型，具体规格如下表。

表 6-1 Redis 4.0 和 Redis 5.0 单机实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
0.125	0.125	10,000/10,000	40/40	80,000	x86: redis.single.xu1.tiny.128 Arm: redis.single.au1.tiny.128
0.25	0.25	10,000/10,000	80/80	80,000	x86: redis.single.xu1.tiny.256 Arm: redis.single.au1.tiny.256
0.5	0.5	10,000/10,000	80/80	80,000	x86: redis.single.xu1.tiny.512 Arm: redis.single.au1.tiny.512
1	1	10,000/50,000	80/80	80,000	x86: redis.single.xu1.large.1 Arm: redis.single.au1.large.1
2	2	10,000/50,000	128/128	80,000	x86: redis.single.xu1.large.2 Arm: redis.single.au1.large.2
4	4	10,000/50,000	192/192	80,000	x86: redis.single.xu1.large.4 Arm: redis.single.au1.large.4



内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
8	8	10,000/50,000	192/192	100,000	x86: redis.single.xu1.large.8 Arm: redis.single.au1.large.8
16	16	10,000/50,000	256/256	100,000	x86: redis.single.xu1.large.16 Arm: redis.single.au1.large.16
24	24	10,000/50,000	256/256	100,000	x86: redis.single.xu1.large.24 Arm: redis.single.au1.large.24
32	32	10,000/50,000	256/256	100,000	x86: redis.single.xu1.large.32 Arm: redis.single.au1.large.32
48	48	10,000/50,000	256/256	100,000	x86: redis.single.xu1.large.48 Arm: redis.single.au1.large.48
64	64	10,000/50,000	384/384	100,000	x86: redis.single.xu1.large.64 Arm: redis.single.au1.large.64

## 主备实例

主备实例有x86和Arm两种CPU类型，支持多副本，副本数支持2到5，如果是Arm类型，即Arm主备2副本、Arm主备3副本……Arm主备5副本。主备实例默认1个主节点、2个副本数（包含主副本）。

同一个内存规格，x86主备实例、Arm主备实例、主备多副本实例有如下差异：

- 可使用内存、连接数上限、基准/最大带宽、QPS：都一样。
- 产品规格编码（实例规格名称）：[表6-2](#)中仅列出了x86和Arm的实例规格名称，如果是其他副本个数，名称中相应修改副本数量，例如，8G规格的Arm主备实例，Arm主备2副本的名称为redis.ha.au1.large.r2.8，3副本为redis.ha.au1.large.r3.8，以此类推。
- 占用IP个数：占用的IP个数=主节点个数\*副本个数。例如：  
主备2副本实例，占用IP个数=1\*2=2；  
主备3副本实例，占用IP个数=1\*3=3。

**表 6-2** Redis 4.0 和 Redis 5.0 主备实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
0.125	0.125	10,000/10,000	40/40	80,000	x86: redis.ha.xu1.tiny.r2.128 Arm: redis.ha.au1.tiny.r2.128
0.25	0.25	10,000/10,000	80/80	80,000	x86: redis.ha.xu1.tiny.r2.256 Arm: redis.ha.au1.tiny.r2.256
0.5	0.5	10,000/10,000	80/80	80,000	x86: redis.ha.xu1.tiny.r2.512 Arm: redis.ha.au1.tiny.r2.512
1	1	10,000/50,000	80/80	80,000	x86: redis.ha.xu1.large.r2.1 Arm: redis.ha.au1.large.r2.1

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
2	2	10,000/50,000	128/128	80,000	x86: redis.ha.xu1.large.r2.2 Arm: redis.ha.au1.large.r2.2
4	4	10,000/50,000	192/192	80,000	x86: redis.ha.xu1.large.r2.4 Arm: redis.ha.au1.large.r2.4
8	8	10,000/50,000	192/192	100,000	x86: redis.ha.xu1.large.r2.8 Arm: redis.ha.au1.large.r2.8
16	16	10,000/50,000	256/256	100,000	x86: redis.ha.xu1.large.r2.16 Arm: redis.ha.au1.large.r2.16
24	24	10,000/50,000	256/256	100,000	x86: redis.ha.xu1.large.r2.24 Arm: redis.ha.au1.large.r2.24
32	32	10,000/50,000	256/256	100,000	x86: redis.ha.xu1.large.r2.32 Arm: redis.ha.au1.large.r2.32

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
48	48	10,000/50,000	256/256	100,000	x86: redis.ha.xu1.large.r2.48  Arm: redis.ha.au1.large.r2.48
64	64	10,000/50,000	384/384	100,000	x86: redis.ha.xu1.large.r2.64  Arm: redis.ha.au1.large.r2.64

## Proxy 集群实例

Proxy集群实例支持x86和Arm两种CPU类型，具体规格如表6-3所示。

Proxy集群实例不支持自定义副本数，每个分片默认为双副本。实例默认的分片数，请参考表5-2，在购买实例时，支持自定义单分片容量。

### 说明

- 下表中仅列举了默认分片的Proxy集群实例规格，当选择自定义分片时，实例规格的最大连接数、基准/最大带宽、产品规格编码（规格名称）请参考控制台购买缓存实例页面的“实例规格”。
- 集群实例的“最大连接数”是实例的最大连接数，单分片的最大连接数=实例最大连接数/分片数。
- 集群实例的“最大带宽/基准带宽”是实例的最大带宽/基准带宽，而不是单个分片的带宽。实例带宽与单分片带宽的关系如下：
  - 实例带宽=单分片带宽\*分片数。
  - 当集群实例单分片内存为1 GB时，单分片带宽为384 Mbit/s，当集群实例单分片内存大于1 GB，单分片带宽为768 Mbit/s。
  - Proxy集群实例的带宽上限为10000 Mbit/s，即当单分片带宽\*分片数大于10000 Mbit/s时，实例带宽均为10000 Mbit/s。

表 6-3 Redis 4.0 和 Redis 5.0 Proxy 集群实例产品规格

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
4	4	3	20,000/20,000	2,304/2,304	240,000	x86: redis.proxy.xu1.l arge.4 Arm: redis.proxy.au1.l arge.4
8	8	3	30,000/30,000	2,304/2,304	240,000	x86: redis.proxy.xu1.l arge.8 Arm: redis.proxy.au1.l arge.8
16	16	3	30,000/30,000	2,304/2,304	240,000	x86: redis.proxy.xu1.l arge.16 Arm: redis.proxy.au1.l arge.16
24	24	3	30,000/30,000	2,304/2,304	240,000	x86: redis.proxy.xu1.l arge.24 Arm: redis.proxy.au1.l arge.24
32	32	3	30,000/30,000	2,304/2,304	240,000	x86: redis.proxy.xu1.l arge.32 Arm: redis.proxy.au1.l arge.32
48	48	6	60,000/60,000	4,608/4,608	480,000	x86: redis.proxy.xu1.l arge.48 Arm: redis.proxy.au1.l arge.48

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的spec_code)
64	64	8	80,000/80,000	6,144/6,144	640,000	x86: redis.proxy.xu1.l arge.64 Arm: redis.proxy.au1.l arge.64
96	96	12	120,000/120,000	9,216/9,216	960,000	x86: redis.proxy.xu1.l arge.96 Arm: redis.proxy.au1.l arge.96
128	128	16	160,000/160,000	10,000/10,000	1,280,000	x86: redis.proxy.xu1.l arge.128 Arm: redis.proxy.au1.l arge.128
192	192	24	200,000/240,000	10,000/10,000	1,920,000	x86: redis.proxy.xu1.l arge.192 Arm: redis.proxy.au1.l arge.192
256	256	32	200,000/320,000	10,000/10,000	>2,000,000	x86: redis.proxy.xu1.l arge.256 Arm: redis.proxy.au1.l arge.256
384	384	48	200,000/480,000	10,000/10,000	>2,000,000	x86: redis.proxy.xu1.l arge.384 Arm: redis.proxy.au1.l arge.384

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的spec_code)
512	512	64	200,000/ 500,000	10,000/ 10,000	>2,000,000	x86: redis.proxy.xu1.l arge.512 Arm: redis.proxy.au1.l arge.512
768	768	96	200,000/ 500,000	10,000/ 10,000	>2,000,000	x86: redis.proxy.xu1.l arge.768 Arm: redis.proxy.au1.l arge.768
1024	1024	128	200,000/ 500,000	10,000/ 10,000	>2,000,000	x86: redis.proxy.xu1.l arge.1024 Arm: redis.proxy.au1.l arge.1024
2048	2048	128	200,000/ 500,000	10,000/ 10,000	>2,000,000	x86: redis.proxy.xu1.l arge.2048 Arm: redis.proxy.au1.l arge.2048
4096	4096	128	200,000/ 500,000	10,000/ 10,000	>2,000,000	x86: redis.proxy.xu1.l arge.2048 Arm: redis.proxy.au1.l arge.2048

## Cluster 集群实例

Cluster实例有x86和Arm两种CPU类型，支持多副本，副本个数支持1到5，即，Cluster 1副本、Cluster 2副本……Cluster 5副本。Cluster集群实例默认副本数为2，Cluster 1副本实例则表示减少副本数。

同一个内存规格，x86 Cluster集群实例、Arm Cluster集群实例、Cluster集群多副本实例有如下差异：

- 可使用内存、分片数（主节点个数）、连接数上限、基准/最大带宽、QPS：都一样。
- 实例规格名称：[表6-4](#)中仅列出了x86和Arm 2副本的实例规格名称，如果是其他副本个数，名称中相应修改副本数量。例如，24G规格的Arm 2副本实例的名称为redis.cluster.au1.large.r2.24，3副本为redis.cluster.au1.large.r3.24，以此类推。
- **占用IP个数**：占用的IP个数=分片数\*副本个数。例如：  
24G规格的Cluster 3副本实例，占用IP个数=3\*3=9。
- **单个节点可使用内存**：单个节点可使用内存=实例可使用内存/主节点个数。例如：  
24G规格x86实例，实例可使用内存为24G，主节点个数为3，则单个节点可使用内存=24/3=8G。
- **单个节点连接数上限**：单个节点连接数上限=实例连接数上限/主节点个数。例如：  
24G规格x86实例，实例连接数上限为150000，主节点个数为3，则单个节点连接数上限=150000/3=50000个。

#### 📖 说明

- 下表中仅列举了默认分片的Cluster集群实例规格，当选择自定义分片时，实例规格的最大连接数、基准/最大带宽、产品规格编码（规格名称）请参考控制台购买缓存实例页面的“实例规格”。
- 集群实例的“最大连接数”是实例的最大连接数，单分片的最大连接数=实例最大连接数/分片数。
- 集群实例的“最大带宽/基准带宽”是实例的最大带宽/基准带宽，而不是单个分片的带宽。实例带宽与单分片带宽的关系如下：
  - 实例带宽=单分片带宽\*分片数。
  - 当集群实例单分片内存为1 GB时，单分片带宽为384 Mbit/s，当集群实例单分片内存大于1 GB，单分片带宽为768 Mbit/s。

**表 6-4** Redis 4.0 和 Redis 5.0 Cluster 集群实例产品规格

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
4	4	3	30,000 / 150,000	2,304/2,304	240,000	x86: redis.cluster.xu1.large.r2.4  Arm: redis.cluster.au1.large.r2.4



规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
8	8	3	30,000 / 150,000	2,304/2,304	240,000	x86: redis.cluster.xu1.large.r2.8 Arm: redis.cluster.au1.large.r2.8
16	16	3	30,000 / 150,000	2,304/2,304	240,000	x86: redis.cluster.xu1.large.r2.16 Arm: redis.cluster.au1.large.r2.16
24	24	3	30,000 / 150,000	2,304/2,304	300,000	x86: redis.cluster.xu1.large.r2.24 Arm: redis.cluster.au1.large.r2.24
32	32	3	30,000 / 150,000	2,304/2,304	300,000	x86: redis.cluster.xu1.large.r2.32 Arm: redis.cluster.au1.large.r2.32
48	48	6	60,000 / 300,000	4,608/4,608	>300,000	x86: redis.cluster.xu1.large.r2.48 Arm: redis.cluster.au1.large.r2.48
64	64	8	80,000 / 400,000	6,144/6,144	500,000	x86: redis.cluster.xu1.large.r2.64 Arm: redis.cluster.au1.large.r2.64

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
96	96	12	120,000 / 600,000	9,216/9,216	>500,000	x86: redis.cluster.xu1.large.r2.96 Arm: redis.cluster.au1.large.r2.96
128	128	16	160,000 / 800,000	12,288/12,288	1,000,000	x86: redis.cluster.xu1.large.r2.128 Arm: redis.cluster.au1.large.r2.128
192	192	24	240,000 / 1,200,000	18,432/18,432	>1,000,000	x86: redis.cluster.xu1.large.r2.192 Arm: redis.cluster.au1.large.r2.192
256	256	32	320,000 / 1,600,000	24,576/24,576	>2,000,000	x86: redis.cluster.xu1.large.r2.256 Arm: redis.cluster.au1.large.r2.256
384	384	48	480,000 / 2,400,000	36,864/36,864	>2,000,000	x86: redis.cluster.xu1.large.r2.384 Arm: redis.cluster.au1.large.r2.384
512	512	64	640,000 / 3,200,000	49,152/49,152	>2,000,000	x86: redis.cluster.xu1.large.r2.512 Arm: redis.cluster.au1.large.r2.512

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	实例最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
768	768	96	960,000 / 4,800,000	73,728/73,728	>2,000,000	x86: redis.cluster.xu1.large.r2.768 Arm: redis.cluster.au1.large.r2.768
1024	1024	128	1,280,000 / 6,400,000	98,304/98,304	>2,000,000	x86: redis.cluster.xu1.large.r2.1024 Arm: redis.cluster.au1.large.r2.1024

## 读写分离实例

- 读写分离实例当前只支持x86 CPU类型，具体规格如表6-5。
- 读写分离实例的连接数限制暂不支持修改。
- 单个数据节点带宽限制(MB/s)=总带宽限制(MB/s)/副本数(主+从)。
- 单个节点参考性能(QPS)=参考性能(QPS)/副本数(主+从)。
- 读写分离实例使用限制：
  - a. 读写分离实例读请求会发送到从节点，从节点从主节点同步数据会有一些的时延。  
请确保业务侧不依赖主从同步的时延，如果对主从同步时延有依赖的场景，不适用读写分离实例，请考虑普通主备或集群。
  - b. 读写分离实例适用于写少读多的场景，如果写流量过大，可能导致主从断连，或断连后主从同步失败，导致读请求性能下降。  
写流量大的场景请考虑普通主备或集群。
  - c. 从节点故障后，需要一定的时间从主节点全量同步数据，同步数据期间，从节点不对外提供服务，此时实例读请求性能会下降。  
推荐使用<32G内存规格的实例，内存规格越小，主从全量同步数据时间越少，同步断链时间越短。

表 6-5 Redis 4.0 和 Redis 5.0 读写分离实例产品规格

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
8	8	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.8
8	8	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.8
8	8	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.8
8	8	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.8
8	8	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.8
16	16	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.16
16	16	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.16
16	16	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.16
16	16	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.16

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
16	16	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.16
32	32	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.32
32	32	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.32
32	32	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.32
32	32	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.32
32	32	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.32
64	64	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.64
64	64	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.64
64	64	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.64

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
64	64	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.64
64	64	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.64

## 6.2 Redis 6.0 实例

本节介绍DCS Redis 6.0实例的产品规格，包括内存规格、实例可使用内存、连接数上限、最大带宽/基准带宽、参考性能（QPS）等。

实例各项指标如下：

- 实例已使用内存：您可以通过查看监控指标“内存利用率”和“已用内存”查看实例内存使用情况。
- 最大连接数：表示允许客户端同时连接的个数，即连接并发数。具体实例的连接数，可查看监控指标“活跃的客户数量”。
- QPS：即Query Per Second，表示数据库每秒执行的命令数。QPS的测试方法，可参考[性能测试方法](#)。
- 带宽：即实例的内网带宽，您可以查看监控指标“流控次数”，确认带宽是否超过限额。其中，带宽还涉及监控指标“带宽使用率”，该指标仅供参考，可能会出现超过100%的情况，具体原因，可查看[带宽使用率指标会超过100%的原因](#)。

Redis 6.0实例目前包含“基础版”、“企业版高性能型”和“企业版存储型”，目前仅在北京四和华南广州等部分Region上线。Redis 6.0的实例架构请参考对应的[实例类型](#)。

Redis 6.0企业版实例目前只支持主备实例，CPU类型为X86结构。

### 📖 说明

不同区域支持的规格可能存在差异，请以控制台可选择的实际规格为准。

## 基础版主备实例

表 6-6 Redis 6.0 基础版主备实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
0.125	0.125	10,000/10,000	40/40	80,000	redis.ha.xu1.tiny.r2.128
0.25	0.25	10,000/10,000	80/80	80,000	redis.ha.xu1.tiny.r2.256
0.5	0.5	10,000/10,000	80/80	80,000	redis.ha.xu1.tiny.r2.512
1	1	10,000/50,000	80/80	80,000	redis.ha.xu1.large.r2.1
2	2	10,000/50,000	128/128	80,000	redis.ha.xu1.large.r2.2
4	4	10,000/50,000	192/192	80,000	redis.ha.xu1.large.r2.4
8	8	10,000/50,000	192/192	100,000	redis.ha.xu1.large.r2.8
16	16	10,000/50,000	256/256	100,000	redis.ha.xu1.large.r2.16
24	24	10,000/50,000	256/256	100,000	redis.ha.xu1.large.r2.24
32	32	10,000/50,000	256/256	100,000	redis.ha.xu1.large.r2.32
48	48	10,000/50,000	256/256	100,000	redis.ha.xu1.large.r2.48
64	64	10,000/50,000	384/384	100,000	redis.ha.xu1.large.r2.64

