

SCS 文档



【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

【免责声明】

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。如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

目录	
目录	2
动态与公告	8
新功能发布记录	8
Redis内存型小版本发布日志	11
Redis磁盘型（PegaDB）小版本发布日志	15
产品简介	16
什么是云数据库 Redis	16
云数据库 Redis 版初识与选型	18
Redis 容量型(原 PegaDB)产品介绍	18
云数据库 Redis 版与自建 Redis 的对比	19
Redis 容量版与 Redis 社区版的特性对比	20
账号管理	68
节点变更	74
安全管理	77
查看安全组规则列表	82
数据迁移	88
变更实例	89
版本升级	98
计费变更	100
监控报警	104
报警通知	109
参数管理	110
日志管理	117
SmartDBA	121
会话筛选	128
kill 所选会话	129
会话历史	129
会话重置。	129
用户维度统计	130
访问来源维度统计	130
访问来源维度统计	130
多用户访问控制	135
标签管理	137
备份管理	141
只读组管理	149
任务管理	150
导出实例列表	154
创建热活实例组	154
创建多活实例组	158
实例状态与影响	162

Baidu 百度智能云文档	目录
Redis 实例生命周期流程	163
迁移可用区	164
IP变更	166
Memcached操作指南	168
实例生命周期	168
计费变更	172
变更实例	177
监控报警	180
安全管理	185
Redis 容量型(原 PegaDB)操作指南	188
监控报警	188
基本信息	190
策略规则	190
报警通知	190
账号管理	193
安全管理	198
安全组规则列表	203
节点变更	206
实例生命周期	207
计费变更	213
变更实例	218
SmartDBA	226
会话筛选	234
kill 所选会话	234
会话历史	234
会话重置。	235
用户维度统计	235
访问来源维度统计	235
访问来源维度统计	236
备份管理	239
概述	239
参数管理	243
操作步骤	243
版本升级	244
性能调优	245
如何处理Redis集群数据倾斜	245
排查Redis实例流量使用率高的问题	246
常见Latency（时延）事件的处理建议	247
排查Redis实例内存使用率高的问题	248
排查Redis实例CPU使用率高的问题	251

Baidu 百度智能云文档	目录
API参考	253
API简介	254
使用须知	254
通用说明	254
错误返回	256
API规格	258
订单服务	258
创建实例	258
变更配置	262
释放实例	264
获取创建实例价格	265
获取实例变更配置价格	267
后付费转预付费API	269
预付费转后付费API	270
取消预付费转后付费API	271
续费API	272
请求说明	272
实例服务	273
获取实例列表	273
获取实例详情	275
修改实例名称	278
修改实例域名	279
重启实例API	280
清空实例API	281
获取实例规格列表	282
获取可用区列表	285
获取子网列表	286
绑定标签	287
解绑标签	288
从回收站中批量恢复实例	288
释放回收站中的实例	289
获取回收站内的实例列表	290
集群状态检查	291
统一读入口负载均衡	293
创建统一读入口	294
拨测接口	295
主从切换API	296
修改维护时间窗口API	297
标准版升级集群版API	298
请求说明	298
迁移可用区API	300

Baidu 百度智能云文档	目录
IP变更API	302
版本升级API	303
账号管理	304
账号列表	304
修改账号密码	305
设置权限	306
删除账号	307
创建账号	308
参数管理	309
获取参数列表	309
修改参数	311
备份恢复	312
查看备份列表API	312
获取备份策略API	314
设置备份策略API	314
删除手动备份API	315
请求说明	315
获取备份的URL API	316
修改备份备注API	317
请求说明	317
手动备份API	318
请求说明	318
网络安全	318
查询IP白名单	318
增加IP白名单	319
删除IP白名单	320
修改访问密码	321
域名交换	322
热活实例组	323
前置检查	323
创建热活实例组	326
获取热活实例组列表	327
热活实例组详情	328
释放热活实例组	330
添加从集群	331
移除从集群	332
修改热活实例组名称	332
变更主角色	333
实例组禁写	334
设置流控规则	335
获取从角色同步状态	335

Baidu 百度智能云文档	目录
获取IP白名单列表	336
添加IP白名单	337
删除IP白名单	338
设置从角色脏读	339
多活实例组	340
多活实例组删除	340
获取同步状态	340
修改多活实例组名称	342
多活实例组详情	342
多活实例组列表	345
前置检查	347
创建多活实例组	350
参数模版	352
创建参数模版	352
获取参数模版列表	354
参数模版详情	356
删除参数模版	358
修改参数模版名称	359
应用参数模版	360
参数模版添加参数	361
参数模版修改参数	362
参数模版删除参数	363
获取系统参数列表	364
获取应用参数模版记录	369
日志管理	371
审计日志开关API	371
日志明细API	372
日志列表API	372
安全管理	374
TDE加密	374
查看安全组API	376
解绑安全组API	378
绑定安全组API	379
更新安全组API	380
获取TLS证书API	381
开启或关闭TLS加密API	382
附录	383
部署集	384
创建部署集API	385
删除部署集API	386
修改部署集API	386

获取部署集列表API	387
SDK	388
GO-SDK	388
概述	388
安装SDK工具包	388
初始化	389
主要接口	392
异常处理	400
Redis使用限制	401
背景信息	401
其他说明	401
Redis命令使用限制	401
Redis容量型（PegaDB）使用命令限制	414
概述	414
Memcached使用限制	427
Memcached-使用限制	427
最佳实践	428
Redis设计规范与最佳实践	428
避免大 Key 热 Key 实践指南	429
消息发布订阅	435
视频专区	436
操作指南	436
常见问题	437
性能类问题	437
安全管控保障	437
费用相关问题	438
Lua脚本使用规范与注意事项	439
常见报错与解决方案	441
企业上云评估参考	442
服务等级协议SLA	445
Redis服务等级协议SLA	445
云数据库Redis服务条款	446
百度智能云可信云	450
典型实践	453
客户端配置注意事项	453
Redis Pipeline 机制介绍	456
消息发布与订阅	457
Redis 事务使用注意事项	458
版本说明	459

动态与公告

新功能发布记录

本文介绍百度云数据库 Redis的产品功能动态及对应的文档。

🔗 Redis 内核版本更新

- [Redis内存型小版本发布日志](#)
- [Redis磁盘型（PegaDB）小版本发布日志](#)

🔗 Redis 功能发布更新

发布时间	功能分类	产品	功能名称	相关文档	发布地域
2025-5-23	新功能	云数据库 Redis	Redis内存型支持清空过期数据	Redis清空过期数据	全部
2025-5-9	新功能	云数据库 Redis	PegaDB支持本地盘	创建PegaDB实例	边缘地区不支持
2025-4-17	新功能	云数据库 Redis	Redis内存型支持按时间点恢复数据	按时间点恢复数据	边缘地区不支持
2025-4-1	新功能	云数据库 Redis	PegaDB支持升级小版本	小版本升级	边缘地区不支持
2025-4-1	新功能	云数据库 Redis	PegaDB支持实例重启	——	边缘地区不支持
2025-3-13	新功能	云数据库 Redis	控制台搜索实例支持按实例名称、实例ID批量模糊匹配（不同字段使用空格间隔）	——	全部
2025-3-13	新功能	云数据库 Redis	Redis支持大版本升级	大版本升级	边缘地区不支持
2025-2-27	新功能	云数据库 Redis	分片数量支持自定义（步长为1）	——	全部
2024-12-3	新功能	云数据库 Redis	PegaDB支持参数管理	参数配置	边缘地区不支持
2024-11-19	新功能	云数据库 Redis	实例概览页面	——	全部
2024-11-5	新功能	云数据库 Redis	支持迁移可用区	迁移可用区	边缘地区不支持
2024-10-8	新功能	云数据库 Redis	PegaDB支持数据备份	数据备份	边缘地区不支持
2024-7-31	新功能	云数据库 Redis	精细化任务管理	任务管理	边缘地区不支持
2024-7-25	新功能	云数据库 Redis	支持Redis标准版升级到集群版	标准版升级集群版	上海、南京、阳泉、边缘地区不支持
2024-7-25	新功能	云数据库 Redis	接入智能助手（智能领航员）	智能领航员	全部
2024-7-23	新功能	云数据库 Redis	PegaDB支持多活实例组	多活实例组介绍	边缘地区不支持
2024-5-24	新功能	云数据库 Redis	Redis支持7.0集群版创建	——	全部

2024-5-31	新功能	云数据库 Redis	Redis 支持 7.0 果研版创建	——	全部
2024-5-24	新功能	云数据库 Redis	Redis 支持秒级监控，用户可以在监控控制台配置秒级监控	查看监控	全部
2024-4-19	新功能	云数据库 Redis	新增服务可用性页面，支持用户主动批量主从切换	主从切换	全部
2024-3-22	新功能	云数据库 Redis	参数管理支持搜索功能	——	全部
2024-3-22	新功能	云数据库 Redis	接入 Explorer API 中心	——	全部
2024-3-14	新功能	云数据库 Redis	多活集群支持账号管理功能	——	全部
2024-3-14	新功能	云数据库 Redis	原 PegaDB 更名为 Redis 容量型	——	全部
2024-1-19	新功能	云数据库 Redis	热活实例组可支持查看热活实例监控	——	全部
2023-12-29	新功能	云数据库 Redis	产品规格最小支持 256MB	——	全部
2023-12-29	新功能	云数据库 Redis	Redis 标准版支持 7.0	——	全部
2023-12-15	新功能	云数据库 Redis	组合购上线	——	全部
2023-12-8	新功能	云数据库 Redis	Redis 新增 SCAN 可选禁用命令	——	全部
2023-12-8	新功能	云数据库 Redis	Redis 6.0 支持审计日志	——	全部
2023-12-1	新功能	云数据库 Redis	PegaDB 支持备份功能	——	全部
2023-11-24	新功能	云数据库 Redis	标准版支持TDE加密	——	全部
2023-11-07	新功能	云数据库 Redis	内存监控指标支持单位切换，且支持同时展示多实例监控	——	全部
2023-11-03	新功能	云数据库 Redis	支持创建集群版实例	——	边缘节点
2023-10-27	新功能	云数据库 Redis	创建实例可自动加入BCM实例组	——	全部
2023-09-08	新功能	云数据库 Redis	副本数上限增加到10副本	——	全部
2023-07-28	新功能	云数据库 Redis	新增24G规格套餐	Redis标准版 Redis集群版	全部
2023-06-30	新功能	云数据库 Redis	支持单副本集群	创建支持单副本	全部
2023-05-19	新功能	云数据库 Redis	支持bloomfilter	——	全部
2023-05-12	新功能	云数据库 Redis	容量版支持单独扩容磁盘	——	全部
2023-04-28	新功能	云数据库 Redis	容量版支持大Key分析和热Key分析	大Key分析热Key分析	全部
2023-04-21	新功能	云数据库 Redis	支持事件监控	——	全部
2023-04-07	新功能	云数据库 Redis	支持标准版实例	Redis边缘节点	边缘节点
2023-02-20	新功能	云数据库 Redis	接入SmartDBA，支持大KEY分析	SmartDBA大Key分析	全部
2022-06-30	新功能	云数据库 Redis	支持KV数据库PegaDB	PegaDB产品介绍	全部
2022-04-15	新功能	云数据库 Redis	proxy命令兼容性提升	Proxy命令支持情况	全部
2022-03-25	新功能	云数据库 Redis	回收站优化	实例回收站	全部
2022-03-25	新功能	云数据库 Redis	支持购物车	——	全部
2022-02-25	新功能	云数据库 Redis	支持时间窗口	设置维护窗口	全部
2022-02-10	新功能	云数据库 Redis	支持任务列表	任务列表	全部
2021-03-25	新功能	云数据库 Redis	支持回收站	——	全部
2021-03-03	新功能	云数据库 Redis	支持端口号修改	修改端口	全部

发布时间	功能	云数据库 Redis	支持公网访问	设置公网访问	全部
2021-03-03	新功能	云数据库 Redis	支持公网访问	设置公网访问	全部
2021-02-02	新功能	云数据库 Redis	支持社区版	——	全部
2021-01-08	新功能	云数据库 Redis	支持新存储类型 AEP 型	——	全部
2021-01-08	新版本	云数据库 Redis	支持 Redis 6.0 版本	功能概览	全部
2021-01-05	体验优化	云数据库 Redis	日志管理支持新地域	日志管理	华北-保定、华南-广州、华东-苏州
2020-09-27	体验优化	云数据库 Redis	支持指令更新	——	全部
2020-09-27	体验优化	云数据库 Redis	控制台支持广告条	——	全部
2020-09-23	新功能	云数据库 Redis	支持账号管理	账号管理	全部
2020-08-14	体验优化	云数据库 Redis	实例支持设置中文名称	——	全部
2020-07-21	新功能	云数据库 Redis	集群版支持纵向扩容、缩容	纵向扩缩容	全部
2020-06-24	新功能	云数据库 Redis	支持开启免密访问（清空实例密码）	支持免密访问	全部
2020-06-17	体验优化	云数据库 Redis for Memcache	支持更丰富的实例规格	——	全部
2020-05-20	API	云数据库 Redis	支持自动续费	——	全部
2020-05-15	新功能	云数据库 Redis	支持多副本实例设置读写分离	——	全部
2020-05-15	体验优化	云数据库 Redis	支持添加、删除从节点	——	全部
2020-04-01	体验优化	云数据库 Redis	支持后付费实例降低配置	——	全部
2020-03-27	体验优化	云数据库 Redis	支持重启实例	——	全部
2020-03-05	体验优化	云数据库 Redis	支持大版本升级、小版本升级	——	全部
2020-02-25	体验优化	云数据库 Redis	支持内存使用率监控	——	全部
2020-01-16	新地域/新可用区	云数据库 Redis	华东-苏州 支持可用区C	——	华东-苏州
2020-01-10	新版本	云数据库 Redis	支持慢日志	——	全部
2020-01-10	新版本	云数据库 Redis	支持 2.8/4.0/5.0 版本	——	全部
2020-01-10	新功能	云数据库 Redis	支持异地多活	——	全部
2019-12-18	新功能	云数据库 Redis	支持大 key 分析	——	全部
2019-12-09	新功能	云数据库 Redis	支持手动主从切换	——	全部
2019-11-30	新功能	云数据库 Redis	接入云编排 COS	——	全部
2019-11-20	体验优化	云数据库 Redis	控制台实例列表支持搜索实例ID/实例IP/域名	——	全部
2019-11-20	新地域/新可用区	云数据库 Redis	支持可用区华东-度小满金融专区	——	华东-度小满金融专区
2019-11-20	新功能	云数据库 Redis	支持实例恢复备份	——	全部
2019-10-31	体验优化	云数据库 Redis	支持更丰富的监控项	——	全部
2019-10-18	新功能	云数据库 Redis	集群版支持分级存储型	——	全部
2019-09-06	API	云数据库 Redis	Open API 第二次更新	API参考	全部
2019-09-06	新地域/新可用区	云数据库 Redis	华北-北京 支持可用区E	——	华北-北京

2019-08-30	体验优化	云数据库 Redis	支持基于标签进行授权和鉴权	——	全部
2019-08-09	新地域/新可用区	云数据库 Redis	支持可用区华北-度小满金融专区二区	——	华北-度小满金融专区二区
2019-07-25	新功能	云数据库 Redis	支持单副本实例	——	全部
2019-07-03	新功能	云数据库 Redis	支持后付费计费方式	按需计费	全部
2019-06-21	新地域/新可用区	云数据库 Redis	支持新加坡	——	新加坡
2019-05-30	新功能	云数据库 Redis	支持参数管理	——	全部
2019-05-17	新功能	云数据库 Redis	支持 IP 白名单	设置IP白名单	全部
2019-04-04	新功能	云数据库 Redis	支持设置实例的密码（创建实例、管理实例）	——	全部
2019-04-02	新功能	云数据库 Redis	支持子账户设置运维权限、子用户设置管理权限	——	全部
2019-03-18	新功能	云数据库 Redis	支持自动续费	——	全部
2019-02-28	新功能	云数据库 Redis for Memcach	支持二进制	——	全部
2019-01-29	新地域/新可用区	云数据库 Redis	华北-北京 支持可用区D	——	华北-北京
2018-12-12	API	云数据库 Redis	支持 Open API	API参考	全部
2018-12-05	体验优化	云数据库 Redis	支持 IPv6 地址	——	全部
2018-11-21	新版本	云数据库 Redis	支持 3.2 版本	——	全部
2018-11-07	体验优化	云数据库 Redis	接入工单服务优化	——	全部
2018-09-19	新功能	云数据库 Redis	支持备份	——	全部
2018-09-05	新地域/新可用区	云数据库 Redis	支持武汉金融专区、香港	——	武汉金融专区、香港

Redis内存型小版本发布日志

为提升用户体验，云数据库 Redis 会不定期地发布小版本，丰富Redis已有功能或修复已知缺陷。您可以参阅本文了解Redis社区版小版本的说明介绍，小版本落后可能导致部分功能无法正常使用，建议您可以选择在业务低峰期及时升级实例的小版本。

您可以通过控制台查看当前实例的版本，并参考下文的内容进行对应，了解目前版本已具备的功能。

 Redis内存型（社区版、集群版）

 大版本为7.0版本

小版本号	发布日期	类型	说明
7.2.13	2025-04-28	新增功能	优化 IO 多线程的实现
7.2.12	2025-04-24	Bug修复	- 支持 sentinel 命令。 - 支持半同步。
7.2.11	2025-04-24	Bug修复	修复 opheader 未正确格式化的问题。
7.2.10	2025-04-7	Bug修复	支持 Redis IO 多线程
7.2.9	2024-11-25	Bug修复	支持全局的命令延迟统计，可以在 info stats 中看到。
7.2.8	2024-08-29	Bug修复	限制 hotkey-top-size参数的范围为 1-1024。
7.2.7	2024-08-16	新增功能	- 支持审计日志，说明文档 - 去掉版本号中的 (Baidu Release) 字段。
7.2.6	2024-05-20	Bug修复	修复开启异步 AOF 写入时，关闭 AOF 时采用了同步关闭的问题。
7.2.5	2024-05-14	Bug修复	修复异步写 AOF 死锁的问题
7.2.4	2024-04-15	新增功能	- 热 key 统计优化 - op header - AOF 异步写盘 - 基于 AOF 的同步 - 支持 PSYNC 拉取含部分指定 slot 的 RDB - 支持禁写 - 支持 VPC 内免 auth - 支持打印危险命令 - 支持 config set 限制 localhost
7.2.3	2023-12-11	新增功能	引入开源的7.2.3

🔗 大版本为6.0版本

小版本号	发布日期	类型	说明
6.2.38	2025-02-07	Bug修复	支持 redis 半同步自动降级，修复半同步超时的时候Redis 少回复响应的问题。
6.2.37	2024-10-30	Bug修复	修复启动启动半同步后，命令使用 pipeline 执行时会卡住的问题。
6.2.36	2024-10-12	Bug修复	修复启动bigkey统计后，出 core 的问题。
6.2.35	2024-08-29	Bug修复	修复设置 hotkey-top-size 为 0 时，Redis 会出 core 的问题。
6.2.34	2024-08-16	Bug修复	修复 sentinel 命令中 role-reported-time 字段返回为 integer 类型的错误。

6.2.33	2024-08-16	新增功能	去掉 info 信息中 version 字段中的 baidu release 字符串。
6.2.32	2024-07-03	Bug 修复	修复开启热key统计后，Redis 无法启动的问题。
6.2.31	2024-03-08	新增功能	- 添加 skip-auth-commands。 - 配置项 添加 sentinel-master-name 配置项，用于 fake-sentinel 实现中返回 master 的 name。
6.2.30	2024-04-03	新增功能	- 支持 debug command::block 系列命令，可以设置客户端阻塞在某个 command 上，用于调试。 - 支持 sentinel 命令
		Bug 修复	- 修复 TDE 加密中缓冲区溢出的 bug - 修复 ARM 上编译失败的问题
6.2.29	2023-11-22		无新增功能，使用 GCC12 编译。
6.2.28	2023-11-21	新增功能	支持审计日志。
6.2.27	2023-10-18	新增功能	新增配置项aof-shift-auto-bgsave。
6.2.26	2023-8-7	新增功能	新增配置项enable-remote-client-update-config。
6.2.25	2023-7-19	Bug 修复	- 修复 bigkey 统计打满 CPU 的问题 - 修复 hotkey 统计 coredump 的 bug，优化 hotkey 统计性能
6.2.24	2023-7-4	新增功能	新增Partial RDB，可以指定 Redis 仅仅同步包含指定 slot 的 RDB。
6.2.23	2023-4-25	新增功能	- 支持主从同步force_diskless和diskless nodelay传输选项 - 优化output buffer超限从循环中break https://github.com/redis/redis/pull/11676 - 优化 d2string https://github.com/redis/redis/pull/11779
		Bug 修复	修复forkless blpop出core的问题。
6.2.22	2023-3-16	Bug 修复	修复动态关闭appendonly时可能导致crash的问题。
6.2.21	2022-8-30	Bug 修复	- 兼容 forkless rdb - 兼容Redis6.0的功能: - limit config command - log dangerous command: flushall flushdb keys swapdb

			• optimize cpu Affinity • bug fix:fix forbid write bug when propagate a ping command
6.2.20	2022-8-19	Bug 修复	• 修复forkless解析opheader的bug。 • 修复forkless期间dict fingerprint断言失败的bug。 • 修复禁止重写set命令 bug。
6.2.19	2023-6-1	新增功能	支持禁止set命令重写
6.2.18	2022-7-18	新增功能	新增VPC 内免密访问
6.2.17	2022-7-12	Bug 修复	修复异地多活场景下带有opheader的数据持久化问题
6.2.16	2022-7-12	新增功能	• 将opheader中时间戳从us调整为ms • 支持热 Key 统计
6.2.15	2022-7-8	Bug 修复	修复 crdt 时间单位为us
6.2.14	2022-7-8	Bug 修复	复在主从同步期间主库aof文件中有可能生成select -1的问题，影响cts解析增量数据
6.2.13	2022-6-10	Bug 修复	• 升级redis module api • 修复注册module命令未初始化bug
6.2.12	2022-7-8	新增功能	支持CRDT，兼容4系
6.2.11	2022-7-8	新增功能	• 适配2、3系redis的db，将多个db转换为一个db • forkless2.0 性能优化
6.2.10	2022-4-19	新增功能	支持aof增量同步
6.2.9	2022-4-18	新增功能	支持aof异步化
6.2.8	2022-3-14	Bug 修复	• 支持半同步 • 修复在从库连接上与主库交互的bug
6.2.7	2022-3-14	新增功能	• 增加命令维度的 latency 统计：增加 latency histogram 命令 • 增加全局视角的 latency 统计：增加所有命令 latency 50，95，99，99.9，99.99 分位值
6.2.6	2022-1-26	新增功能	add sanitizer support
6.2.5	2021-12-7	新增功能	新增支持 CAS/CAD 命令的模块

		能	
--	--	---	--

🔗 大版本为5.0版本及以下

Redis官方社区 4.0、5.0已停止维护，功能相对欠缺，推荐使用Redis 6.0、7.0版本。

Redis磁盘型（PegaDB）小版本发布日志

为提升用户体验，**Redis磁盘型（PegaDB）** 会不定期地发布**小版本**，丰富Redis已有功能或修复已知缺陷。您可以参阅本文了解相关说明介绍，小版本落后可能导致部分功能无法正常使用，建议您可以选择在**业务低峰期**及时升级实例的小版本。

您可以通过控制台查看当前实例的版本，并参考下文的内容进行对应，了解目前版本已具备的功能。

🔗 Redis磁盘型（PegaDB）

🔗 版本说明

小版本号	发布日期	类型	说明
3.0.0	2025-3-17	新增功能	- 支持 hyperloglog - 支持 stream - 支持 function - 修复 log dir 失效的问题
2.2.13	2025-3-17	新增功能	- cdc 支持 with_ts 返回内容包含时间戳 - cdc 支持多任务，内部缓存多个 wal 迭代器 - 修复 repl id
2.2.10	2024-12-12	Bug修复	- 严格校验 cdc sequence - 修复 FLUSHALL 和 FLUSHDB 后导致 compaction 内出现字符串访问越界
2.2.9	2024-11-26	Bug修复	- 修复灌库功能删除 namespace 时 sst 文件没有立即删除的问题 - 修复添加热活时主库夯住问题 - 修复主从复制逻辑遗漏的问题（不影响主从数据的一致性）
2.2.8	2024-10-10	新增功能	- 新增参数配置rocksdb.bottommost_compression。 - 新增参数配置rocksdb.delete_obsolete_files_period_sec。 - 修复灌库文件无法体现在 used_dbsize 的 bug。
2.2.5	2024-2-26	Bug修复	- 修复 slow log get 返回值对空字符串的处理，与 redis 对齐。 - 修复 info replication 返回值 master_last_io_seconds_ago 不准确问题。 - 修复rocksdb.blob_garbage_collection_age_cutoff 动态修改不生效。
2.2.4	2024-2-26	新增功能	增加 rocksdb.pin_index_and_filter_blocks_in_cache 配置项，默认为 yes（和旧版本行为一致）。

	26	能	致)
2.2.3	2023-12-20	新增功能	• 数据迁移目标端使用 low priority write, 减少对目标端的影响, 使得迁移更加平缓, migrate-batch-rate-limit-mb 提升至 8M。 • 配置项变更: redis-id, shard-id, cluster-id 支持动态修改。 • 修复数据迁移丢数据的问题。
2.2.2	2023-11-10	新增功能	• 优化 GC 机制: 优化 Compaction Checker : PegaDB GC机制优化。 优化 Compact 命令, 避免执行期间实例夯住。 • 升级 GCC12
2.2.1	2023-8-28	新增功能	• 发生 Background Error 时, 自动保存 corrupt 的 sst 文件。 • info 中新增 in_background_error 项 (值为 yes 或者 no), 用于说明当前是否处于错误状态 (此状态下实例只读状态) 。 • 修复rocksdb 引擎 rate limiter bug。 • 修复内存限制默认值未生效 bug。
2.2.0	2023-7-17	新增功能	• 支持客户端维度内存限制 (output buffer limit) 。 • FlushAll 可以更快速的释放磁盘空间。 • 修复半同步卡住的 Bug。 • 修复 pegadb-tools 增量同步 redis 时的 Bug。 • 修复全量同步重试时, crash 的Bug

产品简介

什么是云数据库 Redis

什么是云数据库Redis

云数据库 Redis (Simple Cache Service for Redis) 提供稳定、高效以及高可扩展性的分布式缓存服务。云数据库 Redis 兼容 Redis/Memcached 协议, 基于 Redis 提供标准版和集群版的架构模式, 并支持自定义副本数量, 为您提供多样化的数据结构支持。云数据库 Redis 双副本实例默认支持主从热备, 实现故障自动切换、故障迁移、数据备份、实例监控等运维管理功能。

为什么选择云数据库Redis

- 相对于自建Redis, 硬件部署在云端, 提供完善的基础设施规划、网络安全保障和系统维护服务, 您可以专注于业务创新。
- 支持String (字符串)、List (链表)、Set (集合)、Sorted Set (有序集合)、Hash (哈希表)、Stream (流数据) 等多种数据结构, 同时支持Transaction (事务)、Pub/Sub (消息订阅与发布) 等高级功能。
- 高可用和灾备: 云数据库Redis通常提供主从复制、哨兵模式或Redis集群等高可用方案, 确保数据的可靠性和业务的连续性。
- 弹性伸缩: 根据业务需求, 用户可以轻松地扩展或缩减Redis实例的规模, 以满足不同负载的需求。