## 基础版单机实例

表 6-7 Redis 6.0 基础版单机实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
0.125	0.125	10,000/10,000	40/40	80,000	redis.single.xu1.tiny.128
0.25	0.25	10,000/10,000	80/80	80,000	redis.single.xu1.tiny.256
0.5	0.5	10,000/10,000	80/80	80,000	redis.single.xu1.tiny.512
1	1	10,000/50,000	80/80	80,000	redis.single.xu1.large.1
2	2	10,000/50,000	128/128	80,000	redis.single.xu1.large.2
4	4	10,000/50,000	192/192	80,000	redis.single.xu1.large.4
8	8	10,000/50,000	192/192	100,000	redis.single.xu1.large.8
16	16	10,000/50,000	256/256	100,000	redis.single.xu1.large.16
24	24	10,000/50,000	256/256	100,000	redis.single.xu1.large.24
32	32	10,000/50,000	256/256	100,000	redis.single.xu1.large.32
48	48	10,000/50,000	256/256	100,000	redis.single.xu1.large.48
64	64	10,000/50,000	384/384	100,000	redis.single.xu1.large.64

## 基础版 Proxy 集群实例

Proxy 集群实例支持的具体规格如[表6-8](#)所示。

Proxy 集群实例不支持自定义副本数，每个分片默认为双副本。实例默认的分片数，请参考[表5-2](#)，在购买实例时，支持自定义单分片容量。



 说明

- 下表中仅列举了默认分片的Proxy集群实例规格，当选择自定义分片时，实例规格的最大连接数、基准/最大带宽、产品规格编码（规格名称）请参考控制台购买缓存实例页面的“实例规格”。
- 集群实例的“最大连接数”是实例的最大连接数，单分片的最大连接数=实例最大连接数/分片数。
- 集群实例的“最大带宽/基准带宽”是实例的最大带宽/基准带宽，而不是单个分片的带宽。实例带宽与单分片带宽的关系如下：
  - 实例带宽=单分片带宽\*分片数。
  - 当集群实例单分片内存为1 GB时，单分片带宽为384 Mbit/s，当集群实例单分片内存大于1 GB，单分片带宽为768 Mbit/s。
  - Proxy集群实例的带宽上限为10000 Mbit/s，即当单分片带宽\*分片数大于10000 Mbit/s时，实例带宽均为10000 Mbit/s。

表 6-8 Redis 6.0 基础版 Proxy 集群实例产品规格

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
4	4	3	20,000/20,000	2,304/2,304	240,000	redis.proxy.xu1.large.4
8	8	3	30,000/30,000	2,304/2,304	240,000	redis.proxy.xu1.large.8
16	16	3	30,000/30,000	2,304/2,304	240,000	redis.proxy.xu1.large.16
24	24	3	30,000/30,000	2,304/2,304	240,000	redis.proxy.xu1.large.24
32	32	3	30,000/30,000	2,304/2,304	240,000	redis.proxy.xu1.large.32
48	48	6	60,000/60,000	4,608/4,608	480,000	redis.proxy.xu1.large.48
64	64	8	80,000/80,000	6,144/6,144	640,000	redis.proxy.xu1.large.64
96	96	12	120,000/120,000	9,216/9,216	960,000	redis.proxy.xu1.large.96
128	128	16	160,000/160,000	10,000/10,000	1,280,000	redis.proxy.xu1.large.128
192	192	24	200,000/240,000	10,000/10,000	1,920,000	redis.proxy.xu1.large.192
256	256	32	200,000/320,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.256

规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
384	384	48	200,000/480,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.384
512	512	64	200,000/500,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.512
768	768	96	200,000/500,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.768
1024	1024	128	200,000/500,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.1024
2048	2048	128	200,000/500,000	10,000/10,000	>2,000,000	redis.proxy.xu1.large.2048

## 基础版 Cluster 集群实例

- 产品规格编码 (实例规格名称)：表6-9中仅列出了默认2副本的实例规格名称，如果是副本个数为1时，名称中相应修改副本数量。例如，4G规格的2副本实例的规格名称为redis.cluster.xu1.large.r2.4，副本数为1时，规格名称为redis.cluster.xu1.large.r1.4，其他规格以此类推。
- 占用IP个数：占用的IP个数=分片数\*副本个数。例如：24G规格的Cluster 2副本实例，占用IP个数=3\*2=6。
- 单个节点可使用内存：单个节点可使用内存=实例可使用内存/主节点个数。例如：24G规格实例，实例可使用内存为24G，主节点个数为3，则单个节点可使用内存=24/3=8G。
- 单个节点连接数上限：单个节点连接数上限=实例连接数上限/主节点个数。例如：24G规格实例，实例连接数上限为150000，主节点个数为3，则单个节点连接数上限=150000/3=50000个。

### 说明

- 下表中仅列举了默认分片的Cluster集群实例规格，当选择自定义分片时，实例规格的最大连接数、基准/最大带宽、产品规格编码 (规格名称) 请参考控制台购买缓存实例页面的“实例规格”。
- 集群实例的“最大连接数”是实例的最大连接数，单分片的最大连接数=实例最大连接数/分片数。
- 集群实例的“最大带宽/基准带宽”是实例的最大带宽/基准带宽，而不是单个分片的带宽。实例带宽与单分片带宽的关系如下：
  - 实例带宽=单分片带宽\*分片数。
  - 当集群实例单分片内存为1 GB时，单分片带宽为384 Mbit/s，当集群实例单分片内存大于1 GB，单分片带宽为768 Mbit/s。

表 6-9 Redis 6.0 基础版 Cluster 集群实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	分片数 (主节点个数)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
4	4	3	30,000/150,000	2,304/2,304	240,000	redis.cluster.xu1.l arge.r2.4
8	8	3	30,000/150,000	2,304/2,304	240,000	redis.cluster.xu1.l arge.r2.8
16	16	3	30,000/150,000	2,304/2,304	240,000	redis.cluster.xu1.l arge.r2.16
24	24	3	30,000/150,000	2,304/2,304	300,000	redis.cluster.xu1.l arge.r2.24
32	32	3	30,000/150,000	2,304/2,304	300,000	redis.cluster.xu1.l arge.r2.32
48	48	6	60,000/300,000	4,608/4,608	>300,000	redis.cluster.xu1.l arge.r2.48
64	64	8	80,000/400,000	6,144/6,144	500,000	redis.cluster.xu1.l arge.r2.64
96	96	12	120,000/600,000	9,216/9,216	>500,000	redis.cluster.xu1.l arge.r2.96
128	128	16	160,000/800,000	12,288/12,288	1,000,000	redis.cluster.xu1.l arge.r2.128
192	192	24	240,000/1,200,000	18,432/18,432	>1,000,000	redis.cluster.xu1.l arge.r2.192
256	256	32	320,000/1,600,000	24,576/24,576	>2,000,000	redis.cluster.xu1.l arge.r2.256
384	384	48	480,000/2,400,000	36,864/36,864	>2,000,000	redis.cluster.xu1.l arge.r2.384
512	512	64	640,000/3,200,000	49,152/49,152	>2,000,000	redis.cluster.xu1.l arge.r2.512
768	768	96	960,000/4,800,000	73,728/73,728	>2,000,000	redis.cluster.xu1.l arge.r2.768
1024	1024	128	1,280,000/6,400,000	98,304/98,304	>2,000,000	redis.cluster.xu1.l arge.r2.1024
2048	2048	128	2,560,000/12,800,000	98,304/98,304	>2,000,000	redis.cluster.xu1.l arge.r2.2048

## 基础版读写分离实例

- 读写分离实例具体规格如表6-10。
- 读写分离实例的连接数限制暂不支持修改。
- 单个数据节点带宽限制(MB/s)=总带宽限制(MB/s)/副本数(主+从)。
- 单个节点参考性能(QPS)=参考性能(QPS)/副本数(主+从)。
- 读写分离实例使用限制：
  - a. 主备读写分离实例读请求会发送到从节点，从节点从主节点同步数据会有一些的时延。  
请确保业务侧不依赖主从同步的时延，如果对主从同步时延有依赖的场景，不适用读写分离实例，请考虑普通主备或集群。
  - b. 主备读写分离实例适用于写少读多的场景，如果写流量过大，可能导致主从断连，或断连后主从同步失败，导致读请求性能下降。  
写流量大的场景请考虑普通主备或集群。
  - c. 从节点故障后，需要一定的时间从主节点全量同步数据，同步数据期间，从节点不对外提供服务，此时实例读请求性能会下降。  
推荐使用<32G内存规格的实例，内存规格越小，主从全量同步数据时间越少，同步断链时间越短。

表 6-10 Redis 6.0 读写分离实例产品规格

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
8	8	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.8
8	8	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.8
8	8	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.8
8	8	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.8

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
8	8	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.8
16	16	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.16
16	16	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.16
16	16	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.16
16	16	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.16
16	16	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.16
32	32	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.32
32	32	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.32
32	32	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.32

规格	实例可使用内存(GB)	副本数(主+从)	最大连接数(默认/最大可配)	总带宽限制(MB/s)	单个数据节点带宽限制(MB/s)	参考性能(QPS)	单个节点参考性能(QPS)	产品规格编码(对应API的spec_code)
32	32	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.32
32	32	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.32
64	64	2	20,000	192	96	160,000	80,000	redis.ha.xu1.large.p2.64
64	64	3	30,000	288	96	240,000	80,000	redis.ha.xu1.large.p3.64
64	64	4	40,000	384	96	320,000	80,000	redis.ha.xu1.large.p4.64
64	64	5	50,000	480	96	400,000	80,000	redis.ha.xu1.large.p5.64
64	64	6	60,000	576	96	480,000	80,000	redis.ha.xu1.large.p6.64

## 企业版高性能型

Redis 6.0企业版高性能型实例目前只支持主备实例类型，CPU类型为X86结构。

表 6-11 Redis 6.0 企业版高性能型实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
8	8	10,000/50,000	1536/1536	400,000	redis.ha.xu1.large.enthp.8
16	16	10,000/50,000	1536/1536	400,000	redis.ha.xu1.large.enthp.16
32	32	10,000/50,000	1536/1536	400,000	redis.ha.xu1.large.enthp.32
64	64	10,000/50,000	1536/1536	400,000	redis.ha.xu1.large.enthp.64

## 企业版存储型

Redis 6.0企业版存储型实例目前只支持主备实例类型，CPU类型为X86结构。

企业版存储型由内存和SSD磁盘组成，使用内存缓存热数据，全量数据存储于SSD磁盘。下表中的“最大存储容量”即磁盘容量。

表 6-12 Redis 6.0 企业版存储型实例产品规格

内存规格 (GB)	最大存储容量 (GB)	最大连接数 (默认/最大可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
8	64	10000/50000	768/768	70,000	redis.ha.xu1.large.entst.8
16	128	10000/50000	768/768	70,000	redis.ha.xu1.large.entst.16
32	256	10000/50000	768/768	70,000	redis.ha.xu1.large.entst.32

## 6.3 Redis 3.0 实例 (已停售)

本节介绍DCS Redis 3.0实例的产品规格，包括内存规格、实例可使用内存、连接数上限、最大带宽/基准带宽、参考性能 (QPS) 等。

实例各项指标如下：

- 实例已使用内存：您可以通过查看监控指标“内存利用率”和“已用内存”查看实例内存使用情况。

- 连接数上限：表示允许客户端同时连接的个数，即连接并发数。具体实例的连接数，可查看监控指标“活跃的客户数量”。
- QPS：即Query Per Second，表示数据库每秒执行的命令数。

**说明**

- Redis 3.0实例支持“单机”、“主备”和“Proxy集群”三种类型。
- DCS Redis 3.0已暂停售卖，建议使用Redis 4.0/5.0/6.0。

## 单机实例

因系统开销占用一部分资源，Redis单机实例可用内存比实例规格略小，如表6-13所示。

表 6-13 Redis 3.0 单机实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
2	1.5	5,000/50,000	42/512	50,000	dcs.single_node
4	3.2	5,000/50,000	64/1,536	100,000	dcs.single_node
8	6.8	5,000/50,000	64/1,536	100,000	dcs.single_node
16	13.6	5,000/50,000	85/3,072	100,000	dcs.single_node
32	27.2	5,000/50,000	85/3,072	100,000	dcs.single_node
64	58.2	5,000/60,000	128/5,120	100,000	dcs.single_node

## 主备实例

对于Redis主备实例，需要预留持久化的内存，部分规格的实际可使用与单机实例相比略少，如表6-14所示。主备实例可以调整实例可用内存，以更好地支持数据持久化、主从同步等后台任务。



表 6-14 Redis 3.0 主备实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
2	1.5	5,000/50,000	42/512	50,000	dcs.master_standby
4	3.2	5,000/50,000	64/1,536	100,000	dcs.master_standby
8	6.4	5,000/50,000	64/1,536	100,000	dcs.master_standby
16	12.8	5,000/50,000	85/3,072	100,000	dcs.master_standby
32	25.6	5,000/50,000	85/3,072	100,000	dcs.master_standby
64	51.2	5,000/60,000	128/5,120	100,000	dcs.master_standby

## Proxy 集群实例

Redis Proxy集群实例与单机、主备实例的区别，不仅在于支持高规格内存，客户端连接数、内网带宽上限、QPS指标都有很大的提升。

表 6-15 Redis 3.0 Proxy 集群实例产品规格

规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
64	64	90,000/90,000	600/5,120	500,000	dcs.cluster
128	128	180,000/180,000	600/5,120	500,000	dcs.cluster
256	256	240,000/240,000	600/5,120	500,000	dcs.cluster
512	512	480,000/480,000	600/5,120	500,000	dcs.cluster
1024	1024	960,000/960,000	600/5,120	500,000	dcs.cluster

 说明

- 按需付费的Redis Proxy集群实例，支持的实例规格为64GB、128GB和256GB；
- 包年/包月付费的Redis Proxy集群实例，支持的实例规格为64GB、128GB、256GB、512GB和1024GB。

目前支持包年/包月付费方式的Region为中国-香港，若其他Region也需使用此功能，请您在控制台提交工单，技术人员会在后台为您进行开通。

## 6.4 Memcached 实例（已停售）

 说明

DCS Memcached已停售，建议使用Redis实例。

本节介绍DCS Memcached缓存实例的产品规格，包括内存规格、实例可使用内存、连接数上限、最大带宽/基准带宽、参考性能（QPS）等。

连接数上限：表示允许客户端同时连接的个数，即连接并发数。具体实例的连接数，可查看监控指标“活跃的客户数量”。

QPS：即Query Per Second，表示数据库每秒执行的命令数。

 说明

Memcached实例支持“单机”和“主备”两种类型。

### 单机实例

因系统开销占用一部分资源，Memcached单机实例可用内存比实例规格略小，如表6-16所示。

表 6-16 Memcached 单机实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
2	1.5	5,000/50,000	42/128	50,000	dcs.memcached.single_node
4	3.2	5,000/50,000	64/192	100,000	dcs.memcached.single_node
8	6.8	5,000/50,000	64/192	100,000	dcs.memcached.single_node
16	13.6	5,000/50,000	85/256	100,000	dcs.memcached.single_node
32	27.2	5,000/50,000	85/256	100,000	dcs.memcached.single_node

内存规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
64	58.2	5,000/50,000	128/384	100,000	dcs.memcached.single_node

## 主备实例

对于Memcached主备实例，需要预留持久化的内存，其可用内存如表6-17所示。主备实例可以调整实例可用内存，以更好地支持数据持久化、主从同步等后台任务。

表 6-17 Memcached 主备实例产品规格

内存规格 (GB)	实例可使用内存 (GB)	连接数上限 (默认/可配) (个)	基准/最大带宽 (Mbit/s)	参考性能 (QPS)	产品规格编码 (对应API的 spec_code)
2	1.5	5,000/50,000	42/128	50,000	dcs.memcached.master_standby
4	3.2	5,000/50,000	64/192	100,000	dcs.memcached.master_standby
8	6.8	5,000/50,000	64/192	100,000	dcs.memcached.master_standby
16	13.6	5,000/50,000	85/256	100,000	dcs.memcached.master_standby
32	27.2	5,000/50,000	85/256	100,000	dcs.memcached.master_standby
64	58.2	5,000/50,000	128/384	100,000	dcs.memcached.master_standby

# 7 开源命令兼容性

## 7.1 Redis 4.0 支持及禁用的命令

华为云DCS Redis 4.0基于开源4.0.14版本进行开发，兼容开源的协议和命令。本章节主要介绍DCS Redis 4.0命令的兼容性，包括支持命令列表，禁用命令列表。

DCS Redis缓存实例支持Redis的绝大部分命令，任何兼容Redis协议的客户端都可以访问DCS。

- 因安全原因，部分Redis命令在分布式缓存服务中被禁用，具体请见[Redis 4.0禁用的命令](#)。
- DCS集群实例支持多个key，但不支持跨slot访问的Redis命令列表，如[实例受限使用命令](#)所示。
- 部分Redis命令使用时有限制，例如KEYS、FLUSHDB、FLUSHALL等，具体请见[部分命令使用限制](#)。
- 部分高危命令支持命令重命名，具体请见[支持重命名的命令](#)。

### Redis 4.0 支持的命令

- [表7-1](#)和[表7-2](#)列举了华为云Redis 4.0单机/主备/Cluster集群实例支持的Redis命令。
- [表7-3](#)和[表7-4](#)列举了华为云Redis 4.0 Proxy集群支持的Redis命令。
- [表7-5](#)和[表7-6](#)列举了华为云Redis 4.0读写分离支持的Redis命令。

各个命令的具体详细语法请前往[Redis官方网站](#)查看，例如您想了解SCAN命令的使用，可在[Redis官方网站](#)中搜索框中输入“SCAN”查询详细介绍。

#### 📖 说明

- Redis高版本的命令，在低版本中不被兼容。判断DCS Redis是否支持某个命令，可通过在Redis-cli执行该命令，如果得到 ( error ) ERR unknown command 'xxx' 的提示，则说明不支持该命令。
- Redis 4.0 Cluster版本集群实例使用pipeline时，要确保管道中的命令都能在同一分片执行。

表 7-1 Redis 4.0 单机/主备/Cluster 集群支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	CLIENT KILL
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	CLIENT LIST
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGE BYRANK	CLIENT GETNAME
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGE BYCORE	CLIENT SETNAME
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	CONFIG GET
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGE BYSCORE	MONITOR
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	SLOWLOG
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	ROLE
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	SWAPDB
SCAN	MSETNX	HLEN	RPOPLPUSH	SSCAN	ZINTERSTORE	MEMORY
OBJECT	PSETEX	-	RPUSH	-	ZSCAN	CONFIG
PEXPIRE	SET	-	RPUSHX	-	ZRANGEBYLEX	COMMAND
PEXPIREAT	SETBIT	-	LPUSH	-	ZLEXCOUNT	-

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
KEYS	SETEX	-	-	-	ZREMRANGE BYSCORE	-
-	SETNX	-	-	-	ZREM	-
-	SETRANGE	-	-	-	-	-
-	STRLEN	-	-	-	-	-
-	BITFIELD	-	-	-	-	-

表 7-2 Redis 4.0 单机/主备/Cluster 集群支持的命令清单 2

HyperLogLog	Pub/Sub	Transactions	Connection	Scripting	Geo
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST
-	SUBSCRIBE	WATCH	SELECT (Cluster 集群实例不支持)	SCRIPT KILL	GEORADIUS
-	UNSUBSCRIBE	-	-	SCRIPT LOAD	GEORADIUSBYMEMBER

表 7-3 Redis 4.0 proxy 集群支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DUMP	BITCOUNT	HEXISTS	BRPOPLRUSH	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	ROLE
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	MEMORY
RENAME	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	COMMAND
RENAME NX	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	COMMAND COUNT
RESTORE	INCR	HSET	LREM	SPOP	ZREVRANGE	COMMAND GETKEYS
SORT	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND INFO
TTL	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	CONFIG GET
TYPE	MGET	HSCAN	RPOP	SUNION	ZSCORE	CONFIG RESETSTAT
SCAN	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	CONFIG REWRITE
OBJECT	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG SET
PEXPIRE	PSETEX	HKEYS	RPUSHX	-	ZSCAN	-
PEXPIREAT	SET	-	LPUSH	-	ZRANGEBYLEX	-
EXPIREAT	SETBIT	-	-	-	ZLEXCOUNT	-

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
KEYS	SETEX	-	-	-	ZREMRA NGBYSC ORE	-
TOUCH	SETNX	-	-	-	ZREM	-
UNLINK	SETRAN GE	-	-	-	ZREMRA NGBYLE X	-
RANDO MKEY	STRLEN	-	-	-	ZREVRA NGBYLE X	-
-	BITFIELD	-	-	-	-	-
-	GETBIT	-	-	-	-	-

表 7-4 Redis 4.0 proxy 集群支持的命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connect ion	Scripting	Geo	Cluster
PFADD	PSUBSC RIBE	DISCAR D	AUTH	EVAL	GEOADD	CLUSTER INFO
PFCOU NT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	CLUSTER NODES
PFMER GE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	CLUSTER SLOTS
-	PUNSUB SCRIBE	UNWAT CH	QUIT	SCRIPT FLUSH	GEODIST	CLUSTER ADDSLOTS
-	SUBSCRI BE	WATCH	SELECT	SCRIPT KILL	GEORADIU S	ASKING
-	UNSUBS CRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADIU SBYMEMBE R	READONL Y
-	-	-	CLIENT LIST	SCRIPT DEBUG YES  SYNC NO	GEOSEARC H	READWRI TE
-	-	-	CLIENT GETNAM E	-	GEOSEARC HSTORE	-



HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo	Cluster
-	-	-	CLIENT SETNAME	-	-	-

 说明

上表中的Cluster类命令，仅2022/9/1日及之后创建的proxy集群实例支持。

表 7-5 Redis 4.0 读写分离支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	MONITOR
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	SLOWLOG
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	ROLE
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	SWAPDB
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	MEMORY
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	COMMAND COUNT
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	COMMAND GETKEYS
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNION STORE	ZUNION STORE	COMMAND INFO
SCAN	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG GET
OBJECT	PSETEX	-	RPUSHX	-	ZSCAN	CONFIG RESETSTAT
PEXPIRE	SET	-	LPUSH	-	ZRANGEBYLEX	CONFIG REWRITE
PEXPIREAT	SETBIT	-	-	-	ZLEXCOUNT	CONFIG SET
EXPIREAT	SETEX	-	-	-	ZREMRANGEBYSCORE	-
KEYS	SETNX	-	-	-	ZREM	-
TOUCH	SETRANGE	-	-	-	ZREMRANGEBYLEX	-
UNLINK	STRLEN	-	-	-	ZREMRANGEBYLEX	-
-	BITFIELD	-	-	-	-	-
-	GETBIT	-	-	-	-	-

表 7-6 Redis 4.0 读写分离支持的命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH

HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST
-	SUBSCRIBE	WATCH	SELECT	SCRIPT KILL	GEORADIUS
-	UNSUBSCRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADIUSBYMEMBER
-	-	-	CLIENT LIST	SCRIPT DEBUG YES SYNC NO	GEOSEARCH
-	-	-	CLIENT GETNAME	-	GEOSEARCHSTORE
-	-	-	CLIENT SETNAME	-	-

## Redis 4.0 禁用的命令

以下列出了Redis 4.0实例禁用的命令。

表 7-7 Redis 4.0 单机和主备禁用命令

Generic (Key)	Server
MIGRATE	SLAVEOF
-	SHUTDOWN
-	LASTSAVE
-	DEBUG相关类
-	SAVE
-	BGSAVE
-	BGREWRITEAOF
-	SYNC
-	PSYNC

表 7-8 Redis 4.0 Proxy 集群实例禁用命令

Generic (Key)	Server	Sorted Set
MIGRATE	BGREWRITEAOF	BZPOPMAX
MOVE	BGSAVE	BZPOPMIN
WAIT	CLIENT相关命令	ZPOPMAX
-	DEBUG OBJECT	ZPOPMIN
-	DEBUG SEGFAULT	-
-	LASTSAVE	-
-	PSYNC	-
-	SAVE	-
-	SHUTDOWN	-
-	SLAVEOF	-
-	LATENCY相关命令	-
-	MODULE相关命令	-
-	LOLWUT	-
-	SWAPDB	-
-	REPLICAOF	-
-	SYNC	-

表 7-9 Redis 4.0 Cluster 集群禁用命令

Generic (Key)	Server	Cluster
MIGRATE	SLAVEOF	CLUSTER MEET
-	SHUTDOWN	CLUSTER FLUSHSLOTS
-	LASTSAVE	CLUSTER ADDSLOTS
-	DEBUG相关类	CLUSTER DELSLOTS
-	SAVE	CLUSTER SETSLOT
-	BGSAVE	CLUSTER BUMPEPOCH
-	BGREWRITEAOF	CLUSTER SAVECONFIG
-	SYNC	CLUSTER FORGET
-	PSYNC	CLUSTER REPLICATE

Generic (Key)	Server	Cluster
-	-	CLUSTER COUNT-FAILURE-REPORTS
-	-	CLUSTER FAILOVER
-	-	CLUSTER SET-CONFIG-EPOCH
-	-	CLUSTER RESET

表 7-10 Redis 4.0 读写分离禁用命令

Generic	Server	Sorted Set
MIGRATE	BGREWRITEAOF	BZPOPMAX
WAIT	BGSAVE	BZPOPMIN
-	DEBUG OBJECT	ZPOPMAX
-	DEBUG SEGFAULT	ZPOPMIN
-	LASTSAVE	-
-	LOLWUT	-
-	MODULE LIST/LOAD/ UNLOAD	-
-	PSYNC	-
-	REPLICAOF	-
-	SAVE	-
-	SHUTDOWN [NOSAVE  SAVE]	-
-	SLAVEOF	-
-	SWAPDB	-
-	SYNC	-

## 支持重命名的命令

表 7-11 支持重命名的命令

支持重命名的命令	command、keys、flushdb、flushall、hgetall、scan、hscan、sscan、和zscan Proxy集群实例还支持dbsize和dbstats命令重命名，其他命令暂时不支持。
----------	---

重命名的方式	请参考 <a href="#">命令重命名</a> 。
--------	-----------------------------

## 7.2 Redis 5.0 支持及禁用的命令

华为云DCS Redis 5.0基于开源5.0.9版本进行开发，兼容开源的协议和命令。本章节主要介绍DCS Redis 5.0命令的兼容性，包括支持命令列表，禁用命令列表。

DCS Redis缓存实例支持Redis的绝大部分命令，任何兼容Redis协议的客户端都可以访问DCS。

- 因安全原因，部分Redis命令在分布式缓存服务中被禁用，具体请见[Redis 5.0禁用的命令](#)。
- DCS集群实例支持多个key，但不支持跨slot访问的Redis命令列表，如[实例受限使用命令](#)所示。
- 部分Redis命令使用时有限制，例如KEYS、FLUSHDB、FLUSHALL等，具体请见[部分命令使用限制](#)。
- 部分高危命令支持命令重命名，具体请见[支持重命名的命令](#)。

### Redis 5.0 支持的命令

- [表7-12](#)和[表7-13](#)列举了Redis 5.0单机/主备/Cluster集群实例支持的命令。
- [表7-14](#)和[表7-15](#)列举了Redis 5.0 proxy集群支持的命令。
- [表7-16](#)和[表7-17](#)列举了Redis 5.0读写分离支持的命令。

各个命令的具体详细语法请前往[Redis官方网站](#)查看，例如您想了解SCAN命令的使用，可在[Redis官方网站](#)中搜索框中输入“SCAN”查询详细介绍。

#### 📖 说明

- Redis高版本的命令，在低版本中不被兼容。判断DCS Redis是否支持某个命令，可通过在Redis-cli执行该命令，如果得到 ( error ) ERR unknown command ‘xxx’ 的提示，则说明不支持该命令。
- Redis 5.0 Cluster版本集群实例使用pipeline时，要确保管道中的命令都能在同一分片执行。

表 7-12 Redis 5.0 单机/主备/Cluster 集群实例支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	KEYS
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	CLIENT KILL
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	CLIENT LIST
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	CLIENT GETNAME
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	CLIENT SETNAME
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	CONFIG GET
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	MONITOR
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	SLOWLOG
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	ROLE
SCAN	MSETNX	HLEN	RPOPLPUSH	SSCAN	ZINTERSTORE	SWAPDATABASE
OBJECT	PSETEX	-	RPUSH	-	ZSCAN	MEMORY
PEXPIREAT	SET	-	RPUSHX	-	ZRANGEBYLEX	CONFIG
PEXPIRE	SETBIT	-	LPUSH	-	ZLEXCOUNT	COMMAND
KEYS	SETEX	-	-	-	ZPOPMIN	-
-	SETNX	-	-	-	ZPOPMAX	-
-	SETRANGE	-	-	-	ZREMRANGEBYSCORE	-
-	STRLEN	-	-	-	ZREM	-
-	BITFIELD	-	-	-	-	-

表 7-13 Redis 5.0 单机/主备/Cluster 集群实例支持的命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo	Stream
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD	XACK
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	XADD
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	XCLAIM
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST	XDEL
-	SUBSCRIBE	WATCH	SELECT (Cluster 集群实例不支持)	SCRIPT KILL	GEORADIUS	XGROUP
-	UNSUBSCRIBE	-	-	SCRIPT LOAD	GEORADIUS BYMEMBER	XINFO
-	-	-	-	-	-	XLEN
-	-	-	-	-	-	XPENDING
-	-	-	-	-	-	XRANGE
-	-	-	-	-	-	XREAD
-	-	-	-	-	-	XREADGROUP
-	-	-	-	-	-	XREVRANGE
-	-	-	-	-	-	XTRIM

表 7-14 Redis 5.0 proxy 集群支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB



Generic (Key)	String	Hash	List	Set	Sorted Set	Server
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	ROLE
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	MEMORY
RENAME	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	COMMAND
RENAME NX	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	COMMAND COUNT
RESTORE	INCR	HSET	LREM	SPOP	ZREVRANGE	COMMAND GETKEYS
SORT	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND INFO
TTL	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	CONFIG GET
TYPE	MGET	HSCAN	RPOP	SUNION	ZSCORE	CONFIG RESETSTAT
SCAN	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	CONFIG REWRITE
OBJECT	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG SET
PEXPIRE	PSETEX	HKEYS	RPUSHX	-	ZSCAN	-
PEXPIREAT	SET	-	LPUSH	-	ZRANGEBYLEX	-
EXPIREAT	SETBIT	-	-	-	ZLEXCOUNT	-
KEYS	SETEX	-	-	-	ZREMRANGEBYSCORE	-

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
UNLINK	SETNX	-	-	-	ZREM	-
TOUCH	SETRANGE	-	-	-	ZREMRANGEBYLEX	-
RANDOMKEY	STRLEN	-	-	-	ZPOPMAX	-
-	BITFIELD	-	-	-	ZPOPMIN	-
-	GETBIT	-	-	-	BZPOPMAX	-
-	-	-	-	-	BZPOPMIN	-
-	-	-	-	-	ZREVRANGEBYLEX	-

表 7-15 Redis 5.0 proxy 集群支持的命令清单 2

Hyper Logging	Pub/Sub	Transactions	Connection	Scripting	Geo	Cluster
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD	CLUSTER INFO
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	CLUSTER NODES
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	CLUSTER SLOTS
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST	CLUSTER ADDSLOTS
-	SUBSCRIBE	WATCH	SELECT	SCRIPT KILL	GEORADIUS	ASKING
-	UNSUBSCRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADIUSBYMEMBER	READONLY
-	-	-	CLIENT LIST	SCRIPT DEBUG YES SYNC NO	GEOSEARCH	READWRITE

Hyper Logging	Pub/Sub	Transactions	Connection	Scripting	Geo	Cluster
-	-	-	CLIENT GETNAME	-	GEOSEARCH STORE	-
-	-	-	CLIENT SETNAME	-	-	-

 说明

上表中的Cluster类命令，仅2022/9/1日及之后创建的proxy集群实例支持。

表 7-16 Redis 5.0 读写分离支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLPUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	MONITOR
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	SLOWLOG
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	ROLE
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	SWAPDB
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	MEMORY

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	COMMAND COUNT
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	COMMAND GETKEYS
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	COMMAND INFO
SCAN	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG GET
OBJECT	PSETEX	-	RPUSHX	-	ZSCAN	CONFIG RESETSTAT
PEXPIRE	SET	-	LPUSH	-	ZRANGEBYLEX	CONFIG REWRITE
PEXPIREAT	SETBIT	-	-	-	ZLEXCOUNT	CONFIG SET
EXPIREAT	SETEX	-	-	-	ZREVRANGEBYSCORE	-
KEYS	SETNX	-	-	-	ZREM	-
UNLINK	SETRANGE	-	-	-	ZREMRANGEBYLEX	-
TOUCH	STRLEN	-	-	-	BZPOPMAX	-
-	BITFIELD	-	-	-	BZPOPMIN	-
-	GETBIT	-	-	-	ZPOPMAX	-
-	-	-	-	-	ZPOPMIN	-
-	-	-	-	-	ZREMRANGEBYLEX	-

表 7-17 Redis 5.0 读写分离支持的命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connecti on	Scripting	Geo	Stream
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD	XACK
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	XADD
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	XCLAIM
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST	XDEL
-	SUBSCRIBE	WATCH	SELECT	SCRIPT KILL	GEORADIUS	XGROUP
-	UNSUBSCRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADIUSBYMEMBER	XINFO
-	-	-	CLIENT LIST	SCRIPT DEBUG YES SYNC NO	GEOSEARCH	XLEN
-	-	-	CLIENT GETNAME	-	GEOSEARCHSTORE	XPENDING
-	-	-	CLIENT SETNAME	-	-	XRANGE
-	-	-	-	-	-	XREAD
-	-	-	-	-	-	XREADGROUP
-	-	-	-	-	-	XREVRANGE
-	-	-	-	-	-	XTRIM

## Redis 5.0 禁用的命令

以下列出了Redis 5.0实例禁用的命令。

表 7-18 Redis 5.0 单机和主备禁用命令

Generic (Key)	Server
MIGRATE	SLAVEOF
-	SHUTDOWN
-	LASTSAVE
-	DEBUG相关类
-	SAVE
-	BGSAVE
-	BGREWRITEAOF
-	SYNC
-	PSYNC

表 7-19 Redis 5.0 Proxy 集群实例禁用命令

Generic (Key)	Server
MIGRATE	BGREWRITEAOF
MOVE	BGSAVE
WAIT	CLIENT相关命令
-	DEBUG OBJECT
-	DEBUG SEGFAULT
-	LASTSAVE
-	PSYNC
-	SAVE
-	SHUTDOWN
-	SLAVEOF
-	LATENCY相关命令
-	MODULE相关命令
-	LOLWUT
-	SWAPDB
-	REPLICAOF
-	SYNC

表 7-20 Redis 5.0 Cluster 集群禁用命令

Generic (Key)	Server	Cluster
MIGRATE	SLAVEOF	CLUSTER MEET
-	SHUTDOWN	CLUSTER FLUSHSLOTS
-	LASTSAVE	CLUSTER ADDSLOTS
-	DEBUG相关类	CLUSTER DELSLOTS
-	SAVE	CLUSTER SETSLOT
-	BGSAVE	CLUSTER BUMPEPOCH
-	BGREWRITEAOF	CLUSTER SAVECONFIG
-	SYNC	CLUSTER FORGET
-	PSYNC	CLUSTER REPLICATE
-	-	CLUSTER COUNT-FAILURE-REPORTS
-	-	CLUSTER FAILOVER
-	-	CLUSTER SET-CONFIG-EPOCH
-	-	CLUSTER RESET