产品架构

架构类型	说明
Redis容量型(原PegaDB)	完全兼容Redis协议的分布式KV数据库产品。基于百度智能云增强型SSD云盘构建，支持毫秒级在线数据处理、PB级海量数据超低成本存储。
Redis标准版-双副本	系统工作时主节点（Master）和副本（Replica）数据实时同步，主节点故障时系统自动秒级切换，备节点接管业务，全程自动且对业务无影响，主从架构保障系统服务具有高可用性。
Redis集群版-单副本	单副本集群版实例采用集群架构，每个分片服务器采用单副本模式。适用于纯缓存类业务或者QPS压力较大的业务场景。
Redis集群版-双副本	集群（Cluster）实例采用分布式架构，每个数据分片都支持主从（master-replica）高可用，能够自动进行容灾切换和故障迁移。集群版提供多种规格，您可以根据业务压力的大小选择合适的规格，还可以随着业务的发展自由变配。集群版支持两种连接模式：代理模式是集群版的默认连接方式，可提供智能的连接管理，降低应用开发成本。直连模式支持客户端绕过代理服务器直接访问后端数据分片，可降低网络开销和服务响应时间，适用于对Redis响应速度要求极高的业务。
Memcached集群版	Memcached 集群版实例的可自由选择实例规格和分片数量，灵活创建所需的实例容量。

🔗 产品规格

云数据库 Redis标准版和集群版每种架构都有多种规格，详细信息请参见[产品规格](#)。

🔗 产品定价

云数据库 Redis定价请参见[产品定价](#)。更多信息请参见[计费变更](#)、[配置变更](#)。

🔗 应用场景

关于典型的应用场景，请参见[应用场景](#)。

🔗 相关概念

了解以下概念，将帮助您更好地使用Redis：

- **Redis**：Redis 是一个使用 ANSI C 编写的开源、基于内存、可选持久性的键值对存储数据库。
- **Memcached**：Memcached 是开源、基于内存的高速 KV 缓存系统，不支持数据持久化。
- **实例**：每个实例对应一个云数据库 Redis 服务实体，是用户使用云数据库 Redis 服务的基本单位；每个实例由唯一的实例 ID 进行标识，用户可以在控制台中看到自己购买的实例列表。
- **分片**：分片是一组缓存节点的集合，根据所选副本数量的不同，每个分片中包含不同数量的缓存节点。当实例包含多个分片时，数据集被切分到多个分片中，每个分片拥有完整数据集的子集。
- **标准版实例**：标准版架构的 Redis 实例只包含一个分片。兼容 Redis 4.0/5.0/6.0/7.0 版本的协议和命令，是最通用的实例类型。
- **集群版实例**：集群版架构的 Redis 实例包含多个分片，以及多个代理节点。兼容 Redis 4.0/5.0/6.0/7.0 版本的协议和命令，采用分布式架构，支持横向和纵向的扩缩容。集群版实例能够通过横向扩展分片数量，快速扩展数据容量，提升网络/CPU/QPS能力，可承载远高于标准版实例的流量压力。
- **单副本**：单分片中只包含1个节点，该节点充当可读写的主节点。单副本的实例支持读写高并发，但不支持备份或持久化，

不提供数据可靠性，适用于纯缓存等对数据可靠性要求不高的业务场景。

- **双副本**：单分片中包含2个节点，其中一个节点充当可读写的主节点，其余节点充当不可读写且用户不可见的备节点，当主节点故障时，备节点自动升级为主节点。双副本的实例支持秒级故障切换，支持备份和持久化，提供高可靠、高可用特性，适用于各类应用场景。
- **连接地址**：用来连接云数据库 Redis 实例的域名/IP地址。
- **连接密码**：用来连接云数据库 Redis 实例的密码。
- **逐出策略**：与开源版本逐出策略保持一致。

相关产品

- **云服务器BCC**：高性能、高可靠、安全稳定的弹性计算服务。
- **私有网络VPC**：灵活自定义地址空间、提供安全隔离的网络环境。
- **数据传输服务 DTS**：简单可靠的数据库数据传输服务。
- **云数据库 RDS**：专业、高性能、高可靠的云数据库服务。
- **云数据库 GaiaDB-X**：兼容 MySQL 协议的分布式关系型数据库。
- **云数据库 DocDB for MongoDB**：兼容 MongoDB 协议的文档数据库服务。

云数据库 Redis 版初识与选型

Redis 容量型(原 PegaDB)产品介绍

产品介绍

百度智能云K(key)-V(value)数据库是一种字典数据库，以下简称KV数据库。**Redis 容量型**（原PegaDB）是一款完全兼容Redis协议的分布式KV数据库产品。基于百度智能云增强型SSD云盘构建，支持毫秒级在线数据处理、PB级海量数据超低成本存储。具备冷热分离等企业级特性，可以满足业务在超大规模数据场景下对成本及性能的要求，适用于电商、游戏、社交、账单、日志等众多业务场景。

产品特点

1. 全面兼容Redis 7.0协议，支持业务平滑迁移。
2. 基于SSD构建，支持水平扩展，满足大容量、高并发业务需要。
3. 内核层面支持数据自动冷热分离，无需业务侧自己维护。
4. 提供可配置半同步或异步同步的能力，在数据一致性和性能之间做出调整，可以按照业务需求提交 [工单](#) 开通

应用场景

大数据量场景，存储成本高

在业务发展后期，沉淀的数据量越来越多，而访问量越来越低，此时性能存在过剩的情况。使用Redis存储海量KV数据，成本直线上升。同时Redis为内存数据库，在掉电等场景下数据容易丢失。

典型冷热分离场景，传统Cache（缓存）+DB（关系性数据库）架构，业务开发复杂度高

电商、游戏应用通常有海量的商品和玩家数据，同时这些数据有较为明显的冷热属性。如果将海量数据完全放在内存中，存储成本是不可接受的。但是如果完全放在磁盘中，又无法满足热点商品和活跃用户的数据访问，如果业务自己搭建冷数据沉降机制，业务架构变得复杂，维护成本过高。

使用Pika等开源KV数据库，功能&成本&性能无法完全满足需求

客户需要持续投入相应的人力成本，去迭代和维护软件，以应对业务带来的需求。Pika等开源KV数据库无法完美兼容Redis协议，对于业务和开发人员协议不够友好，无法做到无缝的迁移。用户无法在数据一致性和性能之间按照业务需求进行选择，造成运维成本过高。

优势对比

对比项	Redis 容量型	开源 Pika
兼容性	兼容原生Redis命令	和原生Redis命令差别较大
持久化	可配置半同步或异步同步	异步同步
企业级特性	内核支持冷热分离；支持热活实例组	无
性能	约为内存型Redis的80%	约为原生Redis的60%
成本	最低为内存型Redis的15%	自建物理设备成本稍低，运维成本较大

云数据库 Redis 版与自建 Redis 的对比

与自行购买服务器并搭建Redis数据库相比，使用云数据库Redis版在多个方面均展现出显著优势，包括更为强大的数据安全保障、更低的运维成本投入，以及经过深度优化的内核性能。

对比项	云数据库Redis版	自建Redis
安全防护	危险防护： • VPC网络隔离。 • 白名单控制访问。 • 自定义账号与权限。 数据保护：TLS加密。 记录审计：审计日志。	危险防护： • 需自行构建网络安全体系，成本高，难度大。 • 社区版Redis的默认访问配置存在安全漏洞，可能导致Redis数据泄露。 • 无账号鉴权体系。 数据保护：需要自行通过第三方工具实现SSL加密访问。 记录审计：无审计功能。
备份恢复	云数据库 Redis 支持一键恢复到指定实例，或者原地恢复	仅支持一次性全量恢复。
运维管理	• 支持十余组监控指标，最小监控粒度为5秒。 • 支持报警设置。 • 可根据需求创建多种架构的实例，支持变配到其它架构和规格。 • 提供基于快照的大key分析功能，精度高，无性能损耗。	• 需使用管理方式复杂的第三方监控工具实现服务监控。 • 改变规格或架构的操作复杂，且需要停止服务。 • 支持基于采样的大key分析，统计粗糙，精度较低。
部署和扩容	即时开通，弹性扩容。	需要自行完成采购硬件、机房托管、部署机器等工作，周期较长，且需要自行维护节点关系。
高可用	• 单可用区高可用方案。 • 同城容灾方案。 • 高可用性由独立的中心化模块保障，决策效率高且稳定，不会出现脑裂（split brain）现象。	• 需要自行部署基于哨兵模式的机房内高可用架构。 • 可基于哨兵模式搭建同城容灾架构。 • 高可用性由哨兵机制保障，搭建成本高，且在业务高峰期决策效率低，可能发生脑裂导致业务受损。
内存	• 已购内存100%可用，容灾、运维管理、扩容、实例持久化（Fork写时复制）等占用的内存开销均由百度智能云承担，不占用实例内存容量。 • 例如：采购64 GB的云数据库Redis版实例，用户可用内存为64 GB。	• 需预留25% ~ 40%的内存资源用于容灾、运维管理、扩容等用途。 • 例如：采购2台内存为64 GB的BCC搭建Redis主从实例，用户可用内存通常低于45 GB。

Redis 容量版与 Redis 社区版的特性对比

百度智能云Redis内核在功能上较社区版做了大量优化，包括持久化、安全、Module命令、在线性能分析、以及异地多活等企业级特性。以下是主要功能的对比：

类别	功能	社区版	Redis 内存版	Redis 容量版
性能	QPS(以Redis社区版为基准)	100%	100%	String数据类型性能可达80%，复杂数据类型与具体场景有关
	线程模型	多IO线程+单Woker线程	多IO线程+单Woker线程	多IO线程+多Woker线程
数据类型	原生数据结构	100%	100%	70%+
	Module命令支持	不支持	JSON、Search、CAS\CAD、BloomFilter、ExHash	JSON、CAS\CAD
持久化	存储介质	内存	内存	磁盘
	持久化级别	秒级	秒级	命令级别
	AOF异步化	不支持	支持	没有AOF功能，数据实时写磁盘
	RDB Fork优化	不支持	支持	没有RDB功能，数据实时写磁盘
安全	IP白名单	不支持	支持	支持
	SSL加密	支持	支持	支持
	TDE加密	不支持	支持	不支持
	高危命令在线记录和封禁	不支持	支持	不支持
性能分析	访问日志	不支持	支持	支持
	性能分析	支持(slowlog、lantency)	支持(slowlog、lantency)	支持(slowlog、在线命令级Perflog)
	热点key统计	支持	支持	不支持
	大key统计	支持	支持	支持
企业级特性	主从复制一致性	异步复制	支持异步复制和半同步复制	支持异步复制和半同步复制
	主从复制优化	社区基准功能，共享复制Buffer(百度云贡献社区)	新增基于AOF的增量复制、渐进式复制	基于RocksDB WAL的增量复制
	扩缩容优化	慢(基于Key迁移)	快(基于Slot迁移)	快(基于Slot迁移)
	按时间点回档	支持（百度云贡献社区）	支持	支持
	自动读写分离	不支持	支持	不支持
	代理层缓存	不支持	支持	支持
	异地多活	不支持	支持	支持
	DTS数据同步	不支持	支持	支持