表 7-21 Redis 5.0 读写分离禁用命令

Generic	Server
MIGRATE	BGREWRITEAOF
WAIT	BGSAVE
-	DEBUG OBJECT
-	DEBUG SEGFAULT
-	LASTSAVE
-	LOLWUT
-	MODULE LIST/LOAD/UNLOAD
-	PSYNC
-	REPLICAOF
-	SAVE
-	SHUTDOWN [NOSAVE SAVE]
-	SLAVEOF

Generic	Server
-	SWAPDB
-	SYNC

## 支持重命名的命令

表 7-22 支持重命名的命令

支持重命名的命令	command、keys、flushdb、flushall、hgetall、scan、hscan、sscan、和zscan Proxy集群实例还支持dbsize和dbstats命令重命名，其他命令暂时不支持。
重命名的方式	请参考 <a href="#">命令重命名</a> 。

## 7.3 Redis 6.0 支持及禁用的命令

华为云DCS Redis 6.0完全兼容开源Redis 6。

本章节主要介绍DCS Redis 6.0命令的兼容性，包括支持命令列表，禁用命令列表。

命令的具体详细语法，请前往[Redis官方网站](#)查看。

DCS Redis缓存实例支持Redis的绝大部分命令，任何兼容Redis协议的客户端都可以访问DCS。

- 因安全原因，部分Redis命令在分布式缓存服务中被禁用，具体请见[Redis 6.0禁用的命令](#)。
- DCS集群实例支持多个key，但不支持跨slot访问的Redis命令列表，如[实例受限使用命令](#)所示。
- 部分Redis命令使用时有限制，例如KEYS、FLUSHDB、FLUSHALL等，具体请见[部分命令使用限制](#)。
- 部分高危命令支持命令重命名，具体请见[支持重命名的命令](#)。

### Redis 6.0 基础版支持的命令

- [表7-23](#)和[表7-24](#)列举了Redis 6.0单机/主备/Cluster集群实例支持的命令。
- [表7-25](#)和[表7-26](#)列举了Redis 6.0 Proxy集群支持的命令。
- [表7-27](#)和[表7-28](#)列举了Redis 6.0读写分离支持的命令。

各个命令的具体详细语法请前往[Redis官方网站](#)查看，例如您想了解SCAN命令的使用，可在[Redis官方网站](#)中搜索框中输入“SCAN”查询详细介绍。



表 7-23 Redis 6.0 单机/主备/Cluster 集群实例支持命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	CONFIGGET
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	MONITOR
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	SLOWLOG
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	ROLE
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	SWAPDB
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	MEMORY
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	CONFIG
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	ACL
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	COMMAND
SCAN	MSETNX	HLEN	RPOPLPUSH	SSCAN	ZINTERSTORE	-
OBJECT	PSETEX	-	RPUSH	SMISMEMBER	ZSCAN	-
PEXPIREAT	SET	-	RPUSHX	-	ZRANGEBYLEX	-
PEXPIRE	SETBIT	-	LPUSH	-	ZLEXCOUNT	-

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
KEYS	SETEX	-	BLMOVE	-	ZPOPMIN	-
COPY	SETNX	-	LMOVE	-	ZPOPMAX	-
-	SETRANGE	-	LPOS	-	ZREMRANGEBYSCORE	-
-	STRLEN	-	-	-	ZREM	-
-	BITFIELD	-	-	-	ZDIFF	-
-	BITFIELD_RO	-	-	-	ZDIFFSTORE	-
-	GETDEL	-	-	-	ZINTER	-
-	GETEX	-	-	-	ZMSCORE	-
-	-	-	-	-	ZRANDMEMBER	-
-	-	-	-	-	ZRANGESTORE	-
-	-	-	-	-	ZUNION	-

表 7-24 Redis 6.0 单机/主备/Cluster 集群实例支持命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo	Stream
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD	XACK
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	XADD
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	XCLAIM
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST	XDEL
-	SUBSCRIBE	WATCH	SELECT (Cluster 集群实例不支持)	SCRIPT KILL	GEORADIUS	XGROUP

HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo	Stream
-	UNSUBSCRIBE	-	CLIENT CACHING	SCRIPT LOAD	GEORADIUS BYMEMBER	XINFO
-	-	-	CLIENT GETREDIR	-	-	XLEN
-	-	-	CLIENT INFO	-	-	XPENDING
-	-	-	CLIENT TRACKING	-	-	XRANGE
-	-	-	CLIENT TRACKINGINFO	-	-	XREAD
-	-	-	CLIENT UNPAUSE	-	-	XREADGROUP
-	-	-	CLIENT KILL	-	-	XREVRANGE
-	-	-	CLIENT LIST	-	-	XTRIM
-	-	-	CLIENT GETNAME	-	-	XAUTOCLAIM
-	-	-	CLIENT SETNAME	-	-	XGROUP CREATECONSUMER
-	-	-	HELLO	-	-	-
-	-	-	RESET	-	-	-

表 7-25 Redis 6.0 Proxy 集群支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	ROLE
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	MEMORY
RENAME	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	COMMAND
RENAME NX	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	COMMAND COUNT
RESTORE	INCR	HSET	LREM	SPOP	ZREVRANGE	COMMAND GETKEYS
SORT	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND INFO
TTL	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	CONFIG GET
TYPE	MGET	HSCAN	RPOP	SUNION	ZSCORE	CONFIG RESETSTAT
SCAN	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	CONFIG REWRITE
OBJECT	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG SET

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
PEXPIRE	PSETEX	HKEYS	RPUSHX	SMISMEMBER	ZSCAN	-
PEXPIREAT	SET	HRANDFIELD	LPUSH	-	ZRANGEBYLEX	-
EXPIREAT	SETBIT	-	BLMOVE	-	ZLEXCOUNT	-
KEYS	SETEX	-	LMOVE	-	ZREMRANGEBYSCORE	-
UNLINK	SETNX	-	LPOS	-	ZREM	-
TOUCH	SETRANGE	-	-	-	ZREMRANGEBYLEX	-
RANDOMKEY	STRLEN	-	-	-	ZPOPMAX	-
COPY	BITFIELD	-	-	-	ZPOPMIN	-
-	GETBIT	-	-	-	BZPOPMAX	-
-	BITFIELD_RO	-	-	-	BZPOPMIN	-
-	GETDEL	-	-	-	ZREVRANGEBYLEX	-
-	GETEX	-	-	-	ZDIFF	-
-	-	-	-	-	ZDIFFSTORE	-
-	-	-	-	-	ZINTER	-
-	-	-	-	-	ZMSCORE	-
-	-	-	-	-	ZRANDMEMBER	-
-	-	-	-	-	ZRANGESTORE	-
-	-	-	-	-	ZUNION	-

表 7-26 Redis 6.0 Proxy 集群支持的命令清单 2

Hyper Logging	Pub/Sub	Transactions	Connection	Scripting	Geo	Cluster
PFADD	PUBLISH	DISCARD	AUTH	EVAL	GEOADD	CLUSTER INFO
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	CLUSTER NODES
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	CLUSTER SLOTS
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST	CLUSTER ADDSLOTS
-	SUBSCRIBE	WATCH	SELECT	SCRIPT KILL	GEORADIUS	ASKING
-	UNSUBSCRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADIUSBYMEMBER	READONLY
-	-	-	CLIENT LIST	SCRIPT DEBUG YES  SYNC  NO	GEOSEARCH	READWRITE
-	-	-	CLIENT GETNAME	-	GEOSEARCH STORE	-
-	-	-	CLIENT SETNAME	-	-	-
-	-	-	HELLO	-	-	-

表 7-27 Redis 6.0 读写分离支持的命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL (不支持 FLUSHALL SYNC)
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLPUSH	SDIFF	ZCOUNT	DBSIZE

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	MONITOR
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	SLOWLOG
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	ROLE
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	SWAPDB
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	MEMORY
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	COMMAND
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	COMMAND COUNT
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	COMMAND GETKEYS
TYPE	MSET	HSTRLEN	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	COMMAND INFO
SCAN	MSETNX	HLEN	RPUSH	SSCAN	ZINTERSTORE	CONFIG GET
OBJECT	PSETEX	HRANDFIELD	RPUSHX	SMISMEMBER	ZSCAN	CONFIG RESETSTAT
PEXPIRE	SET	-	LPUSH	-	ZRANGEBYLEX	CONFIG REWRITE
PEXPIREAT	SETBIT	-	BLMOVE	-	ZLEXCOUNT	CONFIG SET
EXPIREAT	SETEX	-	LMOVE	-	ZREMRANGEBYSCORE	-

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
KEYS	SETNX	-	LPOS	-	ZREM	-
UNLINK	SETRANGE	-	-	-	ZREMRANGEBYLEX	-
TOUCH	STRLEN	-	-	-	BZPOPMAX	-
COPY	BITFIELD	-	-	-	BZPOPMIN	-
-	GETBIT	-	-	-	ZPOPMAX	-
-	BITFIELD_RO	-	-	-	ZPOPMIN	-
-	GETDEL	-	-	-	ZREVRANGEBYLEX	-
-	GETEX	-	-	-	ZDIFF	-
-	-	-	-	-	ZDIFFSTORE	-
-	-	-	-	-	ZINTER	-
-	-	-	-	-	ZMSCORE	-
-	-	-	-	-	ZRANDMEMBER	-
-	-	-	-	-	ZRANGESTORE	-
-	-	-	-	-	ZUNION	-

表 7-28 Redis 6.0 读写分离支持的命令清单 2

HyperLoglog	Pub/Sub	Transactions	Connecti on	Scripting	Geo	Stream
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD	XAUTOCCLAIM
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH	XGROUPCREATECONSUMER



HyperLoglog	Pub/Sub	Transactions	Connection	Scripting	Geo	Stream
PFMERG E	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS	XACK
-	PUNSUB SCRIBE	UNWATC H	QUIT	SCRIPT FLUSH	GEODIST	XADD
-	SUBSCRI BE	WATCH	SELECT	SCRIPT KILL	GEORADI US	XCLAIM
-	UNSUBS CRIBE	-	CLIENT KILL	SCRIPT LOAD	GEORADI USBYME MBER	XDEL
-	-	-	CLIENT LIST	SCRIPT DEBUG YES  SYNC NO	GEOSEAR CH	XGROUP
-	-	-	CLIENT GETNAM E	-	GEOSEAR CHSTOR E	XINFO
-	-	-	CLIENT SETNAM E	-	-	XLEN
-	-	-	HELLO	-	-	XPENDIN G
-	-	-	-	-	-	XRANGE
-	-	-	-	-	-	XREAD
-	-	-	-	-	-	XREADG ROUP
-	-	-	-	-	-	XREVRA NGE
-	-	-	-	-	-	XTRIM

## Redis 6.0 企业版支持的命令

以下列出了Redis 6.0企业版实例支持的命令。

表 7-29 Redis 6.0 企业版实例支持命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
COPY	APPEND	HDEL	BLMOVE	SADD	BZPOPMAX	FLUSHALL
DEL	BITCOUNT	HEXISTS	LINDEX	SCARD	BZPOPMIN	FLUSHDB
DUMP	BITOP	HGET	LINSERT	SDIFF	ZADD	DBSIZE
EXISTS	BITPOS	HGETALL	LLEN	SDIFFSTORE	ZCARD	TIME
EXPIRE	BITFIELD	HINCRBY	LPOP	SINTER	ZCOUNT	INFO
MOVE	DECR	HINCRBYFLOAT	LPUSHX	SINTERSTORE	ZDIFF	CLIENT KILL
PERSIST	DECRBY	HKEYS	LRANGE	SISMEMBER	ZDIFFSTORE	CLIENT LIST
PTTL	GET	HMGET	LREM	SMEMBERS	ZINCRBY	CLIENT GETNAME
RANDOMKEY	GETRANGE	HMSET	LSET	SMOVE	ZINTER	CLIENT SETNAME
RENAME	GETSET	HSET	LTRIM	SPOP	ZINTERSTORE	CONFIG GET
RENAME NX	GETDEL	HSETNX	RPOP	SRANDMEMBER	ZLEXCOUNT	MONITOR
SORT	GETEX	HVALS	LMOVE	SREM	ZMSCORE	SLOWLOG
TTL	INCR	HSCAN	RPOPLPUSH	SUNION	ZPOPMAX	ROLE
TYPE	INCRBY	HSTRLEN	RPUSH	SUNIONSTORE	ZPOPMIN	SWAPDB
SCAN	INCRBYFLOAT	HLEN	RPUSHX	SSCAN	ZRANDMEMBER	MEMORY
PEXPIREAT	MGET	HRANDFIELD	LPUSH	SMISMEMBER	ZRANGE	LASTSAVE
PEXPIRE	MSET	-	BLPOP	-	ZRANGEBYLEX	REPLCONF EX

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
OBJECT ENCODING	MSETNX	-	BRPOP	-	ZRANGEBYSCORE	LASTSAVE
OBJECT FREQ	PSETEX	-	BRPOPLPUSH	-	ZRANGESTORE	COMMAND
OBJECT IDLETIME	SET	-	LPOS	-	ZRANK	COMMAND COUNT
OBJECT REFCOUNT	SETBIT	-	-	-	ZREM	COMMAND GETKEYS
RESTORE	SETEX	-	-	-	ZREMRANGEBYLEX	COMMAND INFO
TOUCH	SETNX	-	-	-	ZREMRANGEBYRANK	CONFIG
UNLINK	SETRANGE	-	-	-	ZREMRANGEBYSCORE	-
EXPIREAT	STRLEN	-	-	-	ZREVRANGE	-
KEYS	SUBSTR	-	-	-	ZREVRANGEBYLEX	-
WAIT	-	-	-	-	ZREVRANGEBYSCORE	-
-	-	-	-	-	ZREVRANK	-
-	-	-	-	-	ZSCAN	-
-	-	-	-	-	ZSCORE	-
-	-	-	-	-	ZUNION	-
-	-	-	-	-	ZUNIONSTORE	-

表 7-30 Redis 6.0 企业版实例支持命令清单 2

HyperLoglog	Pub/Sub	Connection	Scripting	Geo	Stream	Bitmaps
PFADD	PSUBSCRIBE	AUTH	EVAL	GEOADD	XACK	BITCOUNT

HyperLoglog	Pub/Sub	Connection	Scripting	Geo	Stream	Bitmaps
PFCOUNT	PUBLISH	CLIENT CACHING	EVALSHA	GEODIST	XADD	BITFIELD
PFMERGE	PUBSUB	CLIENT GETNAME	SCRIPT DEBUG	GEOHASH	XAUTOCLAIM	BITFIELD_RO
PFSELFTEST	PUNSUBSCRIBE	CLIENT GETREDIR	SCRIPT EXISTS	GEOPOS	XCLAIM	BITOP
-	SUBSCRIBE	CLIENT ID	SCRIPT FLUSH	GEORADIUS	XDEL	BITPOS
-	UNSUBSCRIBE	CLIENT INFO	SCRIPT KILL	GEORADIUSBYMEMBER	XGROUP	GETBIT
-	-	CLIENT KILL	SCRIPT LOAD	GEORADIUSBYMEMBER_RO	XINFO	SETBIT
-	-	CLIENT LIST	-	GEORADIUS_RO	XLEN	-
-	-	CLIENT PAUSE	-	GEOSEARCH	XPENDING	-
-	-	CLIENT REPLY	-	GEOSEARCHSTORE	XRANGE	-
-	-	CLIENT SETNAME	-	-	XREAD	-
-	-	CLIENT TRACKING	-	-	XREADGROUP	-
-	-	CLIENT TRACKINGINFO	-	-	XREVRANGE	-
-	-	CLIENT UNBLOCK	-	-	XSETID	-
-	-	CLIENT UNPAUSE	-	-	XTRIM	-
-	-	ECHO	-	-	-	-

HyperLoglog	Pub/Sub	Connection	Scripting	Geo	Stream	Bitmaps
-	-	HELLO	-	-	-	-
-	-	PING	-	-	-	-
-	-	QUIT	-	-	-	-
-	-	RESET	-	-	-	-
-	-	SELECT	-	-	-	-

## Redis 6.0 禁用的命令

表 7-31 Redis 6.0 单机/主备/Cluster 集群实例的禁用命令

Generic (Key)	Server	Cluster
MIGRATE	SLAVEOF	CLUSTER MEET
-	SHUTDOWN	CLUSTER FLUSHSLOTS
-	LASTSAVE	CLUSTER ADDSLOTS
-	DEBUG相关类	CLUSTER DELSLOTS
-	SAVE	CLUSTER SETSLOT
-	BGSAVE	CLUSTER BUMPEPOCH
-	BGREWRITEAOF	CLUSTER SAVECONFIG
-	SYNC	CLUSTER FORGET
-	PSYNC	CLUSTER REPLICATE
-	-	CLUSTER COUNT-FAILURE-REPORTS
-	-	CLUSTER FAILOVER
-	-	CLUSTER SET-CONFIG-EPOCH
-	-	CLUSTER RESET

表 7-32 Redis 6.0 Proxy 集群实例禁用命令

Generic (Key)	Server	Connection
MIGRATE	BGREWRITEAOF	CLIENT CACHING
MOVE	BGSAVE	CLIENT GETREDIR

Generic (Key)	Server	Connection
WAIT	CLIENT相关命令	CLIENT INFO
-	DEBUG OBJECT	CLIENT TRACKING
-	DEBUG SEGFAULT	CLIENT TRACKINGINFO
-	LASTSAVE	CLIENT UNPAUSE
-	PSYNC	RESET
-	SAVE	-
-	SHUTDOWN	-
-	SLAVEOF	-
-	LATENCY相关命令	-
-	MODULE相关命令	-
-	LOLWUT	-
-	SWAPDB	-
-	REPLICAOF	-
-	SYNC	-
-	ACL	-
-	FAILOVER	-