产品系列

Redis标准版

标准版架构的 Redis 实例只包含一个分片。兼容 Redis 4.0、5.0、6.0、7.0 版本的协议和命令，是最通用的实例类型。

缓存节点规格

规格	用户可用内存	最大连接数	最大内网带宽	CPU处理能力	**单分片QPS**
cache.n1.nano	0.25GB	1万	10MBS	2核	15万+
cache.n1.micro	1GB	1万	10MBS	2核	15万+
cache.n1.small	2GB	1万	16MBS	2核	15万+
cache.n1.medium	4GB	2万	32MBS	4核	15万+
cache.n1.large	8GB	4万	64MBS	8核	15万+
cache.n1.xlarge	16GB	8万	128MBS	16核	15万+
cache.n1.2xlarge	32GB	16万	256MBS	32核	15万+
cache.n1.4xlarge	64GB	32万	512MBS	64核	15万+
cache.n1.8xlarge	128GB	64万	1024MBS	128核	15万+
cache.n1.16xlarge	256GB	128万	2048MBS	256核	15万+
cache.n1.32xlarge	512GB	256万	4096MBS	512核	15万+
cache.n1.64xlarge	1024GB	512万	8192MBS	1024核	15万+
cache.n1.128xlarge	2048GB	1024万	16384MBS	2048核	15万+
cache.n1.256xlarge	4096GB	2048万	32768MBS	4096核	15万+
cache.n1.512xlarge	8192GB	4096万	65536MBS	8192核	15万+
cache.n1.1024xlarge	16384GB	8192万	131072MBS	16384核	15万+
cache.n1.2048xlarge	32768GB	16384万	262144MBS	32768核	15万+
cache.n1.4096xlarge	65536GB	32768万	524288MBS	65536核	15万+
cache.n1.8192xlarge	131072GB	65536万	1048576MBS	131072核	15万+
cache.n1.16384xlarge	262144GB	131072万	2097152MBS	262144核	15万+
cache.n1.32768xlarge	524288GB	262144万	4194304MBS	524288核	15万+
cache.n1.65536xlarge	1048576GB	524288万	8388608MBS	1048576核	15万+
cache.n1.131072xlarge	2097152GB	1048576万	16777216MBS	2097152核	15万+
cache.n1.262144xlarge	4194304GB	2097152万	33554432MBS	4194304核	15万+
cache.n1.524288xlarge	8388608GB	4194304万	67108864MBS	8388608核	15万+
cache.n1.1048576xlarge	16777216GB	8388608万	134217728MBS	16777216核	15万+
cache.n1.2097152xlarge	33554432GB	16777216万	268435456MBS	33554432核	15万+
cache.n1.4194304xlarge	67108864GB	33554432万	536870912MBS	67108864核	15万+
cache.n1.8388608xlarge	134217728GB	67108864万	1073741824MBS	134217728核	15万+
cache.n1.16777216xlarge	268435456GB	134217728万	2147483648MBS	268435456核	15万+
cache.n1.33554432xlarge	536870912GB	268435456万	4294967296MBS	536870912核	15万+
cache.n1.67108864xlarge	1073741824GB	536870912万	8589934592MBS	1073741824核	15万+
cache.n1.134217728xlarge	2147483648GB	1073741824万	17179869184MBS	2147483648核	15万+
cache.n1.268435456xlarge	4294967296GB	2147483648万	34359738368MBS	4294967296核	15万+
cache.n1.536870912xlarge	8589934592GB	4294967296万	68719476736MBS	8589934592核	15万+
cache.n1.1073741824xlarge	17179869184GB	8589934592万	137438953472MBS	17179869184核	15万+
cache.n1.2147483648xlarge	34359738368GB	17179869184万	274877906944MBS	34359738368核	15万+
cache.n1.4294967296xlarge	68719476736GB	34359738368万	549755813888MBS	68719476736核	15万+
cache.n1.8589934592xlarge	137438953472GB	68719476736万	1099511627776MBS	137438953472核	15万+
cache.n1.17179869184xlarge	274877906944GB	137438953472万	2199023255552MBS	274877906944核	15万+
cache.n1.34359738368xlarge	549755813888GB	274877906944万	4398046511104MBS	549755813888核	15万+
cache.n1.68719476736xlarge	1099511627776GB	549755813888万	8796093022208MBS	1099511627776核	15万+
cache.n1.137438953472xlarge	2199023255552GB	1099511627776万	17592186044416MBS	2199023255552核	15万+
cache.n1.274877906944xlarge	4398046511104GB	2199023255552万	35184372088832MBS	4398046511104核	15万+
cache.n1.549755813888xlarge	8796093022208GB	4398046511104万	70368744177664MBS	8796093022208核	15万+
cache.n1.1099511627776xlarge	17592186044416GB	8796093022208万	140737488355328MBS	17592186044416核	15万+
cache.n1.2199023255552xlarge	35184372088832GB	17592186044416万	281474960710656MBS	35184372088832核	15万+
cache.n1.4398046511104xlarge	70368744177664GB	35184372088832万	562949921421312MBS	70368744177664核	15万+
cache.n1.8796093022208xlarge	140737488355328GB	70368744177664万	1125899842842624MBS	140737488355328核	15万+
cache.n1.17592186044416xlarge	281474960710656GB	140737488355328万	2251799685685248MBS	281474960710656核	15万+
cache.n1.35184372088832xlarge	562949921421312GB	281474960710656万	4503599371370496MBS	562949921421312核	15万+
cache.n1.70368744177664xlarge	1125899842842624GB	562949921421312万	9007198742740992MBS	1125899842842624核	15万+
cache.n1.140737488355328xlarge	2251799685685248GB	1125899842842624万	18014397485481984MBS	2251799685685248核	15万+
cache.n1.281474960710656xlarge	4503599371370496GB	2251799842842624万	36028794970963968MBS	4503599371370496核	15万+
cache.n1.562949921421312xlarge	9007198742740992GB	4503599371370496万	72057589841927936MBS	9007198742740992核	15万+
cache.n1.1125899842842624xlarge	18014397485481984GB	9007198742740992万	144115179683855872MBS	18014397485481984核	15万+
cache.n1.2251799685685248xlarge	36028794970963968GB	18014397485481984万	288230359367711744MBS	36028794970963968核	15万+
cache.n1.4503599371370496xlarge	72057589841927936GB	36028794970963968万	576460718735423488MBS	72057589841927936核	15万+
cache.n1.9007198742740992xlarge	144115179683855872GB	72057589841927936万	1152921437470846976MBS	144115179683855872核	15万+
cache.n1.18014397485481984xlarge	288230359367711744GB	144115179683855872万	2305842874941693952MBS	288230359367711744核	15万+
cache.n1.36028794970963968xlarge	576460718735423488GB	288230359367711744万	4611685749883387904MBS	576460718735423488核	15万+
cache.n1.72057589841927936xlarge	1152921437470846976GB	576460718735423488万	9223371499766775808MBS	1152921437470846976核	15万+
cache.n1.144115179683855872xlarge	2305842874941693952GB	1152921437470846976万	18446742999533551616MBS	2305842874941693952核	15万+
cache.n1.288230359367711744xlarge	4611685749883387904GB	2305842874941693952万	36893485999067103232MBS	4611685749883387904核	15万+
cache.n1.576460718735423488xlarge	9223371499766775808GB	4611685749883387904万	73786971998134206464MBS	9223371499766775808核	15万+
cache.n1.1152921437470846976xlarge	18446742999533551616GB	9223371499883387904万	147573943996268412928MBS	18446742999533551616核	15万+
cache.n1.2305842874941693952xlarge	36893485999067103232GB	18446742999533551616万	295147887992536825856MBS	36893485999067103232核	15万+
cache.n1.4611685749883387904xlarge	73786971998134206464GB	36893485999067103232万	590295775985073651712MBS	73786971998134206464核	15万+
cache.n1.9223371499766775808xlarge	147573943996268412928GB	73786971998134206464万	1180581551970147303424MBS	147573943996268412928核	15万+
cache.n1.18446742999533551616xlarge	295147887992536825856GB	147573943996268412928万	2361163103940294606848MBS	295147887992536825856核	15万+
cache.n1.370295775985073651712xlarge	590295775985073651712GB	295147887992536825856万	472231551970147303424MBS	590295775985073651712核	15万+
cache.n1.740581551970147303424xlarge	1180581551970147303424GB	590295775985073651712万	944463103940294606848MBS	1180581551970147303424核	15万+
cache.n1.147573943996268412928xlarge	2361163103940294606848GB	1180581551970147303424万	1888926207880589213696MBS	2361163103940294606848核	15万+
cache.n1.295147887992536825856xlarge	472231551970147303424GB	2361163103940294606848万	3777852415761178027392MBS	472231551970147303424核	15万+
cache.n1.590295775985073651712xlarge	944463103940294606848GB	472231551970147303424万	7549704831522356054784MBS	944463103940294606848核	15万+
cache.n1.1180581551970147303424xlarge	1888926207880589213696GB	944463103940294606848万	15099409663044712109568MBS	1888926207880589213696核	15万+
cache.n1.2361163103940294606848xlarge	3777852415761178027392GB	1888926207880589213696万	3019804831522356054784MBS	3777852415761178027392核	15万+
cache.n1.472231551970147303424xlarge	7549704831522356054784GB	3777852415761178027392万	6039609663044712109568MBS	7549704831522356054784核	15万+
cache.n1.944463103940294606848xlarge	15099409663044712109568GB	7549704831522356054784万	12079219326089424219136MBS	15099409663044712109568核	15万+
cache.n1.1888926207880589213696xlarge	3019804831522356054784GB	15099409663044712109568万	24158438652178848438272MBS	3019804831522356054784核	15万+
cache.n1.3777852415761178027392xlarge	6039609663044712109568GB	3019804831522356054784万	48316877304357442176544MBS	6039609663044712109568核	15万+
cache.n1.7549704831522356054784xlarge	12079219326089424219136GB	6039609663044712109568万	96632394612714884353088MBS	12079219326089424219136核	15万+
cache.n1.15099409663044712109568xlarge	24158438652178848438272GB	12079219326089424219136万	193164789225429768706176MBS	24158438652178848438272核	15万+
cache.n1.3019804831522356054784xlarge	48316877304357442176544GB	24158438652178848438272万	38633757608685884353088MBS	48316877304357442176544核	15万+
cache.n1.6039609663044712109568xlarge	96632394612714884353088GB	48316877304357442176544万	77264790225429768706176MBS	96632394612714884353088核	15万+
cache.n1.12079219326089424219136xlarge	193164789225429768706176GB	96632394612714884353088万	154529578450859537412352MBS	193164789225429768706176核	15万+
cache.n1.24158438652178848438272xlarge	38633757608685884353088GB	193164789225429768706176万	31307115721771768706176MBS	38633757608685884353088核	15万+
cache.n1.48316877304357442176544xlarge	77264790225429768706176GB	38633757608685884353088万	62614380443543537412352MBS	77264790225429768706176核	15万+
cache.n1.96632394612714884353088xlarge	154529578450859537412352GB	77264790225429768706176万	125228756907089074824704MBS	154529578450859537412352核	15万+
cache.n1.193164789225429768706176xlarge	31307115721771768706176GB	154529578450859537412352万	25045751443543537412352MBS	31307115721771768706176核	15万+
cache.n1.38633757608685884353088xlarge	62614380443543537412352GB	31307115721771768706176万	50091502887087074824704MBS	62614380443543537412352核	15万+
cache.n1.77264790225429768706176xlarge	125228756907089074824704GB	62614380443543537412352万	100183505774178149649408MBS	125228756907089074824704核	15万+
cache.n1.15045751443543537412352xlarge	25045751443543537412352GB	125228756907089074824704万	200367029470376299298816MBS	25045751443543537412352核	15万+
cache.n1.30091502887087074824704xlarge	50091502887087074824704GB	25045751443543537412352万	40073405774178149649408MBS	50091502887087074824704核	15万+
cache.n1.60183505774178149649408xlarge	100183505774178149649408GB	50091502887087074824704万	80366811548356299298816MBS	100183505774178149649408核	15万+
cache.n1.120367029470376299298816xlarge	200367029470376299298816GB	100183505774178149649408万	16073405774178149649408MBS	200367029470376299298816核	15万+
cache.n1.24073405774178149649408xlarge	40073405774178149649408GB	200367029470376299298816万	32146811548356299298816MBS	40073405774178149649408核	15万+
cache.n1.48146811548356299298816xlarge	80366811548356299298816GB	40073405774178149649408万	64293623096712598597732MBS	80366811548356299298816核	15万+
cache.n1.96373623096712598597732xlarge	16073405774178149649408GB	80366811548356299298816万	128587246193425197195464MBS	16073405774178149649408核	15万+
cache.n1.192748474178149649408xlarge	32146811548356299298816GB	16073405774178149649408万	257174492386850394390928MBS	32146811548356299298816核	15万+
cache.n1.38549623096712598597732xlarge	64293623096712598597732GB	32146811548356299298816万	51435246193425197195464MBS	64293623096712598597732核	15万+
cache.n1.76879246193425197195464xlarge	128587246193425197195464GB	64293623096712598597732万	102870492386850394390928MBS	128587246193425197195464核	15万+
cache.n1.153758492386850394390928xlarge	257174492386850394390928GB	128587246193425197195464万	205740984773700788781856MBS	257174492386850394390928核	15万+
cache.n1.307516984773700788781856xlarge	51435246193425197195464GB	257174492386850394390928万	411481963867401577563712MBS	51435246193425197195464核	15万+
cache.n1.61503396954840394390928xlarge	102870492386850394390928GB	51435246193425197195464万	82296392773680394390928MBS	102870492386850394390928核	15万+
cache.n1.12300679390960788781856xlarge	2057				

	cache.n1.medium		4GB		6万		24MBS		2核		15万+	
	cache.n1.large		8GB		6万		24MBS		2核		15万+	
	cache.n1.xlarge		16GB		6万		32MBS		2核		15万+	
	cache.n1.3large		24GB		6万		40MBS		2核		15万+	
	cache.n1.2xlarge		32GB		6万		48MBS		2核		15万+	
	cache.n1.4xlarge		64GB		6万		64MBS		2核		15万+	
	cache.n1.c4.medium		4GB		6万		24MBS		4核		15万+	
	cache.n1.c4.large		8GB		6万		24MBS		4核		15万+	
	cache.n1.c4.xlarge		16GB		6万		32MBS		4核		15万+	
	cache.n1.c4.2xlarge		32GB		6万		48MBS		4核		15万+	
	cache.n1.c4.4xlarge		64GB		6万		64MBS		4核		15万+	

可选副本数量

	副本数量		数据副本		单分片内容		说明	
	---		---		---		---	
	1		数据单副本		包含1个缓存节点：主节点 1个		不提供数据可靠性，不支持备份，不支持持久化	
	2		数据双副本		包含2个缓存节点：主节点 1个 + 备节点 1个		提供数据可靠性，支持备份，支持持久化	

Redis集群版

集群版架构的 Redis 实例包含多个分片，以及多个代理节点。兼容 Redis 4.0、5.0、6.0、7.0 版本的协议和命令，采用分布式架构，支持横向和纵向（即将上线）的扩缩容。集群版实例能够通过横向扩展分片数量，快速扩展数据容量，提升网络/CPU/QPS能力，可承载远高于标准版实例的流量压力。

Redis 集群版实例的可自由选择单分片的缓存节点规格和分片数量，灵活创建所需的实例规格。

缓存节点规格

	规格		用户可用内存		最大连接数		最大内网带宽		CPU处理能力		单分片QPS		可选分片数量	
	---		---		---		---		---		---		---	
	cache.n1.micro		1GB		1万		10MB/S		2核		15万+		1~256自定义	
	cache.n1.small		2GB		1万		16MB/S		2核		15万+		1~256自定义	
	cache.n1.medium		4GB		6万		24MB/S		2核		15万+		1~256自定义	
	cache.n1.c4.medium		4GB		6万		24MB/S		4核		15万+		1~256自定义	
	cache.n1.large		8GB		6万		24MB/S		2核		15万+		1~256自定义	
	cache.n1.c4.large		8GB		6万		24MB/S		4核		15万+		1~256自定义	
	cache.n1.xlarge		16GB		6万		32MB/S		2核		15万+		1~256自定义	
	cache.n1.4c.xlarge		16GB		6万		32MB/S		4核		15万+		1~256自定义	
	cache.n1.3large		24GB		6万		40MB/S		4核		15万+		1~256自定义	
	cache.n1.2xlarge		32GB		6万		48MB/S		2核		15万+		1~256自定义	

集群规格

	项目		说明	
	---		---	
	总用户可用内存		缓存节点用户可用内存 * 分片数量	
	总最大连接数		缓存节点最大连接数 * 分片数量	
	总最大内网带宽		缓存节点最大内网带宽 * 分片数量	
	总CPU处理能力		缓存节点CPU处理能力 * 分片数量	
	总QPS参考		缓存节点QPS参考 * 分片数量	

可选副本数量

	副本数量		数据副本		单分片内容		说明	
	---		---		---		---	
	1		数据单副本		包含1个缓存节点：主节点 1个		不提供数据可靠性，不支持备份，不支持持久化	
	2		数据双副本		包含2个缓存节点：主节点 1个 + 备节点 1个		提供数据可靠性，支持备份，支持持久化	

边缘数据库Redis

概述

边缘云通过将Redis数据库下沉到全国各地的CDN机房、运营商IDC机房，通过地理位置和终端业务拉近距离，有效降低延

产品架构

数据链路

Redis 主从架构

Redis 主从架构提供主节点和从节点的双副本模式，主从数据实时同步，正常情况下主节点对外提供服务，从节点正常运行并备份不影响节点运行；具备高可用性，主节点故障时系统自动切换，从节点提升为主节点提供服务，对业务的访问不产生影响。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/SCS/image_15205e2.png)

Redis 集群架构

Redis 集群（Cluster）实例采用分布式架构，每个 Redis 节点都采用一主一从的高可用架构，能够进行自动容灾切换和故障迁移。多种集群规格可适配不同的业务压力，可按需扩展数据库性能。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/SCS/image_4dbde0e.png)

Memcached 集群架构

Memcached 集群（Cluster）实例采用分布式架构，代理节点按照 Key 进行 Hash 将数据散列到后端不同的 Memcached 节点上。购买时实例规格可配置，分片数量可配置。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/SCS/image_1988e6d.png)

高可用架构

对每一个实例，默认 Redis 主节点提供对外服务，有一个热备节点实时和主节点复制数据，实例上部署 Agent 程序，将 Redis 的状态采集上报给 Cache Master，Cache Master 按照上报的状态进行判断，能很快获取 Redis 主节点的存活状态，发现 Redis 主节点异常，切换到备节点上，更改 BLB 或者 Proxy 的路由，实现秒级切换，对业务无损。

![redis.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/SCS/redis_628819f.png)

KV 数据库 Redis 容量型(原 PegaDB)

Redis 容量型数据库采用分布式架构，每一个 Redis 容量型节点都采用一主一从的高可用架构。代理层支持自动的读写流量分发。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_7b5b8a8.png)

产品特性

架构灵活

单机架构

单机架构只包含一个缓存节点，支持单节点弹性变配，满足高QPS需求，适用于纯缓存场景，提供超高性价比。

主从架构

主从架构采用主从双节点部署架构，包含一主一从两个缓存节点，支持自动故障切换、支持备份和持久化，适用于对高可用、高可靠都有要求的场景。

集群架构

集群（Cluster）实例采用分布式架构，支持自定义分片数、副本数。双副本的集群每个分片都采用一主一从的高可用架构，能够进行自动容灾切换和故障迁移。多种集群规格可适配不同的业务压力，可按需扩展数据库性能。

数据安全

数据持久化存储

采用内存加硬盘的混合存储方式，在提供高速数据读写能力的同时满足数据持久化需求。

备份及一键恢复

每天自动备份数据，数据容灾能力强，免费支持数据一键恢复，有效防范数据误操作，将可能发生的业务损失降到最低。

多层网络安全防护

- * 私有网络 VPC 在 TCP 层直接进行网络隔离保护。
- * DDOS 防护实时监测并清除大流量攻击。
- * 支持 1000 个以上 IP 白名单配置，直接从访问源进行风险控制。

深度内核优化

百度专家团队对 Redis 源码进行深度内核优化，有效防止内存溢出，修复安全漏洞，为您保驾护航。

高可用性

主从双节点

主从版与集群版的双副本实例均有主从双节点，避免单点故障引起的服务中断。

自动检测与恢复

自动侦测硬件故障，发生故障时能够进行故障转移，在数秒内恢复服务。

资源隔离

实例级别的资源隔离可以更好地保障单个用户服务的稳定性。

弹性扩展

数据容量扩展

云数据库 Redis 支持多种内存规格的产品配置，可根据业务量大小在线升级内存规格，无需中断服务，不影响业务。

业务形态扩展

支持单节点缓存架构和双节点存储架构，适配不同业务场景。标准版和双节点版之间能够灵活变配。

性能扩展

支持集群架构下弹性扩展数据库系统的存储空间及吞吐性能，突破海量数据高 QPS 性能瓶颈，轻松应对每秒百万次的读写需求。

智能运维

监控平台

提供 CPU 利用率、连接数、磁盘空间利用率等实例信息实时监控及报警，随时随地了解实例动态。

数据库内核版本管理

主动升级，快速修复缺陷，免去日常版本管理苦恼；优化 Redis 参数配置，最大化利用系统资源。

功能概述

功能概览

由于不同引擎、产品系列、版本的 Redis 实例支持的功能不同，建议您查看各引擎实例支持的功能概况。

- Redis 功能概览

应用场景

数据使用场景

缓存

搭配持久化存储使用，缓解后端存储服务的压力。随着用户对应用延迟的容忍度越来越低，基于磁盘的随机数据访问性能却没有革命性的提升，数据访问性能成为产品性能短板。使用分布式缓存服务可以将有效访问数据放在缓存，缓存与后端的磁盘存储系统通过同步或异步的机制保持数据一致，大部分的数据访问请求可以直接命中缓存，提供比磁盘访问降低好几个数量级的响应时间，当数据访问请求缓存不命中时，需要访问磁盘存储系统并更新缓存。

键值存储

在需要快速响应的业务中可以将分布式缓存单独使用。随着缓存服务可靠性越来越高，使用越来越方便，当某些产品的数据不需要很强的事务支持、强一致的时候，通过弱化对事务、数据一致性的支持，用户的所有数据全部存放在缓存中，键值存储能够提供丰富的数据存储模式，在互联网场景下对业务更为友好，而且获取到了更高的扩展能力、性能和可靠性。

行业应用场景

电商行业应用

1. 商品信息缓存：

Redis能够高速地缓存数据，对于电商网站中经常访问的商品信息，可以将其缓存在Redis中，以减少对后端数据库的访问压力，提高网站的响应速度。由于Redis的响应时间非常快，通常只需几毫秒便能完成操作，因此可以显著提升用户体验。

3. 购物车管理：购物车是电商网站的核心功能之一。使用Redis可以很方便地进行购物车缓存管理，包括添加商品、删除商品等操作。Redis的高性能可以确保购物车功能的流畅使用，同时其集群化功能还能实现购物车数据的高可用性和高性能。

5. 分布式锁：在电商系统中，经常需要对某个资源进行并发访问控制，例如限制同时只能有一个用户下单。Redis的分布式锁功能提供了一种可靠的方式来控制并发访问，有效避免了因并发引起的数据竞争等问题。

7. 消息队列：Redis的list数据结构可以作为一个任务队列来使用。在高并发的电商系统中，特别是在秒杀或抢购场景下，需要实时处理大量请求。此时，Redis可以将这些请求放在list里面，后台程序监听这个list并取出请求进行处理，从而有效应对高并发场景。

9. 实时更新与订阅发布：电商应用中经常需要实时更新数据，例如价格变化或库存变化。Redis的订阅和发布功能可以使得这类应用更有效地更新数据，并实时推送变化给客户端，保持数据的实时性和准确性。

1. 订单信息存储与处理：订单信息是电商企业的核心业务之一。使用Redis可以存储订单信息、配送信息以及订单状态等数据，并且提供基于时间的缓存策略，以确保订单信息的及时性和准确性，同时避免数据过期或混乱。

游戏行业应用

1. 游戏数据缓存：为了提高游戏的读写效率和响应速度，游戏开发者通常会将一些频繁访问的数据，如玩家信息、游戏状态等，缓存在Redis中。由于Redis是基于内存的数据库，其读写速度远快于传统的磁盘数据库，因此能够显著提升游戏的性能。

3. 实时排行榜实现：在游戏中，实时排行榜是一个重要的功能，它能够激发玩家的竞争意识。Redis的有序集合数据结构非常适合用来实现实时排行榜。游戏服务器可以实时更新玩家的分数，并同步到Redis中，从而确保排行榜的实时性和准确性。

5. 聊天服务器支持：游戏中的实时聊天功能对于增强玩家之间的互动至关重要。Redis的发布/订阅模式非常适合用于实现实时聊天功能。聊天服务器可以将玩家发送的消息发布到Redis的指定频道，其他订阅了该频道的玩家就能够实时接收到这些消息。

9. 玩家状态管理：Redis可以用来存储和管理玩家的在线状态、游戏进度等信息。这些信息可以实时更新，并供其他游戏逻辑使用，以实现更丰富的游戏功能和更好的用户体验。

1. 游戏配置和规则存储：游戏的配置信息和规则也可以存储在Redis中，这样便于动态地修改和调整游戏设置，而无需重启游戏服务器。

短视频直播应用

1. 实时消息传递：利用Redis的发布/订阅模型，可以实现直播间的实时消息传递功能。例如，当主播发布新的视频或进行直播时，可以通过Redis实时推送消息给订阅了该主播的用户，确保用户能够及时收到更新。

3. 在线用户管理：Redis的SET数据结构可以用来动态管理在线用户。通过实时添加或删除在线用户的信息，可以准确掌握当前在线用户的状态，为直播间提供实时的观众互动数据。

5. 虚拟礼物与活动通知：在短视频直播中，观众可以赠送虚拟礼物给主播。Redis的LIST数据结构可以用来缓存未发送的礼物信息，确保在网络不稳定时礼物能够延迟发送。同时，Redis还可以用于发布粉丝团活动通知，确保粉丝团成员能够实

时收到活动信息。

8. 推荐系统支持：利用Redis的ZSET数据结构，可以记录观众的喜好并进行实时推荐。例如，根据观众在直播间的互动行为、观看历史等信息，为其推荐相似的直播间或视频内容。

10. 数据统计与分析：Redis还可以用于存储直播间的实时统计数据，如观看人数、点赞数、评论数等。这些数据可以为后续的数据分析提供有力支持，帮助平台更好地了解用户行为和需求。

金融领域应用

1. 高速数据处理与缓存

关键数据缓存：金融机构通常将股票价格、交易订单等关键数据存储在Redis中。由于Redis是基于内存的数据库，因此能够实现亚毫秒级的数据处理速度，显著提高交易效率。

2. 用户信息缓存：金融系统中用户的个人信息，如用户名、密码等，也可存储在Redis中以便快速访问。

4. 实时监控与分析

Redis提供了实时监控功能，结合其Pub/Sub模式，金融机构能够实时接收市场数据更新和业务事件通知。这对于需要实时监控市场动态和业务运行状态的金融行业来说至关重要。

5. 消息队列与异步处理

在金融领域，处理大量的异步消息，如交易记录、账户变化等，是日常业务的一部分。Redis可以作为消息队列，有效地处理这些异步消息，确保系统的顺畅运行。

AI领域应用

1. 模型缓存与在线部署：在机器学习任务中，模型训练往往耗费大量时间。Redis的内存数据库特性使其能够实现在线部署和模型更新，保持模型和应用程序之间的高速和连续性。这意味着当模型需要更新时，可以迅速地将新模型推送到生产环境，而无需长时间的停机或复杂的部署流程。

3. 消息队列与异步任务处理：在AI系统中，经常需要处理大量的请求和任务，如分布式训练、数据处理等。Redis的队列功能使得这些任务可以被有效地管理和分配。通过将大量的训练任务或其他计算任务存储在Redis队列中，并使用异步方式完成它们，可以显著提高系统的可扩展性和稳定性。

5. 实时数据聚合与分析：对于AI场景中的实时数据聚合需求，Redis提供了快速的计算和数据汇总功能。这在需要对大量数据进行迭代和处理的场景中尤为有用，如实时分析用户行为、监控系统性能等。通过使用Redis的聚合功能，可以快速获得分析结果，并将其应用于任何需要的应用程序。

7. 数据存储与参数管理：Redis支持多种数据类型和数据结构，使其非常适合存储和处理AI应用中的各种数据，如机器学习模型的参数、文本数据、结构化数据等。此外，Redis的高性能读写速度使得这些数据可以被迅速访问和更新，从而提高AI应用的运行效率。

使用须知

云数据库 Redis 优势

使用云数据库 Redis 以后，您不需要关心缓存的基础运维工作，如高可用、数据可靠性、水平扩展、备份、监控等工作。您只需要关注您的业务使用情况，观察云数据库 Redis 实例的性能是否足够，如果不够可以考虑实例升级、或和我们一起来分析业务数据来优化业务。

云数据库 Redis 连接须知

云数据库 Redis 实例对外提供域名和内网IP两种访问方式，两种连接方式都能访问云数据库 Redis 实例；但是在高并发情况下，建议您使用内网 IP 来访问，这种方式可以支撑更大的访问量。同时您不用担心后端节点发生故障 IP 地址会改变，对外的内网IP是固定分配给这个实例做负载均衡的 IP，在任何情况下都不会改变。

一般情况下我们建议您对 Redis 连接和读写操作设置 3 次的超时重试，能有效为您过滤掉网络抖动带来的影响，为您业务带来更高的可用性结果。

****云数据库 Redis 数据备份风险须知****

一般情况下 Redis 主备节点已经提供了高可用的落盘数据备份。目前云数据库 Redis 的数据备份功能提供的天级别的数据快照的备份是应对极端故障或者业务误删除来使用的。一般需要您设置一个业务空闲的时间段来进行快照备份，以免 Redis 产生快照时对业务的正常读写带来影响。在发生极端情况下可以用历史数据进行全量或者部分的恢复。在进行恢复之前请您再三确认并手动生成一份最新的备份，以免数据恢复操作造成数据丢失。

****云数据库 Redis 实例磁盘容量风险须知****

您购买的云数据库 Redis 实例只是按照内存使用来进行收费，Redis 还需要磁盘空间来存储持久化数据，由于 Redis 本身的特性，我们为您的实例预留了足够的空间。但是在极端情况下，持久化数据可能会临时占满预留的磁盘，对实例的运行带来风险。这主要是与您业务的写入量相关，因此请您按照自己业务写入的 QPS 选择相应规格的产品，一般而言集群版能够比标准版提供更大的写入 QPS。

****云数据库 Redis 故障切换须知****

云数据库 Redis 实例采用主备节点的高可用模式，当主节点出现不可使用的情况时，自动恢复模块会在 1 分钟之内将您的数据库入口切换到热备节点，请您提前知晓并配置好业务与服务器的重连，避免入口变更影响业务不可用。

多活实例组介绍

简介

若业务有跨地域数据同步，同时对写入性能要求较高，您可以使用云数据库 Redis 版自研的多活实例组产品。多活实例组由至多5个云数据库 Redis实例组成，每个实例均可以进行本地域读写，每个地域数据由同步组件自动同步至其他地域。多活实例组应用场景如下：