表 7-33 Redis 6.0 读写分离禁用命令

Generic	Server	Connection
MIGRATE	BGREWRITEAOF	CLIENT CACHING
WAIT	BGSAVE	CLIENT GETREDIR
-	DEBUG OBJECT	CLIENT INFO
-	DEBUG SEGFAULT	CLIENT TRACKING
-	LASTSAVE	CLIENT TRACKINGINFO
-	LOLWUT	CLIENT UNPAUSE
-	MODULE LIST/LOAD/ UNLOAD	RESET
-	PSYNC	-
-	REPLICAOF	-
-	SAVE	-

Generic	Server	Connection
-	SHUTDOWN [NOSAVE SAVE]	-
-	SLAVEOF	-
-	SWAPDB	-
-	SYNC	-
-	ACL	-
-	FAILOVER	-

表 7-34 Redis 6.0 企业版实例的禁用命令

Generic (Key)	Server	HyperLoglog
MIGRATE	SLAVEOF	PFDEBUG
-	SHUTDOWN	-
-	SAVE	-
-	BGSAVE	-
-	BGREWRITEAOF	-
-	SYNC	-
-	PSYNC	-
-	REPLICAOF	-

## 支持重命名的命令

表 7-35 支持重命名的命令

支持重命名的命令	command、keys、flushdb、flushall、hgetall、scan、hscan、sscan、和zscan Proxy集群实例还支持dbsize和dbstats命令重命名，其他命令暂时不支持。
重命名的方式	请参考 <a href="#">命令重命名</a> 。

## 7.4 Web CLI 中支持和禁用命令

本章节主要介绍DCS管理控制台Web CLI工具的命令兼容性，列举支持和禁用的命令列表。当前仅Redis 4.0、Redis 5.0和Redis 6.0版本支持Web CLI功能。

 说明

- 当前在Web CLI下所有命令参数暂不支持中文且key和value不支持空格。
- 当value值为空时，执行get命令返回nil。

## Web CLI 支持的命令

以下列出了通过Web CLI连接Redis实例时支持的命令。各个命令的具体详细语法请前往[Redis官方网站](#)查看，例如您想了解SCAN命令的使用，可在[Redis官方网站](#)中搜索框中输入“SCAN”查询详细介绍。

表 7-36 Web CLI 支持命令清单 1

Generic (Key)	String	List	Set	Sorted Set	Server
DEL	APPEND	R PUSH	SADD	ZADD	FLUSHALL
OBJECT	BITCOUNT	R PUSHX	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	LLEN	SINTERSTORE	ZRANGEBYSCORE	CLIENT KILL
PTTL	GET	LPOP	SISMEMBER	ZRANK	CLIENT LIST
RANDOM KEY	GETRANGE	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	CLIENT GETNAME
RENAME	GETSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	CLIENT SETNAME
RENAMENX	INCR	LREM	SPOP	ZREVRANGE	CONFIG GET
SCAN	INCRBY	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	SLOWLOG
SORT	INCRBYFLOAT	LTRIM	SREM	ZREVRANK	ROLE
TTL	MGET	RPOP	SUNION	ZSCORE	SWAPDB
TYPE	MSET	RPOPLPUSH	SUNIONSTORE	ZUNIONSTORE	MEMORY
-	MSETNX	RPOPLPUSH	SSCAN	ZINTERSTORE	-



Generic (Key)	String	List	Set	Sorted Set	Server
-	PSETEX	-	-	ZSCAN	-
-	SET	-	-	ZRANGEBYLEX	-
-	SETBIT	-	-	ZLEXCOUNT	-
-	SETEX	-	-	-	-
-	SETNX	-	-	-	-
-	SETRANGE	-	-	-	-
-	STRLEN	-	-	-	-
-	BITFIELD	-	-	-	-

表 7-37 Web CLI 支持命令清单 2

Hash	HyperLoglog	Connection	Scripting	Geo	Pub/Sub
HDEL	PFADD	AUTH	EVAL	GEOADD	UNSUBSCRIBE
HEXISTS	PFCOUNT	ECHO	EVALSHA	GEOHASH	PUBLISH
HGET	PFMERGE	PING	SCRIPT EXISTS	GEOPOS	PUBSUB
HGETALL	-	QUIT	SCRIPT FLUSH	GEODIST	PUNSUBSCRIBE
HINCRBY	-	-	SCRIPT KILL	GEORADIUS	-
HINCRBYFLOAT	-	-	SCRIPT LOAD	GEORADIUS BYMEMBER	-
HKEYS	-	-	-	-	-
HMGET	-	-	-	-	-
HMSET	-	-	-	-	-
HSET	-	-	-	-	-
HSETNX	-	-	-	-	-
HVALS	-	-	-	-	-
HSCAN	-	-	-	-	-

Hash	HyperLoglog	Connection	Scripting	Geo	Pub/Sub
HSTRLEN	-	-	-	-	-

## Web CLI 禁用的命令

以下列出了通过Web CLI连接Redis实例时禁用的命令。

表 7-38 通过 Web CLI 禁用的命令 1

Generic (Key)	Server	Transactions	Cluster
MIGRATE	SLAVEOF	UNWATCH	CLUSTER MEET
WAIT	SHUTDOWN	REPLICAOF	CLUSTER FLUSHSLOTS
DUMP	DEBUG相关类	DISCARD	CLUSTER ADDSLOTS
RESTORE	CONFIG SET	EXEC	CLUSTER DELSLOTS
-	CONFIG REWRITE	MULTI	CLUSTER SETSLOT
-	CONFIG RESETSTAT	WATCH	CLUSTER BUMPEPOCH
-	SAVE	-	CLUSTER SAVECONFIG
-	BGSAVE	-	CLUSTER FORGET
-	BGREWRITEAOF	-	CLUSTER REPLICATE
-	COMMAND	-	CLUSTER COUNT-FAILURE-REPORTS
-	KEYS	-	CLUSTER FAILOVER
-	MONITOR	-	CLUSTER SET-CONFIG-EPOCH
-	SYNC	-	CLUSTER RESET
-	PSYNC	-	-
-	ACL	-	-
-	MODULE	-	-

表 7-39 通过 Web CLI 禁用的命令 2

List	Connection	Sorted Set	Pub/Sub
BLPOP	SELECT	BZPOPMAX	PSUBSCRIBE

List	Connection	Sorted Set	Pub/Sub
BRPOP	-	BZPOPMIN	SUBSCRIBE
BLMOVE	-	BZMPOP	-
BRPOPLPUSH	-	-	-
BLMPOP	-	-	-

## 7.5 实例受限使用命令

Cluster集群实例支持多个Key，但不支持跨slot访问的Redis命令。受限使用的命令如表7-40。

Proxy集群实例支持多Key的命令中，部分命令不支持跨slot访问，请参考表7-42。受限使用的命令如表7-41。

读写分离实例受限使用的命令，如表7-43。

### 说明

当用户执行比较耗时的命令（如flushall）时，可能会导致缓存实例在命令执行期间对外不响应用户的其它命令，造成状态监控失效，此时Console上缓存实例的状态会变成异常，命令执行结束后，实例状态会恢复正常。

### Cluster 集群实例受限使用的 Redis 命令

表 7-40 Cluster 集群实例受限使用的 Redis 命令

命令类型	命令描述
<b>Set（集合）</b>	
SINTER	返回一个集合的全部成员，该集合是所有给定集合的交集
SINTERSTORE	类似SINTER，但结果保存到destination集合
SUNION	返回一个集合的全部成员，该集合是所有给定集合的并集
SUNIONSTORE	和SUNION类似，但它将结果保存到destination集合
SDIFF	返回一个集合的全部成员，该集合是所有给定集合之间的差集
SDIFFSTORE	和SDIFF类似，但它将结果保存到destination集合
SMOVE	将member元素从source集合移动到destination集合
<b>SortedSet（有序集合）</b>	
ZUNIONSTORE	计算给定的一个或多个有序集的并集
ZINTERSTORE	计算给定的一个或多个有序集的交集

命令类型	命令描述
<b>HyperLogLog</b>	
PFCOUNT	返回储存在给定键（或多个键）的HyperLogLog的近似基数
PFMERGE	将多个HyperLogLog合并（merge）为一个HyperLogLog
<b>Keys（键）</b>	
RENAME	将key改名
RENAMENX	将key改名，新key必须是之前不存在的
BITOP	对一个或多个保存二进制位的字符串key进行位元操作，并将结果保存到destkey上
RPOPLPUSH	返回并移除存储在source的列表的最后一个元素（列表尾部元素），并把该元素放入存储在destination的列表的第一个元素位置（列表头部）
<b>String（字符串）</b>	
MSETNX	同时设置一个或多个key-value对

## Proxy 集群实例受限使用的 Redis 命令

表 7-41 Proxy 集群实例受限使用的 Redis 命令

命令类型	命令	受限使用条件
Set（集合）	SMOVE	proxy集群要求源key和目标key在同一个slot。
Sorted Set（有序集合）	BZPOPMAX	Proxy集群实例要求传入的key都在同一个slot中。 Redis 4.0 Proxy集群实例不支持。
	BZPOPMIN	
GEO（地理位置）	GEORADIUS	<ul style="list-style-type: none"> <li>Proxy集群实例要求传入的key都在同一个slot中。</li> <li>Proxy集群的多DB模式下暂不支持带STORE参数。</li> </ul>
	GEORADIUSBYMEMBER	
	GEOSEARCHSTORE	
Connection（连接）	CLIENT KILL	<ul style="list-style-type: none"> <li>仅支持两种形式： <ul style="list-style-type: none"> <li>CLIENT KILL ip:port</li> <li>CLIENT KILL ADDR ip:port</li> </ul> </li> <li>id字段为随机值，不满足idc1&lt;idc2 → Tc1&lt;Tc2</li> </ul>

命令类型	命令	受限使用条件
	CLIENT LIST	<ul style="list-style-type: none"> <li>仅支持两种形式：                             <ul style="list-style-type: none"> <li>CLIENT LIST</li> <li>CLIENT LIST [TYPE normal master replica pubsub]</li> </ul> </li> <li>id字段为随机值，不满足<math>idc1 &lt; idc2 \rightarrow Tc1 &lt; Tc2</math></li> </ul>
	SELECT index	Proxy集群的多DB支持当前通过改key实现，不推荐使用该方案。 Proxy集群支持多DB限制请参考 <a href="#">Proxy集群使用多DB限制</a> 。
HyperLogLog	PFCOUNT	Proxy集群实例要求传入的key都在同一个slot中。
	PFMERGE	
Keys (键)	RENAME	Proxy集群实例要求传入的key都在同一个slot中。
	RENAMENX	
	SCAN	<ul style="list-style-type: none"> <li>Proxy集群实例不支持在pipeline中使用SCAN命令。</li> <li>Proxy集群SCAN命令支持指定分片：在命令后面添加 ip:port。(执行iclusternodes命令可以查询不同分片的IP地址及端口。)如果执行SCAN命令指定分片未能返回需要信息，可能是因为您的Proxy集群的Proxy节点为老版本，请联系技术支持升级Proxy版本。</li> </ul> <pre>iclusternodes xxx 192.168.00.00:1111@xxx xxx connected 10923-16383 xxx 192.168.00.01:2222@xxx xxx connected 0-5460 xxx 192.168.00.02:3333@xxx xxx connected 5461-10922 SCAN 0 match * COUNT 5 192.168.00.02:3333 1) "0" 2) 1) "key1"    2) "key2"    3) "key3"    4) "key4"    5) "key5"</pre>
Lists (列表)	BLPOP	Proxy集群实例要求传入的key都在同一个slot中。
	BRPOP	
	BRPOPLPUSH	
Pub/Sub (发布/订阅)	PSUBSCRIBE	Proxy集群事件订阅，不支持键空间事件订阅，键空间事件订阅虽不会失败，但功能本身不支持。

命令类型	命令	受限使用条件
Scripting (脚本)	EVAL	<ul style="list-style-type: none"> <li>Proxy集群实例要求传入的key都在同一个slot中。</li> <li>Proxy集群开启多DB时，KEYS参数会被修改，Lua脚本中使用到KEYS的地方需要注意。</li> </ul>
	EVALSHA	
Server (服务器)	MEMORY DOCTOR	<p>Proxy集群要在命令后面加上节点的ip:port。</p> <p>获取节点IP和端口的方式请参考（以MEMORY USAGE为例）：</p> <ol style="list-style-type: none"> <li>执行命令<b>cluster keyslot key</b>查询key所在的slot号。</li> <li>执行<b>icluster nodes</b>查询不同slot区间对应的IP地址及端口，获取key值对应slot号区间对应的IP和端口。 如果执行<b>icluster nodes</b>未能返回需要信息，可能是因为您的Proxy集群实例为老版本，请执行<b>cluster nodes</b>重新查询。</li> <li>执行<b>MEMORY USAGE key ip:port</b>。 如果Proxy集群开启了多DB，则执行<b>MEMORY USAGE xxx:As{key} ip:port</b>，其中xxx为key值所在的DB，例如DB0，DB1，DB255分别对应000，001，255。 单DB Proxy集群实例示例如下：</li> </ol> <pre> set key1 value1 OK get key1 value1 cluster keyslot key1 9189 icluster nodes xxx 192.168.00.00:1111@xxx xxx connected 10923-16383 xxx 192.168.00.01:2222@xxx xxx connected 0-5460 xxx 192.168.00.02:3333@xxx xxx connected 5461-10922 MEMORY USAGE key1 192.168.00.02:3333 54                     </pre>
	MEMORY HELP	
	MEMORY MALLOC-STATS	
	MEMORY PURGE	
	MEMORY STATS	
	MEMORY USAGE	
	MONITOR	
Strings (字符串)	BITOP	Proxy集群实例要求传入的key都在同一个slot中。
	MSETNX	
Transactions (事务)	WATCH	Proxy集群实例要求传入的key都在同一个slot中。

命令类型	命令	受限使用条件
	MULTI	不保证事务中跨slot命令的有序性。 事务中禁用的命令：WATCH, MONITOR, RANDOMKEY, KEYS, SCAN, SUBSCRIBE, UNSUBSCRIBE, PSUBSCRIBE, PUNSUBSCRIBE, SCRIPT, EVAL, EVALSHA, DBSIZE, AUTH, FLUSHDB, FLUSHALL, CLIENT, MEMORY
	EXEC	
Streams (流)	XACK	proxy集群目前不支持streams的使用。
	XADD	
	XCLAIM	
	XDEL	
	XGROUP	
	XINFO	
	XLEN	
	XPENDING	
	XRANGE	
	XTRIM	
	XREVRANGE	
	XREAD	
	XREADGROUP GROUP	
XAUTOCLAIM		

## Proxy 集群多 Key 命令说明

表 7-42 Proxy 集群多 Key 命令说明

类型	命令
支持跨slot的多Key命令	DEL、MGET、MSET、EXISTS、SUNION、SINTER、SDIFF、SUNIONSTORE、SINTERSTORE、SDIFFSTORE、ZUNIONSTORE、ZINTERSTORE
不支持跨slot的多Key命令	SMOVE、SORT、BITOP、MSETNX、RENAME、RENAMENX、BLPOP、BRPOP、RPOPLPUSH、BRPOPLPUSH、PFMERGE、PFCOUNT、BLMOVE、COPY、GEOSEARCHSTORE、LMOVE、ZRANGESTORE

## 读写分离实例受限使用的 Redis 命令

表 7-43 读写分离实例受限使用的 Redis 命令

命令类型	命令	受限使用条件
Connection (连接)	CLIENT KILL	<ul style="list-style-type: none"> <li>仅支持两种形式：                             <ul style="list-style-type: none"> <li>CLIENT KILL ip:port</li> <li>CLIENT KILL ADDR ip:port</li> </ul> </li> <li>id字段为随机值，不满足<math>idc1 &lt; idc2 \rightarrow Tc1 &lt; Tc2</math></li> </ul>
	CLIENT LIST	<ul style="list-style-type: none"> <li>仅支持两种形式：                             <ul style="list-style-type: none"> <li>CLIENT LIST</li> <li>CLIENT LIST [TYPE normal master replica pubsub]</li> </ul> </li> <li>id字段为随机值，不满足<math>idc1 &lt; idc2 \rightarrow Tc1 &lt; Tc2</math></li> </ul>

## 7.6 部分命令使用限制

本章节主要介绍部分Redis命令使用时的限制。

### Key 相关命令使用限制

使用KEYS命令时，若缓存数据量较大，可能会较长时间阻塞其它业务命令操作，甚至可能过高地占用额外内存。因此使用KEYS命令时请尽量描述精确的pattern、不要使用“keys \*”进行全通配。建议尽量避免在生产环境使用，否则会影响服务的健康运行。

### Server 相关命令使用限制

- 当用户执行比较耗时的命令（如flushall）时，可能会导致缓存实例在命令执行期间对外不响应用户的其它命令，造成状态监控失效，此时Console上缓存实例的状态会变成异常，命令执行结束后，实例状态会恢复正常。
- 使用FLUSHDB、FLUSHALL命令时，若缓存数据量较大，可能会较长时间阻塞其它业务命令操作。

### EVAL 和 EVALSHA 相关命令使用限制

- 使用EVAL和EVALSHA命令时，命令参数中必须带有至少1个key。否则客户端会提示“ERR eval/evalsha numkeys must be bigger than zero in redis cluster mode”的错误。
- 使用EVAL和EVALSHA命令时，DCS Redis集群实例使用第一个key来计算slot，用户代码需要保证操作的key是在同一个slot，具体请参考[Redis命令](#)。
- 使用EVAL命令时：
  - 建议使用前先了解Redis的lua脚本特性，具体可参考[Lua脚本](#)。
  - lua脚本的执行超时时间为5秒钟，建议不要在lua脚本中使用比较耗时的代码，比如长时间的sleep、大的循环等语句。