- * 异地多活：业务需求应用就近访问同时数据整体一致。
- * 数据灾备：可实现同城灾备、两地三中心灾备及三地灾备等多种数据灾备场景。

架构

****多活实例组****是所有多活实例组成员实例，及同步通道的逻辑集合，所有实例通过同步通道保持实时数据同步，各组件介绍如下：

*** **实例****

* 构成多活实例组的实例，是构成多活实例组的基本服务单元。所有子实例均可本地读写，并且相互之间保持实时双向数据同步。您可以在不同的地域创建子实例，实现跨地域容灾和多活的需求。

*** **同步通道****

* 负责实例之间实时数据同步的组件及网络链路，多活实例组在原生Redis AOF日志的基础上增加了opid, 时间戳等信息，同步通道通过传输aof同步数据。

*** **多活实例组中控****

* 管理多活实例组生命周期，负责部署，配置同步组件，监控相关组件工作等。

**产品优势**

| 优势 | 说明 |

| --- | --- |

| 高可靠 | 支持断点续传，可容忍一定时间的网络中断，避免了原生Redis同步协议在跨地域时的局限性。实例发生主从故障切换时，自动从新主重新建立同步通道，进行断点续传。 |

| 高性能 | 高吞吐：同步通道单向可达**5**万QPS。 | 低延迟：本地域读写与单实例的延迟基本一致，跨地域间同步，延迟几十毫秒至几百，由链路吞吐和链路的往返时延RTT（Round-trip time）决定。 |

| 高正确性 | Binlog按产生的顺序同步到对端。 | 支持回环控制，避免Binlog循环同步。 |

灾备方案介绍

云数据库Redis容灾架构

为保障数据安全性，云数据库Redis提供了多种灾备方案供您选择。

| 方案名称 | 方案内容 |

| --- | --- |

| 单可用区高可用方案 | 主备节点部署在同一可用区中的不同计算节点上，当任一节点发生故障时，由中控系统自动执行故障切换，避免单点故障引起的服务中断。 |

| 同城高可用方案 | 备节点部署在不同可用区的计算节点上，当任一可用区发生故障时，由中控系统自动执行故障切换，保证实例的可用性。 |

| 跨地域容灾方案 | 由多个实例组成热活实例组或异地多活实例组，适用于异地灾备、异地多活、应用就近访问、分摊负载等场景。 |

单可用区高可用方案

除单副本实例外均支持单机房高可用架构。由中控监控主备节点的健康状态并自动执行故障切换，避免单点故障引起的服务中断。

| 部署架构 | 说明 |

| --- | --- |

| 标准版 |  (http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_d4826f6.png) 标准架构（双副本）实例采用主从架构，中控模块检测到主节点故障时，会自动进行主从切换，将从提升为主，流量也修改为指向新主，从而恢复实例的访问。 |

| 企业集群版 |  (http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_9e0a513.png) 企业集群版架构（双副本）数据被分成多个。 |

同城高可用方案

可在创建云数据库Redis实例时，选择支持同城容灾的可用区（即多可用区）。

数据通过可用区之间网络专线进行同步，由中控监控多个可用区主备节点的健康状态并自动执行故障切换。

跨地域容灾方案

为了实现跨地域的容灾，百度云Redis提供了两种跨地域的产品。

| 跨地域产品 | 说明 |

| --- | --- |

| 热活实例组 | 热活实例组由1个云数据库 Redis实例作为主角色，至多4个云数据库 Redis实例作为从角色组成；所有实例均可写，其中主角色为本地写，其他角色的写转发至主角色；所有实例均可本地读。 |

| 多活实例组 | 多活实例组由至多5个云数据库 Redis实例组成，每个实例均可以进行本地域读写，每个地域数据由同步组件自动同步至其他地域。 |

可观测性能力介绍

云数据库Redis支持的可观测能力如下

| 可观测性能力 | 标准版 | 企业集群版 |

| --- | --- | --- |

| 指标 | [性能指标](https://cloud.baidu.com/doc/SCS/s/akcmrhczv) | ✓ | ✓ |

| 日志 | [运行日志](https://cloud.baidu.com/doc/SCS/s/ukb1q89v3) | ✓ | ✓ |

| | [慢日志](https://cloud.baidu.com/doc/SCS/s/4kb1pfgwr) | ✓ | ✓ |

| | 审计日志 | ✓ | ✓ |

| 分析能力 | [热Key分析](https://cloud.baidu.com/doc/SCS/s/Lllykaixn) | ✓ | ✓ |

| | [大Key分析](https://cloud.baidu.com/doc/SCS/s/Akb1qmozd) | ✓ | ✓ |

日志

云数据库Redis版提供了查询运行日志、慢日志、审计日志、时延洞察等功能。

*****运行日志（Redis log）****：运行日志记录运行过程中的日志信息，包括持久化、同步复制、报错信息以及代码中定义的调试记录等，在控制台目标实例详情页的日志管理 > 运行日志下载页签中，可下载指定时间段Redis的运行日志。

*****慢日志（Slowlog）****：记录Redis中执行时间（不含命令排队与网络传输时间）超过指定阈值的请求，慢日志信息包含执行时间戳、执行时长、命令参数、客户端信息等。您可以通过该功能第一时间查询耗时过长的命令列表，并进行相应优化，避免线上服务发生阻塞。在控制台目标实例详情页的日志管理 > 慢日志下载页签中，可下载指定时间段慢日志。

情况，避免线上服务及上阻塞。在控制台中对实例详情页的日志管理 > 审计日志页面中开启审计日志。（注意：开启审计日志会对性能造成负面影响，请在业务低峰期谨慎开启）

*****审计日志****：记录Redis中所有写请求，包含敏感操作FLUSHALL、FLUSHDB、DEL等，在控制台目标实例详情页的日志管理 > 审计日志页面中开启审计日志。（注意：开启审计日志会对性能造成负面影响，请在业务低峰期谨慎开启）

分析能力

*****热Key分析****：对某个或某几个Key的异常高频访问称为热Key，若未能及时处理热Key可能会导致访问倾斜甚至缓存击穿等问题；控制台目标实例详情页的分析缓存> 热Key页签下开启实时热Key分析。

*****大Key分析****：当某个Key含有较多数据成员或者占用较大内存时，可以将其称为大Key；控制台目标实例详情页的分析缓存> 大Key页签下启动离线大Key分析任务，分析任务完成后可在该页面查看占用空间Top100的Key。

热活实例组介绍

简介

若业务有跨地域数据同步需求，读多写少的场景，您可以使用云数据库 Redis 版自研的热活实例组产品。热活实例组由1个云数据库 Redis实例作为主角色，至多4个云数据库 Redis实例作为从角色组成；所有实例均可本地读；热活实例组应用场景如下：

* 跨地域业务读多写少：业务有跨地域数据同步的需求且读多写少。

* 数据灾备：可实现同城灾备、两地三中心灾备及三地灾备等多种数据灾备场景。

架构

****热活实例组****是所有热活实例组成员实例的逻辑集合：

*****主实例****

* 热活实例组的基准实例，除了处理本实例上读写请求外，所有从实例的写请求都会转发到主实例上，所有从实例从主实例同步数据。

*****从实例****

* 本实例上读请求本地进行响应，写请求转发到主实例上，从主实例同步数据。

*****热活实例组中控****

* 管理热活实例组生命周期，负责部署，配置流量转发，监控相关组件工作等。

产品优势

| 优势 | 说明 |

| --- | --- |

| 高可靠 | 自动完成主实例故障时的切换，以及相关从实例同步的重建。 |

| 高性能 | 高吞吐：每个实例可承担10万以上的读请求，可通过增加从实例的方式线性扩展读的能力。 |

| 高一致性 | 由于写请求由一个实例进行处理，相对多活实例组有更高的一致性。 |

行业案例

广告行业应用

云数据库Redis容量型（PegaDB）赋能广告投放系统，实现业务场景落地。

一、前言

当今互联网广告已经深入到我们的日常生活中，成为不可或缺的一部分。“百度一下”、“微信搜一搜”、“抖音刷一刷”即可获取到信息。这些平台不仅是大家日常获取信息的重要渠道，同时也是众多商家广告投放的关键途径。据 2023 年的数据显示，互联网广告发布收入达到 7190.6 亿元，同比增长 33.4%。

二、计算广告的业务流程

在传统的计算广告业务流程中，广告展示需要涉及多个参与方，包括**用户、广告主、媒体平台、广告交易平台（ADX），以及需求方平台（DSP）**等。并且，这一过程通常需要经历十几次的交互才能最终完成广告展示。

****主要流程****

*****设置广告需求****：广告主在 DSP（需求方平台）上设置广告计划、预算、目标等。

*****发起访问请求****：用户在媒体平台上浏览内容或进行搜索等操作。

*****产生广告请求****：当媒体检测到广告位需要展示广告时，会向 ADX（广告交易平台）发起广告请求。

*****发起曝光竞标请求****：ADX 收到媒体的广告请求后，会向多个 DSP 并行发起曝光竞标请求。

*****曝光参与请求同步****：DSP 将曝光参与请求同步给广告主，询问广告主的决策。

*****决策并反馈给 DSP****：广告主根据实时数据和策略做出决策，并将决策结果反馈给 DSP。

*****DSP 决定是否出价****：DSP 根据广告主的决策和自身估值模型决定是否出价，并给出此次曝光的报价。

*****ADX 返回竞标结果****：ADX 集齐 DSP 报价后进行拍卖，并将竞标结果返回给 DSP。

* **广告展示：**ADX 按照媒体广告模板进行样式渲染后，将获胜 DSP 的广告返回给用户展示。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_24b66d6.png)

在互联网广告业务中，整个广告展示过程需要控制在100 到 200 毫秒之内完成，以确保用户体验和企业效益的最大化。在这一高效率要求中，数据库的角色至关重要。它不仅影响广告投放的效率，也直接关系到广告展示的性能。其中，对数据库的要求如下：

* **快速读写能力：**广告系统频繁地更新和检索数据，包括用户行为数据、广告表现数据等，广告加载时间对用户体验至关重要，因此需要一个能够支持高速读写操作的数据库。

* **可扩展性：**随着用户量和数据量的增加，数据库需要能够水平和垂直扩展，以处理更多的查询和存储需求。

* **高可用性：**对于广告系统来说，数据库的高可用性至关重要，应有灾难恢复计划和备份机制，以应对数据中心故障等极端情况以确保广告服务的连续性和稳定性。

三、广告行业数据库解决方案的演进

* 基于业务场景的演进，同时结合业务场景中出现的问题，广告主也在不断升级业务解决方案。在广告行业发展的早期阶段，广告主选择采用高性能的 Redis 内存型数据库，来满足整个业务流程对低延迟的诉求。

* 然而，随着广告行业规模的增大导致整个广告系统的数据存储成本、维护成本也会增加。为了解决这一问题，许多系统采用了 **Redis+MySQL 的架构**（下文将详细介绍）。后续随着广告行业更进一步规模的扩大、数据量剧

增，Redis+MySQL 的架构方案会出现**数据一致性维护成本高**的问题，业界开始尝试使用新的方案解决这一问题。

* 为了解决数据一致性维护成本等一系列的问题，同时基于广告客户的业务场景的需求，百度智能云推出了 PegaDB。

PegaDB 采用内存+磁盘存储结合的方式，通过内核自研**冷热数据分离与交换、异地多活、Bulkload** 等企业级特性来满足现代广告系统对高效率、高可用性和低成本的综合需求。

四、百度智能云数据库 Redis 容量型（PegaDB）

结合广告场景中的具体问题，百度智能云数据库 PegaDB 在**产品层、架构层、业务应用层**做了全面的提升，在产品层，通过内存 + 磁盘存储结合的方式，内核自研，冷热数据分离与交换，解决了数据同步一致性的问题；在架构层，采用异地多活的高可用架构，实现了跨地域数据灾备与自动化同步；在业务应用层，PegaDB 数据快速灌库 Bulkload 解决了传统数据库只能通过标准协议逐条写入数据，写入速度慢、资源占用影响线上业务的问题，数据导入速度相比传统提升 5 - 10 倍。帮助多家企业提升了业务处理的效率和准确性，降低运营成本。

4.1 冷热数据分离与交换

传统方案与挑战

考虑到兼顾性能与成本，传统模式下广告行业通常采用的是 **Redis + MySQL** 架构设计，热数据存放在内存型数据库 Redis 中，全量存储数据存放在 MySQL。服务会优先访问 Redis 数据库，获取有效数据即返回数据。如果在 Redis 中没有找到所需数据，系统便会查询 MySQL，获取数据后需要运维人员手工将数据再次同步回 Redis。在海量场景下，这种情况频繁发生，不仅会拖慢系统性能，同时也会增加业务人员的工作量。

百度智能云 PegaDB 解决方案

自研Redis 容量型（PegaDB）产品，基于百度智能云增强型 SSD 云盘构建，将内存与磁盘相结合，内存支持毫秒级在线数据处理，磁盘提供数据持久化能力，实现数据高可靠。同时内核自研的冷热数据分离与交换功能，解决了数据同步一致性的问题。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_93ce21b.png)

百度智能云数据库 Redis 容量型（PegaDB）可对广告位的配置信息以及策略信息等热点数据进行冷热分离。更新 Cache 的逻辑设定为每 10 秒钟一个周期，每周期清空 LRU Cache 中的旧数据（即最长时间未被访问的数据，被视为冷数据），同时更新最新的热点数据，将热点数据加载到缓存中。热点数据在内存中实现高效访问，而非频繁访问的冷数据存储在本成本较低的磁盘中。此外，借助 LRU 算法淘汰机制实现冷热数据高效切换，有效避免人工大量处理的问题，节省人工成本。

**在处理大规模业务数据时，PegaDB 能显著降低成本，单 GB 成本比传统 Redis 解决方案降低超过 70%，同时保持了接近原有 Redis 性能的 80%。 **

具体实现

* PegaDB 会维护一个 LRU Cache，用于缓存热数据，通过 LRU 机制淘汰冷数据。

* 数据读取时，优先访问 LRU Cache 中的 key，若命中则返回 Cache。若未命中则从 DB 读取数据，读取后自动数据同步到 LRU Cache，下次就可以在 LRU Cache 中直接访问到该数据。