- 调用lua脚本时，建议不要使用随机函数去指定key，否则在主备节点上执行结果不一致，从而导致主备节点数据不一致。

## Lua 脚本调试命令

Proxy集群和读写分离实例在执行Lua脚本调试命令时，仅支持异步非阻塞--ldb模式，不支持同步阻塞--ldb-sync-mode。且每个Proxy节点默认限制并发2个。其他Redis实例类型无此限制。

## 其他限制

- 单个Redis命令处理时长限制为15秒左右，超过15秒未处理完，会导致客户的其它业务失败，因此内部会触发主从倒换。
- 2018年7月10日前创建的Redis集群实例需要升级才支持以下命令：  
SINTER、SDIFF、SUNION、PFCOUNT、PFMERGE、SINTERSTORE、SUNIONSTORE、SDIFFSTORE、SMOVE、ZUNIONSTORE、ZINTERSTORE、EVAL、EVALSHA、BITOP、RENAME、RENAMENX、RPOPLPUSH、MSETNX、SCRIPT LOAD、SCRIPT KILL、SCRIPT EXISTS、SCRIPT FLUSH。

## 7.7 Redis 3.0 支持及禁用的命令（已停售）

华为云DCS Redis 3.0基于开源3.0.7版本进行开发，兼容开源的协议和命令。本章节主要介绍DCS Redis 3.0命令的兼容性，包括支持命令列表，禁用命令列表，以及不支持的高版本Redis脚本和命令列表，以及命令使用限制说明。

### 📖 说明

DCS Redis 3.0已下线，暂停售卖，建议使用Redis 4.0/5.0/6.0。

DCS Redis缓存实例支持Redis的绝大部分命令，任何兼容Redis协议的客户端都可以访问DCS。

- 因安全原因，部分Redis命令在分布式缓存服务中被禁用，具体请见[Redis 3.0禁用的命令](#)。
- DCS集群实例支持多个key，但不支持跨slot访问的Redis命令列表，如[实例受限使用命令](#)所示。
- 部分Redis命令使用时有限制，例如KEYS、FLUSHDB、FLUSHALL等，具体请见[部分命令使用限制](#)。

## Redis 3.0 支持的命令

以下列出了Redis 3.0实例支持的命令。各个命令的具体详细语法请前往[Redis官方网站](#)查看，例如您想了解SCAN命令的使用，可在[Redis官方网站](#)中搜索框中输入“SCAN”查询详细介绍。

 说明

- Redis高版本的命令，在低版本中不被兼容。判断DCS Redis是否支持某个命令，可通过在Redis-cli执行该命令，如果得到 ( error ) ERR unknown command ‘xxx’ 的提示，则说明不支持该命令。
- 如果是Proxy集群实例，不支持表格中以下命令：
  - “List” 类型中的BLPOP、BRPOP、BRPOPLRUSH命令。
  - “Server” 类型的CLIENT相关命令，包括CLIENT KILL、CLIENT GETNAME、CLIENT LIST、CLIENT SETNAME、CLIENT PAUSE、CLIENT REPLY。
  - “Server” 类型的MONITOR命令。
  - “Transaction” 类型的UNWATCH、WATCH命令。
  - 如果是比较旧的集群实例，不支持“Key” 类型中的RANDOMKEY命令。

表 7-44 Redis 3.0 支持命令清单 1

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
DEL	APPEND	HDEL	BLPOP	SADD	ZADD	FLUSHALL
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD	FLUSHDB
EXISTS	BITOP	HGET	BRPOPLRUSH	SDIFF	ZCOUNT	DBSIZE
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY	TIME
MOVE	DECR	HINCRBY	LINSERT	SINTER	ZRANGE	INFO
PERSIST	DECRBY	HINCRBYFLOAT	LLEN	SINTERSTORE	ZRANGEBYSCORE	CLIENT KILL
PTTL	GET	HKEYS	LPOP	SISMEMBER	ZRANK	CLIENT LIST
RANDOMKEY	GETRANGE	HMGET	LPUSHX	SMEMBERS	ZREMRANGEBYRANK	CLIENT GETNAME
RENAME	GETSET	HMSET	LRANGE	SMOVE	ZREMRANGEBYSCORE	CLIENT SETNAME
RENAME NX	INCR	HSET	LREM	SPOP	ZREVRANGE	CONFIG GET
RESTORE	INCRBY	HSETNX	LSET	SRANDMEMBER	ZREVRANGEBYSCORE	MONITOR
SORT	INCRBYFLOAT	HVALS	LTRIM	SREM	ZREVRANK	SLOWLOG

Generic (Key)	String	Hash	List	Set	Sorted Set	Server
TTL	MGET	HSCAN	RPOP	SUNION	ZSCORE	ROLE
TYPE	MSET	-	RPOPLPU	SUNIONSTORE	ZUNIONSTORE	-
SCAN	MSETNX	-	RPOPLPUSH	SSCAN	ZINTERSTORE	-
OBJECT	PSETEX	-	RPUSH	-	ZSCAN	-
KEYS	SET	-	RPUSHX	-	ZRANGEBYLEX	-
-	SETBIT	-	-	-	-	-
-	SETEX	-	-	-	-	-
-	SETNX	-	-	-	-	-
-	SETRANGE	-	-	-	-	-
-	STRLEN	-	-	-	-	-

表 7-45 Redis 3.0 支持命令清单 2

HyperLogLog	Pub/Sub	Transactions	Connection	Scripting	Geo
PFADD	PSUBSCRIBE	DISCARD	AUTH	EVAL	GEOADD
PFCOUNT	PUBLISH	EXEC	ECHO	EVALSHA	GEOHASH
PFMERGE	PUBSUB	MULTI	PING	SCRIPT EXISTS	GEOPOS
-	PUNSUBSCRIBE	UNWATCH	QUIT	SCRIPT FLUSH	GEODIST
-	SUBSCRIBE	WATCH	SELECT	SCRIPT KILL	GEORADIUS
-	UNSUBSCRIBE	-	-	SCRIPT LOAD	GEORADIUSBYMEMBER

## Redis 3.0 禁用的命令

以下列出了Redis 3.0实例禁用的命令。

表 7-46 Redis 3.0 单机和主备实例禁用命令

Generic (Key)	Server
MIGRATE	SLAVEOF
-	SHUTDOWN
-	LASTSAVE
-	DEBUG相关类
-	COMMAND
-	SAVE
-	BGSAVE
-	BGREWRITEAOF

表 7-47 Redis 3.0 Proxy 集群实例禁用命令

Generic (Key)	Server	List	Transactions	Connection	Cluster	codis相关
MIGRATE	SLAVEOF	BLPOP	DISCARD	SELECT	CLUSTER	TIME
MOVE	SHUTDOWN	BRPOP	EXEC	-	-	SLOTSINFO
-	LASTSAVE	BRPOPLPUSH	MULTI	-	-	SLOTSDEL
-	DEBUG相关类	-	UNWATCH	-	-	SLOTSMGRTSLOT
-	COMMAND	-	WATCH	-	-	SLOTSMGRTONE
-	SAVE	-	-	-	-	SLOTSCHECK
-	BGSAVE	-	-	-	-	SLOTSMGRTTAGSLOT
-	BGREWRITEAOF	-	-	-	-	SLOTSMGRTTAGONE
-	SYNC	-	-	-	-	-
-	PSYNC	-	-	-	-	-
-	MONITOR	-	-	-	-	-
-	CLIENT相关类	-	-	-	-	-

Generic (Key)	Server	List	Transactions	Connection	Cluster	codis相关
-	OBJECT	-	-	-	-	-
-	ROLE	-	-	-	-	-

## 7.8 Memcached 支持和禁用的命令（已停售）

### 📖 说明

DCS Memcached已停售，建议使用Redis实例。

Memcached引擎支持基于TCP（Memcached Text Protocol）的文本协议和二进制（Memcached Binary Protocol）协议，任何兼容Memcached协议的客户端都可以访问DCS。

### 文本协议

Memcached文本协议通过ASCII文本传递命令，便于用户编写客户端和调测问题，甚至可以直接使用Telnet连接Memcached实例。

Memcached文本协议与二进制协议相比，兼容更多的开源类型客户端，但文本协议不支持认证操作。

### 📖 说明

Memcached实例需要开启免密访问模式后，才能通过文本协议连接。此时Memcached实例的访问将不再受用户名、密码的认证保护。同一VPC内符合安全组规则的任何Memcached客户端均可连接访问Memcached实例，存在安全风险，请谨慎使用。

DCS Memcached实例对文本协议命令的支持情况如表7-48所示。

表 7-48 Memcached 实例文本协议支持的命令说明

命令	功能	是否支持
add	新增数据	是
set	设置数据，主要包括新增或者修改数据	是
replace	更新数据	是
append	向指定key的value末尾追加数据	是
prepend	向指定key的value头部追加数据	是
cas	检查并修改数据	是
get	查询数据	是
gets	查询数据详细信息	是
delete	删除数据	是

命令	功能	是否支持
incr	算数增	是
decr	算数减	是
touch	修改数据过期时间	是
quit	断开连接	是
flush_all	清空数据 <b>说明</b> 若提供可选参数delay，则delay取值必须为0。	是
version	查询服务版本信息	是
stats	运行统计信息管理 <b>说明</b> 当前只支持查询基础统计信息，不支持可选参数操作命令。	是
cache_memlimit	设置内存使用限制	否
slabs	查询内部存储结构使用情况	否
lru	数据过期删除策略管理	否
lru_crawler	数据过期删除线程管理	否
verbosity	日志级别管理	否
watch	运行事件监控	否

## 二进制协议

Memcached二进制协议将命令及所操作的内容按照特定结构进行编码后发送，通过预定的字节串表示命令。

Memcached二进制协议与文本协议相比，功能更多，支持SASL认证更加安全，但可用的客户端数量较少。

DCS Memcached实例对二进制协议命令的支持情况如表7-49所示。

表 7-49 Memcached 实例二进制协议支持的命令说明

命令编码	命令	功能	是否支持
0x00	GET	查询数据	是
0x01	SET	设置数据，主要包括新增或者修改数据。	是
0x02	ADD	新增数据	是
0x03	REPLACE	更新数据	是

命令编码	命令	功能	是否支持
0x04	DELETE	删除数据	是
0x05	INCREMENT	算数增	是
0x06	DECREMEN T	算数减	是
0x07	QUIT	断开连接	是
0x08	FLUSH	清空数据 <b>说明</b> 若提供可选参数delay参数，则delay取值必须为0。	是
0x09	GETQ	查询数据，在出现错误时不返回任何信息。	是
0x0a	NOOP	空操作，相当于ping。	是
0x0b	VERSION	查询服务版本信息	是
0x0c	GETK	查询数据并返回key	是
0x0d	GETKQ	查询数据并返回key，在出现错误时不返回任何信息。	是
0x0e	APPEND	向指定key的value末尾追加数据	是
0x0f	PREPEND	向指定key的value头部增加数据	是
0x10	STAT	查询缓存实例的统计信息 <b>说明</b> 当前只支持查询基础统计信息，不支持可选参数操作命令。	是
0x11	SETQ	设置数据，主要包括新增或者修改数据。 在成功时不返回任何信息。	是
0x12	ADDQ	新增数据，在成功时不返回任何信息。	是
0x13	REPLACEQ	更新数据，在成功时不返回任何信息。	是
0x14	DELETEQ	删除数据，在成功时不返回任何信息。	是
0x15	INCREMENT Q	算数增，在成功时不返回任何信息	是
0x16	DECREMEN TQ	算数减，在成功时不返回任何信息	是
0x17	QUITQ	断开连接	是
0x18	FLUSHQ	清空数据并不返回任何信息 <b>说明</b> 若提供delay参数，则delay取值必须为0。	是

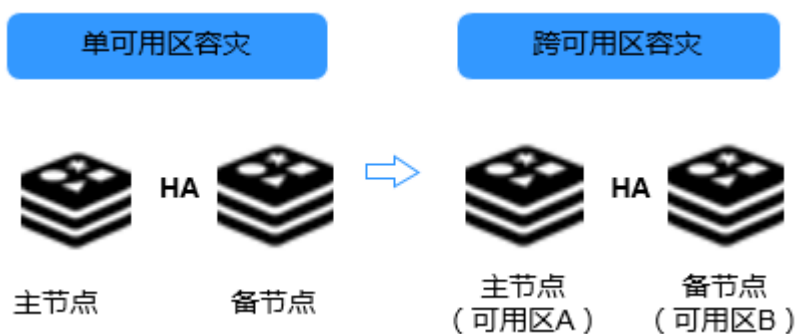
命令编码	命令	功能	是否支持
0x19	APPENDQ	向指定key的value末尾追加数据，在成功时不返回任何信息	是
0x1a	PREPENDQ	向指定key的value头部增加数据，在成功时不返回任何信息	是
0x1c	TOUCH	修改数据过期时间	是
0x1d	GAT	查询数据同时修改过期时间	是
0x1e	GATQ	查询数据同时修改过期时间，在操作失败时不返回任何信息。	是
0x23	GATK	查询数据并返回key，同时修改过期时间。	是
0x24	GATKQ	查询数据并返回key，同时修改过期时间，在操作失败时不返回任何信息。	是
0x20	SASL_LIST_MECHS	获取服务端支持的SASL认证机制	是
0x21	SASL_AUTH	SASL认证	是
0x22	SASL_STEP	SASL复杂认证后续步骤	是



# 8 容灾和多活策略

DCS缓存实例都存储着大量关键数据，不论是作为数据库前端缓存，还是作为数据存储引擎，数据的可靠性与服务的连续可用性是DCS服务设计上为客户考虑的核心因素，下图展示了DCS在数据和服务方面的容灾架构设计演进。

图 8-1 DCS 容灾架构演进



根据对数据与服务不同可靠性要求，您可以选择将缓存实例部署在单可用区内（单机房），或者跨可用区（同城灾备）。

## 实例单 Region 单可用区高可用

同一机房即单可用区。单可用区灾备策略主要包括进程/服务高可用，数据持久化到磁盘，以及实例节点间热备三种不同层次。

在单可用区内，单机实例通过进程守护的方式确保服务高可用，当DCS监测到缓存实例进程故障，马上拉起一个新的进程继续提供服务。

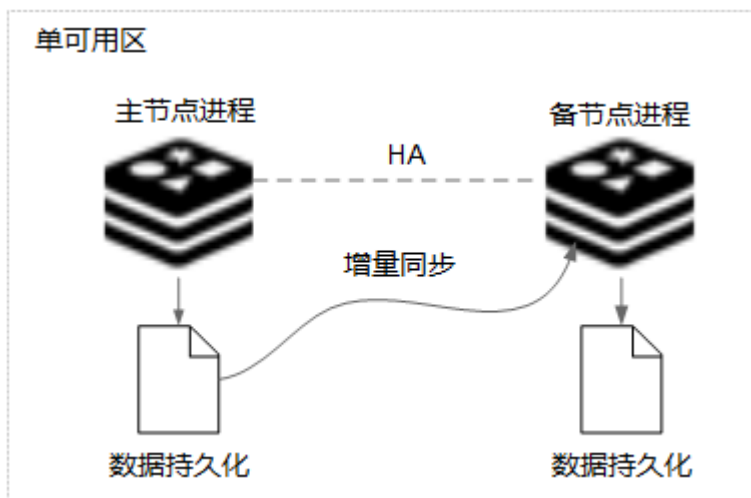
图 8-2 单可用区内单机实例高可用



除单机实例外，主备、读写分离和集群实例默认都支持数据持久化，数据不仅会持久化到实例主节点磁盘，还会增量同步到备节点，同时备节点也会持久化一份数据。实现了节点热备和持久化文件多个备份。

主备、读写分离实例的主备节点进程，以及集群实例每个分片内主备节点进程的数据同步和持久化方式如下图所示。

图 8-3 单可用区内主备节点间高可用



### 实例单 Region 跨可用区灾备

除单机实例外，其他类型的缓存实例都支持将主备节点部署在不同的可用区内（即不同的物理机房）。不同可用区的电力、网络相互隔离，当主节点所在的机房因为电力或者网络出现故障，备节点将接管服务，客户端与备节点正常建立连接以及读写数据。