* 通过读操作可将最新的值缓存到 Cache 中，更新 Cache 的逻辑一般以 10s 为周期，每 10s 进行一次操作，清空 LRU Cache 的旧数据，更新最新的热数据。时间周期可调整。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_6ec2216.png)

4.2 异地多活

****传统方案与挑战****

在多地域数据管理中，保持数据的同步性常常面临着诸多挑战。不同地域的数据信息同步需要使用专业的数据复制和同步工具（如 GoldenGate、SharePlex 等）手动写入，并且操作过程中需要专业人员执行、维护。此外，还有一种方法是将数据库的事务日志定期或实时传输到远程地点，然后在远程数据库上重放这些日志来实现数据同步。尽管这种方法可以在不同地域之间同步数据，但整个过程复杂性高、时间周期长。

****百度智能云 PegaDB 解决方案****

Redis 容量型（PegaDB）自研多活实例组产品。多活实例组由至多 5 个实例组成，每个实例均可以进行本地域读写。无需专业人员手动写入、维护，每个地域数据由同步组件自动化同步至其他地域，省时省力。多活实例组应用场景如下：

* 异地多活：业务需求应用就近访问同时数据整体一致。

* 数据灾备：可实现同城灾备、两地三中心灾备及三地灾备等多种数据灾备场景。

百度智能云数据库 Redis 容量型（PegaDB）异地多活同步通道单向可达 5 万 QPS。本地域读写与单实例的延迟基本一致。跨地域间同步，延迟几十毫秒至几百毫秒。不仅通过异地多活的架构设计解决了多地域数据同步一致性的问题。同时，数据自动化同步至其他地域，无需专业数据复制与同步工具手动写入，节省了人力成本。

****具体实现****

* 在异地多活集群中，每个实例都会为其配置一个数据同步组件，这个同步组件的叫做 SyncAgent。单个分片内部，只有主节点的 SyncAgent 会工作。SyncAgent 会解析 Redis 生成的 AOF 并将增量的 AOF 同步到目标集群。

* 在 PegaDB 架构中，引入 operator header，用于为生成 AOF 中的每一条命令增加一个编号，这样 SyncAgent 就可以记录下 opid 来追踪同步进度了，当 SyncAgent 故障重启后，SyncAgent 会从之前的同步位点继续同步，避免数据丢失。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_af51c85.png)

****4.3 Bulkload 快速灌库********传统方案与挑战****

当双十一大促等特殊活动时，客户需要增加广告投放和活动预算，此时需要将大批量数据快速导入数据库。大批量的数据导入占用资源会影响线上业务。按照传统协议的命令，批量写入数据一方面写入速度有限，另一方面写入数据占用资源会使得线上业务访问延迟增加，影响用户体验。

****百度智能云 PegaDB 解决方案****

数据灌库（Bulkload）依托 RTA 业务场景大数据平台的高并发处理能力和自身存储引擎的数据编排能力，支持大规模的数据通过专属通道直接传入存储引擎，数据导入速度相比传统提升 5-10 倍，并降低对在线业务的影响。

百度智能云数据库 Redis 容量型（PegaDB）离线灌库方案从数据库层面解决了业务场景批量离线快速灌库、全量数据原子生效、秒级切库和回滚等需求。业务层面，解决了传统数据库只能通过标准协议逐条写入数据，写入速度慢、资源占用影响线上业务的问题。

****具体实现****

* **准备原始数据**：使用百度智能云 BOS 服务，业务将原始 KV 数据以指定的格式写到文件并上传到 BOS。

* **生成 SST 文件**：借助 BMR 平台，业务向 BMR 平台提交 MR 作业，业务需要指定两个 BOS 地址，分别为原始 KV 数据文件 BOS 地址和生成 SST 文件输出的 BOS 地址。作业的功能是将原始 KV 数据转化为 PegaDB 能够识别和加载的 SST 文件。

* **注入 SST 文件**：业务提交注入任务，注入完成后，新注入的 SST 文件对 PegaDB 可见，但对业务不可见。每一批生成的 SST 文件会对应一个命名空间 namespace，注入完成后，集群会新增一个 namespace，新 namespace 则表示刚注入的数据。（通过 API 可以查询）

* **数据切换**：业务提交切换任务，将数据切换到新注入的 namespace，切换完成后，线上访问的对象为新注入的数据。切换耗时为秒级，不断连不阻写。（若数据不合预期，业务可以提交切换任务回切到老的 namespace）

* **数据删除**：切换完成后，老数据还会占用磁盘空间，业务可以提交命令空间删除任务，删除耗时为分钟级，任务完成后，磁盘空间会立即腾出，删除过程不断连不阻写。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_42d93f5.png)

五、方案总结与客户案例

PegaDB 通过在产品层、架构层、业务应用层的提升，帮助多家企业提升了业务处理的效率和准确性，显著降低了运营成本。

* **产品层**：PegaDB 通过内存+磁盘存储的系统设计，数据冷热分离自动切换实现了数据高一致性，兼顾存储成本与性能的情况下降低了数据库的运维成本。

* **架构层**：PegaDB 采用异地多活的高可用架构，实现了跨地域数据灾备与同步，业务场景上解决了不同地域之间数据高性能读写的需求。

* **业务应用层**：PegaDB 数据快速灌库 Bulkload 在业务应用层面，显著提升了业务数据写入速度，效率提升 5 倍+。

目前，度秘广告业务已将 PegaDB 作为其广告系统数据库，用于存储广告物料、投放数据和用户画像等。此外，通过 PegaDB 对缓存数据和物料耗时数据的优化，实现了数据之间的高效同步和大规模数据批量导入效率的显著提升，实现运维成本降低 50%，不同地域业务数据之间高效同步，大规模数据批量导入效率提升 5 倍+。

性能降低 10%+，个别地域业务数据访问延迟增加，大规模数据批量写入效率提升 10%+。

产品定价

计费说明

计费方式说明

计费项

云数据库Redis版的计费组成如下。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_fd80280.png)

<p></p>

<table>

<tr><th style="width:80px">计费项</th><th>说明</th></tr>

<tr>

<td>实例规格</td>

<td>

费用说明：创建实例时，根据实例规格和购买时长产生费用

计费方式：分为预付费和后付费，具体可参考下文计费方式介绍

</td>

</tr>

<tr>

<td>实例日志</td>

<td>

费用说明：日志服务当前免费使用

日志说明：创建实例时，您可以在控制台上开通实例的相关日志服务，包括慢日志、运行日志

</td>

</tr>

</table>

<p></p>

计费方式

<p></p>

<table>

<tr><th style="width:80px">计费方式</th><th>优势</th><th>说明</th><th>注意事项</th></tr>

<tr><td>预付费（包年包月计费）</td><td>价格便宜</td><td>即包年、包月付费，在创建实例时需要支付费用适合业务稳定的长期需求，费用比后付费更低，且购买时间越长，折扣越多</td><td>预付费实例支持升级配置，不支持降级配置和释放预付费实例支持退订。</td></tr>

后付费（按量计费）	灵活调整	即小时计费，在北京时间整点扣费并生成账单	出账单时间是当前计费周期结束后1小时内	适合业务瞬间有较大变化的场景，资源使用完即可立即释放，降低成本
后付费实例可随时被释放	后付费实例支持变更配置（配置升级/降级）、计费变更（后付费转预付费）	购买前需保证账户无欠款，且账户余额和可用代金券总和 ≥ 100 元		

相关文档

- [预付费](https://cloud.baidu.com/doc/SCS/s/njwvxtvzq)
- [后付费](https://cloud.baidu.com/doc/SCS/s/yjwvxtwbs)
- [续费说明](https://cloud.baidu.com/doc/SCS/s/fkck53fe4)
- [余额不足提醒和欠费处理](https://cloud.baidu.com/doc/SCS/s/dkckbwnbl)

计费方式变更

概述

云数据库 Redis 实例的支持预付费（包年包月）和后付费（按需计费）两种计费方式。

计费方式之间支持相互变更，即支持预付费转后付费、后付费转预付费。

预付费转后付费

- * 预付费转后付费后，计费方式变更将在预付费合同到期后立即生效。
- * 预付费转后付费后，合同未到期前，实例的支付方式状态为「预付费（待变更）」，此阶段可取消「计费变更」。
- * 预付费转后付费前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元，避免计费方式变更失败。
- * 对于已启用自动续费的预付费实例，不能进行计费变更。

后付费转预付费

实例的计费方式可以从后付费变更为预付费，且计费方式变更成功后，将立即按照预付费方式计费。

变配计费说明

云数据库 Redis 支持按需升降配，帮助您提高资源利用率，优化成本结构。

变配流程及影响

变配完成后，源实例的实例ID、连接地址、数据、白名单配置以及已创建的账号密码配置等均不会改变。更多关于变更配置的流程、影响及操作方法，请参见[变配配置](https://cloud.baidu.com/doc/SCS/s/0kck8qbzw)

变配规则

<p></p>

计费方式	升级配置	降级配置
计费说明		
预付费	- Redis 标准版实例支持升级节点规格、副本数量。 - Redis 集群版实例支持升级分片数量、副本数量。 - Memcached 实例支持升级分片数量。	暂不支持降级配置。
对于未到期的实例，升级配置所需费用 = (升级后配置的单日价格 - 升级前配置的单日价格) × 到期前的剩余天数		
后付费	- Redis 标准版实例支持升级节点规格、副本数量。 - Redis 集群版实例支持升级分片数量、副本数量。 - Memcached 实例支持升级分片数量。	- Redis 标准版实例支持降级节点规格、副本数量。 - Redis 集群版实例支持降级分片数量、副本数量。
配置变更成功后，将立即按照新配置的价格进行计费。		

<p></p>

版本间变配

- Redis 标准版可以变配到集群版。

- Redis 集群版不允许变配到标准版。

续费说明

概述

- 预付费实例：增加使用时间。
- 后付费实例：无到期时间，无须续费。
- 续费前：请保证账户余额充足，余额 > 100元。

续费说明

云数据库 Redis 支持**手动续费**和**自动续费**两种续费方式，用户可以根据不同的应用场景选择合适的续费方式，推荐使用自动续费。两种不同续费方式的使用场景如下：

- **手动续费**：初期使用云数据库 Redis，有运维人员按时进行续费操作。
- **自动续费**：适用于长期使用云数据库 Redis，对服务可用性、数据安全性、运维自动化要求更高的场景。

相关文档

- [手动续费](https://cloud.baidu.com/doc/SCS/s/Tkc1rabxp)
- [自动续费](https://cloud.baidu.com/doc/SCS/s/kkc1rik6e)

账单和用量查询

您可以通过百度智能云控制台查看云数据库 Redis 的消费明细。

查看月账单概览

在右上角财务的消费总览页面，可以查看消费金额、已还款金额以及欠款金额，并可以直接进行还款、导出盖章版PDF文件等操作。详情请参见[消费总览](https://console.bce.baidu.com/billing/#/consumption/overview)。

查看账单详情

在右上角财务的账单明细页面，可以查看流水账单，资源账单，计费项账单等账单详情。详情请参见[账单明细](https://console.bce.baidu.com/billing/bill/hour/list)

退款说明

本文介绍云数据库 Redis 实例的退款说明与退订规则。

注意事项

- 仅运行状态为运行中的实例允许被退订。
- 实例退订后会进入回收站，金额会原路返回，在实例被删除前可以重新开机计费，详情参考[实例回收站](https://cloud.baidu.com/doc/SCS/s/xl3bc1tuw)。

退订按量付费实例

按量付费的 Redis 实例会根据实际使用情况产生账单，因此不涉及退款。如果您不再需要使用按量付费的Redis实例，可释放 Redis 实例，释放后将不再进行计费。

退订包年包月实例

包年包月 Redis 实例需根据实例规格和购买时长预先支付费用。如果您不再需要使用包年包月的Redis实例，需要进行退费工单咨询。

预付费

Redis

计费概述

云数据库 Redis 包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。具体价格请参考：[Redis价格详情](https://cloud.baidu.com/product-price/scs.html)。

计费规则

- Redis 引擎计费项：缓存节点、代理节点。
- 购买实例前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。
- 预付费方式，在创建实例时就需要预先支付费用。
- 在合同期内，包年包月的实例不支持释放。
- 对于包月的实例，购买1年/2年/3年即可分别享受8.3折/7折/5折优惠。

实例价格

计费公式

<!--乘号用转义符表示为 ∗-->

费用 = (缓存节点单价 ∗ 缓存节点数量 + 代理节点单价 ∗ 代理节点数量) ∗ 使用时长

>注：缓存节点数量 = 副本数量 ∗ 分片数量

计费示例

以华北-北京区域购买实例为例：

- * 购买标准版，缓存节点cache.n1.medium（4GB），双副本，一个月费用是170∗2=340元。
- * 购买集群版，缓存节点cache.n1.xlarge（16GB），双副本，8个分片，8个代理节点，一个月费用是

购买标准版，缓存节点cache.n1.medium（4GB），双副本，一年费用是650∗2∗8+100∗8=11200元。

* 购买标准版，缓存节点cache.n1.medium（4GB），双副本，一年费用是(170∗2)∗12∗0.83=3386.4元。

到期提醒和处理

- 到期提醒：服务到期前7天、3天、1天，系统均会给您发送即将到期提醒通知。

- 到期后处理：到期后立即停服，系统会发送欠费停服通知。实例为您保留7天，期间不收取费用，7天内未充值则释放，释放前1天和释放时系统都会发送释放通知。

Memcached

概述

云数据库 Redis for Memcached 包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。具体定价请参考：[Redis价格详情](https://cloud.baidu.com/product-price/scs.html)。

计费规则

- Memcached 引擎计费项：缓存容量。
- 购买实例前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。
- 预付费方式，在创建实例时就需要预先支付费用。
- 在合同期内，包年包月的实例不支持释放。
- 对于包月的实例，购买1年/2年/3年即可分别享受8.3折/7折/5折优惠。