图 8-4 主备实例跨可用区示意图

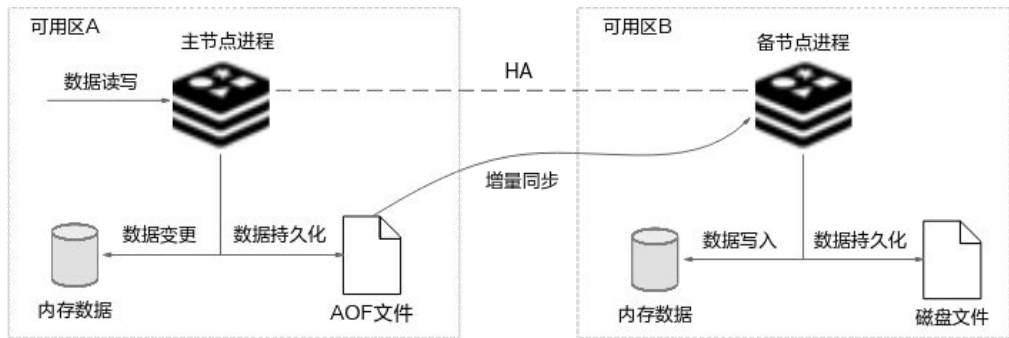


图 8-5 读写分离实例跨可用区示意图

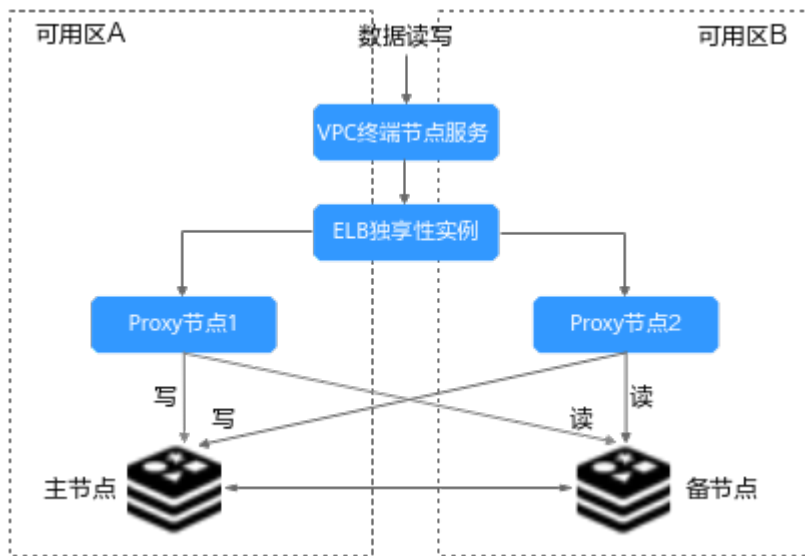


图 8-6 Proxy 集群实例跨可用区示意图

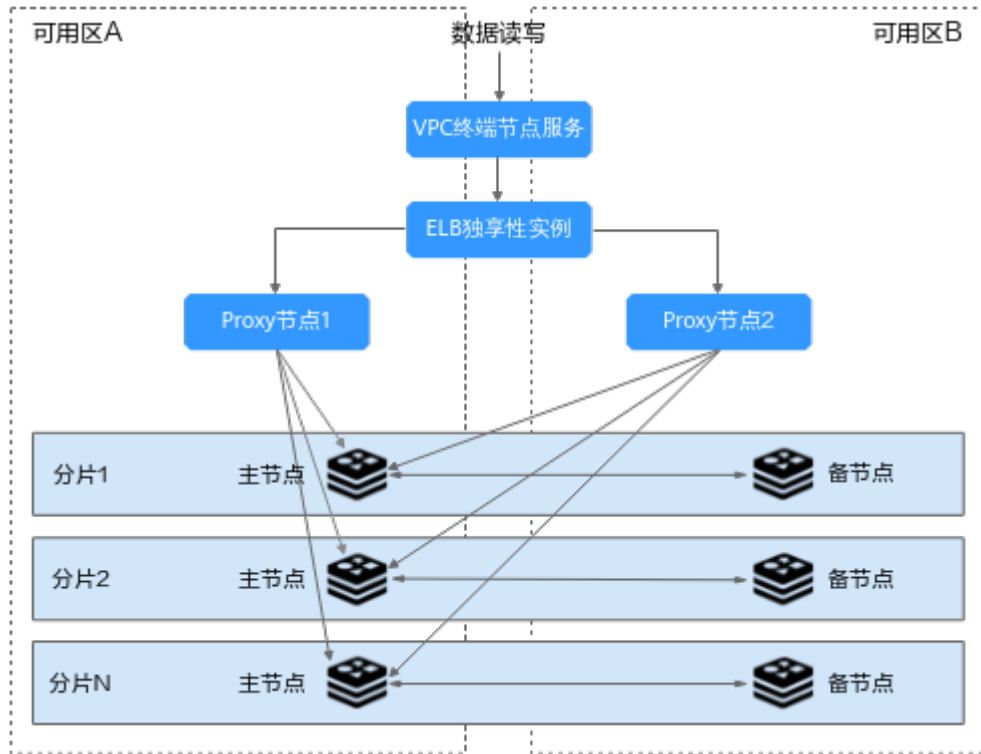
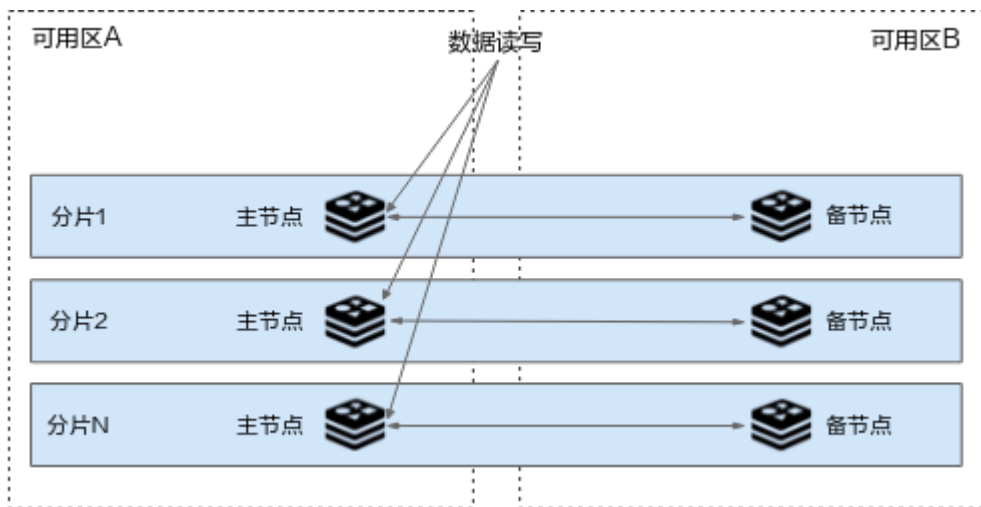


图 8-7 Cluster 集群实例跨可用区示意图



对于同城容灾，只需要在创建主备/集群/读写分离实例时，选择与主可用区不同的备可用区。如下图所示：

图 8-8 购买实例选择不同可用区部署



**说明**

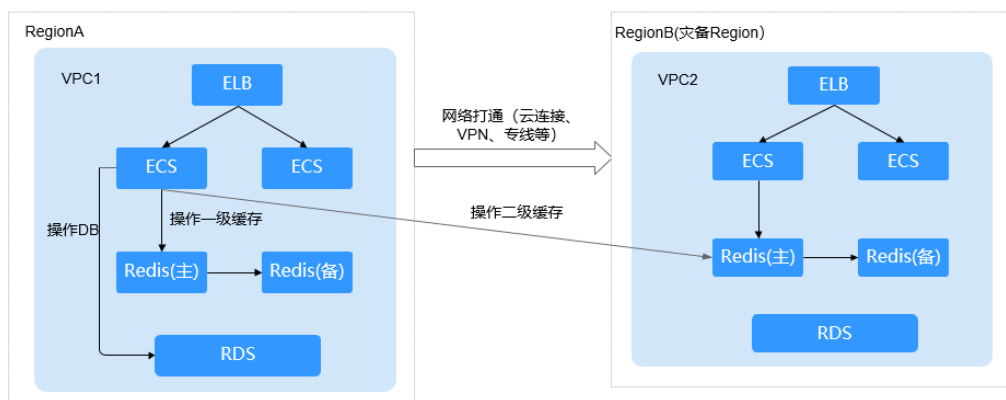
您的应用也可以部署为跨可用区的HA模式，这样不仅能保证数据高可靠，还能在机房遇到断电或网络故障时，服务继续可用。

**实例跨 Region 多活**

当前Redis没有成熟的双活（**Active-Active**，注意不同于灾备或者主备高可用）方案，华为云DCS不支持跨Region多活功能。

这是由于私有定制Redis通信协议（RESP协议）的标准无法统一等原因无法实现跨云/Region的Redis侧双活，因此，如果需要双活，可以通过**应用侧进行双写改造**实现。

图 8-9 应用侧双写实现多活



说明：

1. 双写方案**无法保证缓存的一致性**（网络出现问题），需要**应用侧可以容忍对应的缓存不一致场景**（通过设置过期时间等达到最终一致性），应用如果依赖缓存强一致性，则不合适该方案，跨Region缓存强一致性当前业界没有方案可以实现。
2. 建议操作跨Region的二级缓存采用**异步的方式**。

# 9 产品规格差异

## 9.1 Redis 版本差异

DCS在创建实例时，Redis可选择“版本号”、“实例类型”。

### 📖 说明

- DCS Redis 3.0已下线，暂停售卖，建议使用Redis 4.0/5.0/6.0。
- 由于Redis不同版本的底层架构不一样，在创建Redis实例时，确定Redis版本后，将不能修改，如Redis 4.0暂不支持升级到Redis 5.0或者Redis 6.0。如果需要由低版本升级到高版本，建议重新购买高版本实例，然后进行数据迁移。
- **版本号**  
Redis版本号有3.0，4.0，5.0，6.0，它们的区别如[表9-1](#)。更多Redis特性，请参考[Redis4.0新特性介绍](#)、[Redis5.0新特性介绍](#)和[Redis6.0新特性介绍](#)。

**表 9-1** 不同版本支持的特性、性能差异说明

比较项	Redis 3.0 (已停售)	Redis 4.0 & Redis 5.0	Redis 6.0
兼容开源版本	Redis 3.0兼容开源3.0.7版本	Redis 4.0兼容开源4.0.14版本，Redis 5.0兼容开源5.0.9版本	Redis 6.0基础版兼容开源6.2.7版本 企业版兼容开源Redis 6
实例部署模式	采用虚拟机部署	在物理机上容器化部署	基础版在物理机上容器化部署 企业版采用虚拟机部署
创建实例耗时	3~15分钟，集群约10~30分钟	约8秒	基础版：约8秒 企业版：5~15分钟

比较项	Redis 3.0 (已停售)	Redis 4.0 & Redis 5.0	Redis 6.0
QPS	单节点约10万QPS 具体每种实例规格对应的QPS请参考 <a href="#">Redis 3.0实例 (已停售)</a>	单节点约10万QPS 具体每种实例规格对应的QPS请参考 <a href="#">Redis 4.0/5.0实例</a>	基础版单节点约10万QPS 企业版单节点约40万QPS 具体每种实例规格对应的QPS请参考 <a href="#">Redis 6.0实例</a>
公网访问	支持	暂不支持	暂不支持
域名连接	支持VPC内使用域名连接	支持VPC内使用域名连接	支持VPC内使用域名连接
可视化数据管理	不支持	提供Web CLI访问Redis, 管理数据	提供Web CLI访问Redis, 管理数据
实例类型	支持单机、主备、Proxy集群	支持单机、主备、读写分离、Proxy集群、Cluster集群	基础版: 单机、主备、读写分离、Proxy集群、Cluster集群 企业版: 主备
扩容/缩容	支持在线扩容和缩容	支持在线扩容和缩容	支持在线扩容和缩容
备份恢复	主备和Proxy集群实例支持	主备、Proxy集群、Cluster集群和读写分离实例支持	主备、Proxy集群、Cluster集群和读写分离实例支持

- **实例类型**

Redis实例类型分为单机、主备、Proxy集群、Cluster集群和读写分离，它们的架构与应用场景，请参考[DCS实例类型](#)。

## 9.2 Redis 企业版与基础版差异

DCS企业版为全自研的版本，100%兼容开源Redis软件，由传统的单master-单worker线程模型升级为master-N\*worker线程模型，每个worker线程都可承担端口监听（监听建立网络链接的请求）、accept请求（接纳并建立网络链接）、读取数据（从网络链接如socket读写数据）和解析协议（解析并处理Redis命令等），依托于服务器计算能力的提升，整体性能N倍提升。

表 9-2 企业版和基础版的能力对比

对比项	DCS基础版	DCS企业版
兼容开源Redis版本	兼容开源Redis 4.0/5.0/6.0, 单线程	兼容开源Redis 6.0, 多线程

对比项	DCS基础版	DCS企业版
性能	单分片QPS达10万/秒，最大时延1ms	企业版高性能型：单分片QPS达40万/秒 企业版存储型：单分片QPS达7万/秒 最大时延1ms
实例规格	提供单机、主备、集群、读写分离多种实例类型： <ul style="list-style-type: none"> <li>• 单机、主备实例可选择128MB-64G缓存容量</li> <li>• 读写分离实例可选择8G-32G多种缓存容量</li> <li>• 集群实例最大规格达2048GB</li> </ul>	目前仅提供主备实例类型，可选择 <b>企业版高性能型</b> 和 <b>企业版存储型</b> ： <ul style="list-style-type: none"> <li>• 企业版高性能型可选择8GB-64GB内存容量</li> <li>• 企业版存储型可选择8GB-32GB内存容量，最大存储容量(SSD磁盘容量)可达256GB</li> </ul> 更多企业版高性能型和存储型的差异请参考 <a href="#">表9-3</a> 。
数据安全	<ul style="list-style-type: none"> <li>• 支持细粒度授权和IP白名单</li> <li>• 除单机实例外，均支持持久化和数据备份</li> <li>• 支持跨AZ容灾</li> <li>• 支持异常自动切换</li> <li>• 支持一键在线扩容和变更缓存类型</li> </ul>	<ul style="list-style-type: none"> <li>• 支持细粒度授权和安全组配置</li> <li>• 支持持久化和数据备份</li> <li>• 支持跨AZ容灾</li> <li>• 支持异常自动切换</li> <li>• 支持一键在线扩容</li> </ul>

表 9-3 企业版高性能型和存储型的差异

对比项	企业版高性能型	企业版存储型
QPS	单分片QPS达40万/秒，性能高于存储型。	单分片QPS达7万/秒。
存储方式	不支持SSD存储，纯内存。使用aof的方式做数据持久化。	内存+SSD磁盘。使用内存缓存热数据，全量数据存储存储在SSD磁盘中。
支持的备份文件格式	aof、rdb。	rdb。



# 10 与开源服务的差异

DCS提供单机、主备、集群等丰富的实例类型，满足用户高读写性能及快速数据访问的业务诉求。支持丰富的实例管理操作，帮助用户省去运维烦恼。用户可以聚焦于业务逻辑本身，而无需过多考虑部署、监控、扩容、安全、故障恢复等方面的问题。

DCS基于开源Redis、Memcached向用户提供一定程度定制化的缓存服务，因此，除了拥有开源服务缓存数据库的优秀特性，DCS提供更多实用功能。

## 与开源 Redis 差异

表 10-1 DCS 与自建开源 Redis 的差异说明

比较项	开源Redis	DCS Redis
服务搭建	从自行准备服务器资源到Redis搭建，需要0.5~2天。	<ul style="list-style-type: none"> <li>Redis 3.0和Redis 6.0企业版，采用虚拟机部署，版本5~15分钟完成创建。</li> <li>Redis 4.0、Redis 5.0及Redis 6.0基础版，采用容器化部署，8秒完成创建。</li> </ul>
版本	-	深度参与开源社区，及时支持最新Redis的版本。目前支持Redis 3.0、Redis 4.0、Redis 5.0及Redis 6.0版本。
安全	自行保证网络与服务器的安全。	<ul style="list-style-type: none"> <li>使用华为云虚拟私有云与安全组，确保网络安全。</li> <li>主备与集群多副本、定时备份，确保数据高可靠。</li> </ul>
性能	-	单节点达10万QPS（Query Per Second），Redis 6.0单节点达40万QPS。

比较项	开源Redis	DCS Redis
监控	提供简单的信息统计。	<p>提供30余项监控指标，并支持用户自定义监控阈值和告警策略。</p> <ul style="list-style-type: none"> <li>● 指标类型丰富                             <ul style="list-style-type: none"> <li>- 常见的外部业务监控和统计：命令数、并发操作数、连接数、客户端数、拒绝连接数等。</li> <li>- 常见的资源占用监控和统计：cpu占用率、物理内存占用、网络输入/输出流量等。</li> <li>- 常见的关键内部监控和统计：键个数、键过期个数、容量占用量、pubsub通道个数、pubsub模式个数、keyspace命中、keyspace错过。</li> </ul> </li> <li>● 自定义监控阈值及告警 提供基于各项监控制定阈值告警，支持客户自定义，便于及时发现业务异常。</li> </ul>
备份恢复	支持。	<ul style="list-style-type: none"> <li>● 提供定时与手动备份数据能力，支持备份文件下载到本地。</li> <li>● 支持控制台一键恢复数据。</li> </ul>
可视化维护缓存参数	不具备，需要自行开发。	<ul style="list-style-type: none"> <li>● web控制台可视化维护。</li> <li>● 可在线修改配置参数。</li> <li>● 支持在web控制台连接并操作数据。</li> </ul>
可扩展性	需要中断服务。首先为服务器调整运行内存，然后调整Redis内存配置并重启操作系统与服务。	<ul style="list-style-type: none"> <li>● 提供不中断服务的在线扩容或缩容能力。</li> <li>● 规格可根据实际需要，在DCS支持的规格范围内进行扩容或者缩容。</li> </ul>
易运维	人工运维。	完善的运维服务配套，7*24小时监控与运维保障。

## 与开源 Memcached 差异

表 10-2 DCS 与自建开源 Memcached 的差异说明

比较项	开源Memcached	DCS Memcached
服务搭建	从自行准备服务器资源到Memcached搭建，需要0.5~2天。	5~15分钟完成创建。

比较项	开源Memcached	DCS Memcached
安全	自行保证网络与服务器的安全。	<ul style="list-style-type: none"> <li>使用华为云虚拟私有云与安全组，确保网络安全。</li> <li>主备与集群多副本、定时备份，确保数据高可靠。</li> </ul>
性能	-	单节点达10万QPS ( Query Per Second ) 。
监控	提供简单的信息统计。	<p>提供30余项监控指标，并支持用户自定义监控阈值和告警策略。</p> <ul style="list-style-type: none"> <li>指标类型丰富                             <ul style="list-style-type: none"> <li>常见的外部业务监控和统计：命令数、并发操作数、连接数、客户端数、拒绝连接数等。</li> <li>常见的资源占用监控和统计：cpu占用率、物理内存占用、网络输入/输出流量等。</li> <li>常见的关键内部监控和统计：键个数、键过期个数、容量占用量、pubsub通道个数、pubsub模式个数、keyspace命中、keyspace错过。</li> </ul> </li> <li>自定义监控阈值及告警 提供基于各项监控制定阈值告警，支持客户自定义，便于及时发现业务异常。</li> </ul>
备份恢复	不支持。	<ul style="list-style-type: none"> <li>提供定时与手动备份数据能力。</li> <li>支持控制台一键恢复数据。</li> </ul>
可视化维护	不具备，需要自行开发。	<ul style="list-style-type: none"> <li>web控制台可视化维护。</li> <li>可在线修改配置参数。</li> </ul>
可扩展性	需要中断服务。首先为服务器调整运行内存，然后调整Memcached内存配置并重启操作系统与服务。	<ul style="list-style-type: none"> <li>提供在线不断服务的扩容能力。</li> <li>规格可根据实际需要，在DCS支持的规格范围内进行扩容或者缩容。</li> </ul>
易运维	人工运维。	完善的运维服务配套，7*24小时监控与运维保障。
持久化	不支持。	主备实例支持持久化。

# 11 约束与限制

## Redis 实例

表 11-1 Redis 实例约束与限制

限制项	约束与限制
实例版本	DCS目前支持的Redis版本为：Redis 3.0（已停售）、Redis 4.0、Redis 5.0、和Redis 6.0。 Redis实例创建后，不支持直接升级实例版本，可以通过数据迁移的方式将低版本实例中的数据迁移到高版本。
数据安全	<ul style="list-style-type: none"> <li>Redis 3.0和Redis 6.0企业版：通过配置安全组访问规则控制，不支持白名单功能。</li> <li>Redis 4.0/Redis 5.0和Redis 6.0基础版：通过白名单控制，不支持安全组。</li> <li>Redis 6.0基础版：支持开启SSL链路加密传输，其他版本暂不支持。</li> </ul>
数据持久化	<ul style="list-style-type: none"> <li>单机实例：不支持数据持久化。</li> <li>主备、读写分离和集群实例（单副本集群除外）：默认支持数据持久化。</li> </ul>
读写分离	<ul style="list-style-type: none"> <li>读写分离实例：默认为从服务端实现的读写分离。</li> <li>Cluster集群、主备实例：支持从客户端实现读写分离，需要用户在客户端做配置。</li> <li>其他实例暂不支持读写分离。</li> </ul>
备份数据	单机实例不支持控制台自动或手动备份数据外，其他实例类型支持。
修改实例虚拟私有云和子网	实例创建后，不支持修改实例的虚拟私有云和子网。

## 变更实例

表 11-2 变更实例约束与限制

限制项	约束与限制
变更Redis实例规格/类型	<ul style="list-style-type: none"> <li>建议在业务低峰期变更实例，业务高峰期变更实例可能会失败。</li> <li>副本数变更和容量变更不支持同时进行，需分开两次执行变更。</li> <li>删除副本时，每次操作仅支持删除一个副本。</li> <li>更多变更实例规格/类型的限制与须知请参考：<a href="#">变更DCS实例规格</a>。</li> </ul>
调整DCS实例带宽	<ul style="list-style-type: none"> <li>企业版Redis暂不支持调整带宽。</li> <li>只有在运行中的实例支持调整带宽，如果是变更中、故障中、重启中等其他状态下的实例不支持调整实例带宽。</li> <li>带宽的调整范围在实例基准带宽到最大可调整的带宽之间，通常在实例节点所在物理机资源充足的前提下，实例可调整的单分片最大带宽为2048 Mbit/s。</li> </ul>
变更DCS集群实例为多可用区	<p>仅副本数<math>\geq 2</math>的单可用区集群实例支持该功能，其他场景均不支持变更实例可用区。</p> <ul style="list-style-type: none"> <li>升级Proxy集群实例可用区须知： <ul style="list-style-type: none"> <li>迁移可用区过程中会有秒级业务中断，建议在业务低峰时进行变更。</li> <li>请确保您的客户端应用具备重连机制和处理异常的能力，否则在迁移可用区后有可能需要重启客户端应用。</li> </ul> </li> <li>升级Cluster集群实例可用区须知： <ul style="list-style-type: none"> <li>迁移可用区过程不会影响主节点，业务不会中断，但是性能会略有下降，建议在业务低峰时进行。</li> <li>迁移可用区过程会导致部分副本连接中断，需确保您的客户端应用具备重连机制和处理异常的能力。</li> </ul> </li> </ul>

## 数据迁移

表 11-3 数据迁移约束与限制

限制项	约束与限制
迁移版本	<p>迁移实例时，目标端创建的实例版本不能低于源端Redis版本，将高版本Redis实例数据迁移到低版本Redis实例可能会迁移失败。</p>

限制项	约束与限制
在线迁移	<ul style="list-style-type: none"> <li>● 通过DCS控制台在线迁移Redis实例，必须满足源端Redis和目标Redis网络相通、源端Redis放通SYNC和PSYNC命令两个前提。</li> <li>● 在线迁移不支持公网方式直接迁移。</li> <li>● 源端仅支持Redis 3.0及3.0以上的Redis版本。</li> <li>● 建议在业务低峰期执行在线迁移，否则可能导致源端实例CPU瞬时冲高，时延增大。</li> </ul>
交换源Redis和目标Redis连接地址（交换IP）	<ul style="list-style-type: none"> <li>● 当源实例和目标实例必须都为DCS Redis实例，在执行全量+增量方式的在线迁移后，支持通过控制台交换源Redis和目标Redis的IP和域名地址。</li> <li>● 企业版实例不支持交换IP。</li> <li>● Cluster集群实例不支持交换IP。</li> </ul>

# 12 计费说明

华为云分布式缓存服务支持按需和包周期两种付费模式。具体收费介绍，请参考[价格详情](#)。

## 说明

部分Region不支持包周期的付费方式，请以控制台购买页面的计费模式为准。

## 计费项

华为云DCS按照您选择的DCS实例规格进行计费。

计费项	计费说明
DCS实例	华为云DCS按照您选择的DCS实例规格进行计费

说明：华为云DCS按照您选择的DCS实例规格进行计费，而不是按照实际使用的缓存容量收费。

## 计费模式

提供按小时、按月、按年的计费方式供您灵活选择，使用越久越便宜。

- 预付费（包年包月）：这种购买方式相对于按需付费提供更大的折扣，对于长期使用用户，推荐该方式。
- 按需付费（小时）：这种购买方式比较灵活，可以即开即停，DCS实例从创建成功开启计费到删除结束计费，以秒为单位统计时长，按实际使用时长计费。
- 计费方式更改：支持包周期和按需计费方式转换。

## 变更配置

您可以对Redis和Memcached缓存实例进行规格变更，即扩容/缩容和实例类型变更。实例变更完成后，即刻按照新的实例规格进行计费。更多的规格变更说明，请参考[变更缓存实例规格](#)。

## 续费

为防止资源到期或者浪费，已经购买包年/包月缓存实例的用户，可执行续费操作，延长资源包的有效期，也可以设置到期自动续费。续费的相关操作，请参考[续费管理](#)。

## 计费 FAQ

更多计费相关问题，请参考[DCS常见问题](#)。



# 13 权限管理

如果您需要对华为云上购买的DCS资源，给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，您可以使用统一身份认证服务（Identity and Access Management，简称IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制华为云资源的访问。

通过IAM，您可以在华为云账号中给员工创建IAM用户，并使用策略来控制他们对华为云资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有DCS的使用权限，但是不希望他们拥有删除DCS实例等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用DCS，但是不允许删除DCS实例的权限策略，控制他们对DCS资源的使用范围。

如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用DCS服务的其它功能。

IAM是华为云提供权限管理的基础服务，无需付费即可使用，您只需要为您账号中的资源进行付费。关于IAM的详细介绍，请参见《[IAM产品介绍](#)》。

## DCS 权限

默认情况下，管理员创建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

DCS部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域（如中国-香港）对应的项目（ap-southeast-1）中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问DCS时，需要先切换至授权区域。

权限根据授权精细程度分为角色和策略。

- **角色**：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于华为云各服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- **策略**：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。例如：针对DCS服务，管理员能够控制IAM用户仅能对DCS实例进行指定的管理操作。权限策略以API接口为粒度进行权限拆分，DCS服务支持的API授权项请参见[权限策略支持的授权项](#)。

如表1所示，包括了DCS的所有系统权限。

表 13-1 DCS 系统策略

系统角色/策略名称	描述	类别	依赖关系
DCS FullAccess	分布式缓存服务所有权限，拥有该权限的用户可以操作所有分布式缓存服务的功能。	系统策略	购买企业版实例需要配置如下授权项： iam:permissions:listRolesForAgencyOnProject iam:agencies:listAgenciesiam:roles:listRoles iam:permissions:grantRoleToAgencyOnProject iam:agencies:createAgency iam:agencies:deleteAgency
DCS UserAccess	分布式缓存服务普通用户权限（无实例创建、修改、删除、扩容和缩容的权限）。	系统策略	无
DCS ReadOnlyAccess	分布式缓存服务的只读权限，拥有该权限的用户仅能查看分布式缓存服务数据。	系统策略	无
DCS Administrator	分布式缓存服务管理员权限，拥有该权限的用户可以操作所有分布式缓存服务的功能。	系统角色	依赖Server Administrator和Tenant Guest角色，在同项目中勾选依赖的角色。

系统角色/策略名称	描述	类别	依赖关系
DCS AgencyAccess	<p>分布式缓存服务申请创建租户委托时需要授权的操作权限。</p> <p>该权限为租户委托权限，用于租户在需要时委托DCS服务对租户资源做以下相关操作，与授权用户操作无关。</p> <ul style="list-style-type: none"> <li>• 查询子网</li> <li>• 查询子网列表</li> <li>• 查询端口</li> <li>• 查询端口列表</li> <li>• 更新端口</li> <li>• 创建端口</li> </ul>	系统策略	无

### 说明

由于DCS UserAccess策略和DCS FullAccess策略存在差异，如果您同时配置了这两个系统策略，由于DCS UserAccess策略存在Deny，根据Deny优先原则，您无法执行实例创建、修改、删除、扩容和缩容操作。

表2列出了DCS常用操作与系统策略的授权关系，您可以参照该表选择合适的系统策略。

表 13-2 常用操作与系统策略的关系

操作	DCS FullAccess	DCS UserAccess	DCS ReadOnlyAccess	DCS Administrator
修改实例配置参数	√	√	×	√
删除实例后台任务	√	√	×	√
Web CLI	√	√	×	√
修改实例运行状态	√	√	×	√
缓存实例扩容	√	×	×	√
修改实例访问密码	√	√	×	√
修改缓存实例	√	×	×	√

操作	DCS FullAccess	DCS UserAccess	DCS ReadOnlyAccess	DCS Administrator
实例主备切换	√	√	×	√
备份实例数据	√	√	×	√
分析实例的大key或者热key	√	√	×	√
创建缓存实例	√	×	×	√
删除实例数据备份文件	√	√	×	√
恢复实例数据	√	√	×	√
重置实例访问密码	√	√	×	√
迁移实例数据	√	√	×	√
下载备份实例数据	√	√	×	√
删除缓存实例	√	×	×	√
查询实例配置参数	√	√	√	√
查询实例数据恢复日志	√	√	√	√
查询实例数据备份日志	√	√	√	√
查询缓存实例信息	√	√	√	√
查询实例后台任务	√	√	√	√
查询实例列表	√	√	√	√
操作慢查询	√	√	√	√

## 相关链接

- [IAM产品介绍](#)
- [创建用户并授予使用DCS](#)
- [权限策略和授权项](#)

# 14 基本概念

## 缓存实例

DCS向用户提供服务的最小资源单位。

缓存实例拥有Redis、Memcached两种存储引擎，每种引擎有单机、主备、集群等不同实例类型。不同实例类型含有多种规格。

详情参考：[产品规格介绍](#)，[产品架构介绍](#)

## 公网访问

针对Redis 3.0实例，可以为其绑定一个弹性公网IP地址，客户端通过公网ip进行访问。Redis 4.0、Redis 5.0和Redis 6.0实例支持通过ELB实现公网访问功能。

公网访问使用Stunnel工具对通信内容加密，网络延时比VPC内通信略高，适用于开发阶段联调本地环境测试。

具体使用请参考：[公网连接Redis3.0实例以及Stunnel工具的使用](#)

## 免密访问

缓存实例支持不设置密码，在VPC内直接连接实例进行数据读写。由于不涉及密码鉴权，数据读写延时会更低。

对于实例数据敏感性一般的业务，您可以对实例开启免密访问。出于数据安全考虑，公网访问的实例不允许同时开启免密访问。

具体使用请参考：[配置Redis访问密码（修改Redis实例访问方式）](#)

## 维护时间窗

指允许DCS产品服务团队为实例进行升级维护的时间段。

DCS对实例升级维护频率较低，一般每季度一次。虽然频率低，且升级过程不会影响业务，但建议您选择业务量较少的时间段作为维护时间窗。

在创建实例时，都会要求设置一个维护时间窗，您也可以在实例创建后，对维护时间窗进行修改。

具体使用请参考：[查看和修改DCS实例基本信息](#)

## 跨可用区部署

将主备、集群、或读写分离实例的主、备节点部署在不同的AZ（可用区域）内，节点间电力与网络均物理隔离。您可以将应用程序也进行跨AZ部署，从而达到数据与应用全部高可用。

在创建实例时，可以分别选择主可用区和备可用区。

## 分片

分片也叫**条带**，指Redis集群的一个管理组，对应一个redis-server进程。一个Redis集群由若干条带组成，每个条带负责若干个slot（槽），数据分布式存储在slot中。Redis集群通过条带化分区，实现超大容量存储以及并发连接数提升。

每个集群实例由多个分片组成，每个分片默认为一个双副本的主备实例。分片数等于实例中主节点的个数。

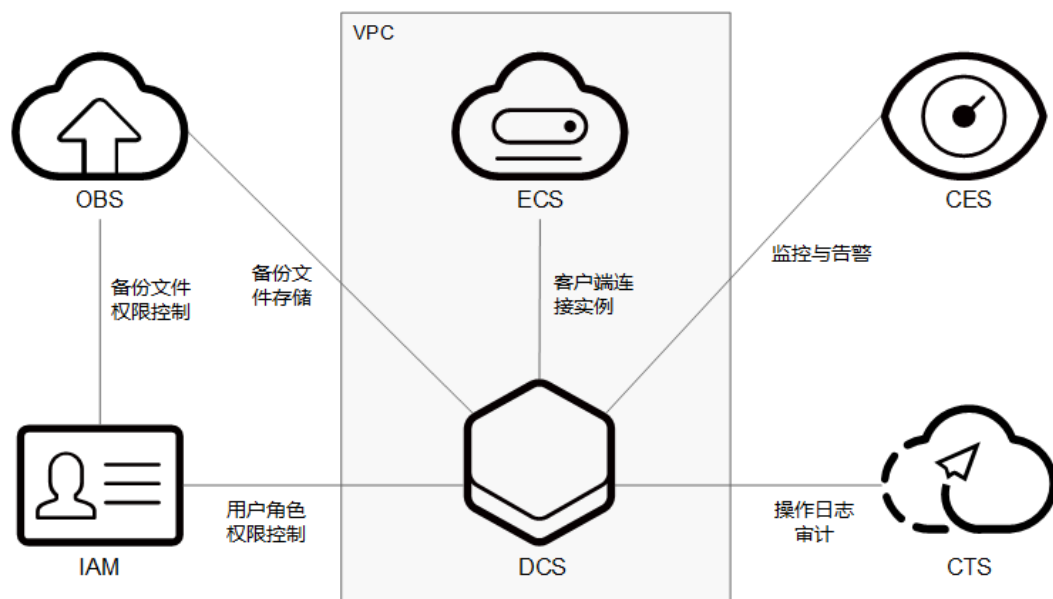
## 副本

副本指缓存实例的**节点**。单副本表示实例没有备节点，双副本表示实例有备节点（一个主节点，一个备节点）。例如主备实例默认为双副本，当主备实例的副本数设置为3时，表示该实例有1个主节点，2个备节点。单机实例，只有一个节点。

# 15 与其他服务的关系

DCS在使用时与华为云其他服务配合使用，本节简单介绍虚拟私有云、弹性云服务器、统一身份认证服务、云监控服务、云审计服务以及对象存储服务。

图 15-1 DCS 缓存服务与其他服务的关系



## 虚拟私有云

虚拟私有云（Virtual Private Cloud，简称VPC）是用户在华为云上申请的隔离的、私密的虚拟网络环境。用户可以自由配置VPC内的IP地址段、子网、安全组等子服务，也可以申请弹性带宽和弹性IP搭建业务系统。

分布式缓存服务运行于虚拟私有云，由虚拟私有云协助管理IP和带宽。虚拟私有云还具备安全组访问控制功能，通过绑定安全组并设置访问规则，可以增强访问分布式缓存服务的安全性。

## 弹性云服务器

弹性云服务器（Elastic Cloud Server，简称ECS）是一种可随时自助获取、可弹性伸缩的云服务器，帮助用户打造可靠、安全、灵活、高效的应用环境。



成功申请分布式缓存服务后，您可以通过弹性云服务器创建的弹性云主机，连接和使用分布式缓存实例。

## 统一身份认证服务

统一身份认证（Identity and Access Management，简称IAM）是公有云系统的身份管理服务，包括用户身份认证、权限分配、访问控制等功能。

通过统一身份认证服务，实现对分布式缓存服务的访问控制。

## 云监控服务

云监控服务（Cloud Eye）是公有云提供的安全、可扩展的统一监控方案，通过云监控服务集中监控DCS的各种指标，基于云监控服务实现告警和事件通知。

## 云审计服务

云审计服务（Cloud Trace Service，简称CTS），为您提供云服务资源的操作记录，记录内容包括您从公有云管理控制台或者开放API发起的云服务资源操作请求以及每次请求的结果，供您查询、审计和回溯使用。

## 对象存储服务

对象存储服务（Object Storage Service，简称OBS）是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力，包括：创建、修改、删除桶，上传、下载、删除对象等。

DCS使用OBS存储实例数据备份文件。