计费公式

<!--乘号用转义符表示为 ∗-->

****费用 = 缓存容量单价 ∗ 缓存容量 ∗ 使用时长****

计费示例

以华北-北京区域购买实例为例：

* 购买10GB，一个月费用是54∗10=540元。

* 购买10GB，一年费用是(54∗10)∗12∗0.83=5378.4元。

到期提醒和处理

- 到期提醒：服务到期前7天、3天、1天，系统均会给您发送即将到期提醒通知。

- 到期后处理：到期后立即停服，系统会发送欠费停服通知。实例为您保留7天，期间不收取费用，7天内未充值则释放，释放前1天和释放时系统都会发送释放通知。

后付费

Redis

云数据库 Redis 按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。具体定价请参考：[Redis价格详情](https://cloud.baidu.com/product-price/scs.html)。

计费规则

- Redis 引擎计费项：缓存节点、代理节点。
- 购买实例前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。
- 按分钟计费，不足1分钟按1分钟计。

- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 后付费方式，在生成账单后自动从账户余额或可用代金券中扣除费用结算账单。
- 按需计费的实例可以随时释放，释放后不再计费。

实例价格

计费公式

<!--乘号用转义符表示为 ∗-->

****费用 = (缓存节点单价 ∗ 缓存节点数量 + 代理节点单价 ∗ 代理节点数量) ∗ 使用时长****

> 注：缓存节点数量 = 副本数量 ∗ 分片数量

计费示例

以华北-北京区域购买实例为例：

- * 购买标准版，缓存节点cache.n1.medium（4GB），双副本，单日费用是(0.005903∗2)∗60∗24=17.00064元。
- * 购买集群版，缓存节点cache.n1.xlarge（16GB），双副本，8个分片，8个代理节点，单日费用是(0.022569∗2∗8+0.003472∗8)∗60∗24=559.9872元。

Memcached

云数据库 Redis for Memcached 按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。具体定价请参考：[Redis价格详情](https://cloud.baidu.com/product-price/scs.html)。

计费规则

- Memcached 引擎计费项：缓存容量。
- 购买实例前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。
- 按分钟计费，不足1分钟按1分钟计。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 后付费方式，在生成账单后自动从账户余额或可用代金券中扣除费用结算账单。
- 按需计费的实例可以随时释放，释放后不再计费。

实例价格

计费公式

<!--乘号用转义符表示为 ∗-->

****费用 = 缓存容量单价 * 缓存容量 * 使用时长****

计费示例

以华北-北京区域购买实例为例：

- * 购买10GB，单日费用是(0.001667*10)*60*24=24.0048元。

到期和欠费处理

Redis 实例到期或欠费后，在一定时间内可以激活或重建实例。

2. [连接实例](SCS/快速入门/连接实例.md#连接Redis实例)：在云服务器 BCC 使用 `redis-cli` 通过域名访问 Redis 实例。

3. 管理数据库：开始数据交互，命令详情参考“支持的Redis命令”。

Redis~~~~ for Memcached实例

1. [创建实例](SCS/快速入门/创建实例.md#创建Memcache实例)：登录并根据业务需要购买云数据库 Redis for Memcached 实例。

2. [连接实例](SCS/快速入门/连接实例.md#连接Memcached实例)：在云服务器 BCC 使用 `telnet` 通过域名访问 Memcached 实例。

3. 管理数据库：开始数据交互。

创建实例

本文为您介绍通过云数据库 Redis 管理控制台创建实例的基本方法。

创建 Redis 实例

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。
2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis-实例列表”页面。

3. 点击**创建实例**，在实例创建页面，选择计费方式：

* **预付费**：包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。

* **后付费**：按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。

4. 设置以下参数。

参数	
说明	
地域	实例所在的地域，即实例所在的地理位置。如需修改为其他地域，请通过主导航进行切换。
- 购买后不能更换，请谨慎选择。 - 实例仅支持同 VPC 内设备访问，请确保与需要连接的 BCC 实例创建于同一个地域。	
可用区	可用区 地域 内的独立物理区域。部分地域支持多可用区：
- 单可用区：可降低访问时延，但不再具有跨机房灾备优势。 - 多可用区：提高数据库可用性，但可能增加访问时延。	

<td>引擎类型</td>	
<td>	数据库引擎的类型，当前支持Redis和Memcache，这里选择Redis。</td>
</tr>	
<td>架构类型</td>	
<td>	不同的实例架构类型适用于不同场景，创建后不支持更改。
	标准版：标准版架构的 Redis 实例只包含一个分片。兼容 Redis 3.2/4.0 版本的协议和命令，是最通用的实例类型。
	集群版：集群版架构的 Redis 实例包含多个分片，以及多个代理节点。兼容 Redis 3.2 版本的协议和命令，采用分布式架构，支持横向和纵向（即将上线）的扩缩容。 集群版实例能够通过横向扩展分片数量，快速扩展数据容量，提升网络/CPU/QPS能力，可承载远高于标准版实例的流量压力。
</td>	
</tr>	
<tr>	
<td>引擎版本</td>	
<td>	
	6.0/5.0/4.0/3.2：支持标准版、集群版。推荐使用当前主流版本Redis 6.0。
	2.8：支持标准版。
</td>	
</tr>	
<tr>	
<td>存储类型</td>	
<td>	
	高性能内存：使用内存存储全量数据集，性能卓越。
	分级存储型(公测)：冷热数据分层存储，成本更低，容量更大。暂仅支持集群版。
</td>	
</tr>	
<tr>	
<td>节点规格</td>	
<td>	用于保存数据的缓存节点的规格，不同规格的缓存节点提供不同的容量和性能。详细信息请参见 Redis标准版 、 Redis集群版 。
</td>	
</tr>	
<tr>	
<td>分片数量</td>	
<td>	仅适用于集群版实例。
	分片是一组缓存节点的集合，当实例包含多个分片时，数据集被切分到多个分片中，每个分片拥有完整数据集的子集。
	实例的总容量等于单个缓存节点的容量与分片数量的乘积。

```

    </td>
  </tr>
<tr>
  <td>副本数量</td>
  <td>
    <ul>
    <li>
      单副本：单分片中只包含1个节点，该节点充当可读写的主节点。单副本的实例支持读写高并发，但不支持备份或持久化，不提供数据可靠性，适用于纯缓存等对数据可靠性要求不高的业务场景。
    </li>
    <li>
      双副本：单分片中包含2个节点，其中一个节点充当可读写的主节点，其余节点充当不可读写且用户不可见的备节点，当主节点故障时，备节点自动升级为主节点。双副本的实例支持秒级故障切换，支持备份和持久化，提供高可靠、高可用特性，适用于各类应用场景。
    </li>
    </ul>
  </td>
</tr>
<tr>
  <td>私有网络
  </td>
  <td>私有网络即 VPC。云数据库 Redis 实例仅允许相同 VPC 内的设备访问，请确保与需要连接的 BCC 实例创建于同一个 VPC。仅支持选择“通用型”子网。创建后暂不支持更改。
  </td>
</tr>
<tr>
  <td>端口
  </td>
  <td>端口可选1025-65534，不支持22222，默认6379。创建后暂不支持更改。
  </td>
</tr>
<tr>
  <td>实例名称
  </td>
  <td>由大小写字母、数字以及_/.特殊字符组成，必须以字母开头，长度1-64，名称不可重复。
  </td>
</tr>
<tr>
  <td>实例密码
  </td>
  <td>
    <ul>
    <li>暂不设置：创建实例后可随时对密码进行设置或修改。
    </li>
    <li>立即设置：在下方设置密码。
    </li>
    </ul>
  </td>
</tr>
<tr>
  <td>自动备份
  </td>
  <td>
    <ul>
    <li>启用后，系统将在指定时间自动对数据集执行RDB快照备份。可通过展开备份策略设置对默认的备份策略进行修改。
    </li>
    <li>不适用于单副本的实例。</li>
    </ul>
  </td>
</tr>
<tr>
  <td>购买信息</td>
  <td>
    <ul>

```

```

<li>购买时长(预付费)：选择所需的时长，系统会自动计算对应的配置费用，购买1年及以上有额外优惠。
</li>
<li>实例个数：用于批量购买相同规格的实例，最多可购买10个 Redis ~~~~实例。
</li>
<li>自动续费（预付费）：
<ul>
<li>默认不勾选，不进行自动续费。
</li>
<li>勾选后实例自动续费，自动续费周期与原订单周期一致。
</li>
</ul>
</li>
</ul>
</td>
</tr>
</tbody>
</table>

```

5. 确认配置信息后，点击**下一步**按钮进入付款页面，可以在这里看到选择的实例的套餐和价格，进行确认。
6. 确认后点击右下角的**去支付**按钮，跳转进入“支付”页面，确认付款后订单完成。
7. 购买成功后，返回“云数据库 Redis-实例列表”页面，可看到已创建的实例，以及实例状态、位置、数据库类型等信息。其中，预付费实例还可以看到 Redis 实例的到期时间。

创建 Redis for Memcache 实例

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。
2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis-实例列表”页面。
3. 点击**创建实例**，在实例创建页面，选择计费方式：

* **预付费**：包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。

* **后付费**：按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。

4. 设置以下参数。

```

<table>
<thead>
<tr>
<th style="width:80px">参数</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>地域</td>
<td>
实例所在的地域，即实例所在的地理位置。如需修改为其他地域，请通过主导航进行切换。
<ul>
<li>购买后不能更换，请谨慎选择。
</li>
<li>实例仅支持同 VPC 内设备访问，请确保与需要连接的 BCC 实例创建于同一个地域。
</li>
</ul>
</td>
</tr>
<tr>
<td>可用区</td>
<td>
<b>可用区</b>地域内的独立物理区域。部分地域支持多可用区：
<ul>
<li>单可用区：可降低访问时延，但不再具有跨机房灾备优势。
</li>
<li>多可用区：提高数据库可用性，但可能增加访问时延。
</li>
</ul>
</td>
</tr>

```

<td>引擎类型</td>	
<td>	数据库引擎的类型，当前支持Redis和Memcache，这里选择Memcache。</td>
<td>节点规格</td>	
<td>	用于保存数据的缓存节点的规格，不同规格的缓存节点提供不同的容量和性能。详细信息请参见 Memcached集群版 。
<td>分片数量</td>	
<td>	仅适用于集群版实例。
	- 分片是一组缓存节点的集合，当实例包含多个分片时，数据集被切分到多个分片中，每个分片拥有完整数据集的子集。 - 实例的总容量等于单个缓存节点的容量与分片数量的乘积。
<td>私有网络</td>	
<td>	私有网络即 VPC。云数据库 Redis 实例仅允许相同 VPC 内的设备访问，请确保与需要连接的 BCC 实例创建于同一个 VPC。仅支持选择“通用型”子网。创建后暂不支持更改。
<td>端口</td>	
<td>	端口可选1025-65534，不支持22222，默认6379。创建后暂不支持更改。
<td>实例名称</td>	
<td>	由大小写字母、数字以及_/.特殊字符组成，必须以字母开头，长度1-64，名称不可重复。
<td>购买信息</td>	
<td>	- 购买时长(预付费)：选择所需的时长，系统会自动计算对应的配置费用，购买1年及以上有额外优惠。 - 实例个数：用于批量购买相同规格的实例，最多可购买10个 Redis 实例。 - 自动续费（预付费）： - 默认不勾选，不进行自动续费。 - 勾选后实例自动续费，自动续费周期与原订单周期一致。

5. 确认配置信息后，点击**下一步**按钮进入付款页面，可以在这里看到选择的实例的套餐和价格，进行确认。
6. 确认后点击右下角的**去支付**按钮，跳转进入“支付”页面，确认付款后订单完成。
7. 购买成功后，返回“云数据库 Redis-实例列表”页面，可看到已创建的实例，以及实例状态、位置、数据库类型等信息。其中，预付费实例还可以看到 Redis ~~~~实例的到期时间。

常见问题

- 创建实例需要多长时间？

答：创建实例消耗的时间与实例的分片数成正相关。实例的分片数越多，需要的调配的资源越多，创建需要的时间则越长。

相关 API

- API 接口：[创建实例](SCS/API参考/订单服务/创建实例.md)

连接实例

本文介绍通过DMS或redis-cli快速连接云数据库Redis 实例和 Memcached 实例的基本方法。

前提条件

用户可以通过内网和公网访问云数据库 Redis 数据库。

- 内网连接：[设置白名单](https://cloud.baidu.com/doc/SCS/s/Ejz3v637y)并用内网域名连接。

- 公网连接：[设置白名单](https://cloud.baidu.com/doc/SCS/s/Ejz3v637y)并[开通公网访问](https://cloud.baidu.com/doc/SCS/s/lkm06kj3i)，用公网IP或域名连接。

连接 Redis 实例

> **前提条件：云数据库 Redis 仅支持百度智能云内网访问，即只有在相同 VPC 内的设备上安装 `redis-client` 才能与 Redis 建立连接并进行数据操作。所以，在连接之前，您需要先购买一个云服务器 BCC。

通过 redis-cli连接

1. 创建云服务器 BCC 实例，具体操作请参见[创建 BCC 实例](https://cloud.baidu.com/doc/BCC/s/wkap3s48k)。

2. 登录云服务器 BCC，通过命令行执行 `yum install redis`，完成 Redis 客户端的安装。

3. 通过命令行使用 Redis 客户端 `redis-cli` 工具，[远程访问购买的 Redis 实例](https://cloud.baidu.com/doc/SCS/s/clwqbqxw)。

* **命令格式为：`redis-cli -h Redis实例域名 -p Redis实例端口号`**。

5. 完成连接后可输入命令使用，若提示 **NOAUTH Authentication required**，需要账号密码认证。

* 输入**AUTH password** 进行认证后即可使用。默认账号的密码为创建实例填写的登陆密码**。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_49e4058.png)

> **参数说明：*

Redis实例域名：实例连接地址，您可以在实例基本信息的网络信息区域获取实例的内网连接域名。

Redis实例密码：数据库账号管理模块下各个账号密码。（不同账号可能不同）

Redis实例端口号：端口号，默认为6379。您可以在实例基本信息的网络信息区域获取实例的端口号。

通过 DMS 连接

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。

2. 在控制台页面的实例列表中，点击实例名称进入实例信息页面，点击**数据开发**。

3. 按照提示开通相关权限后即可使用**DMS**进行**数据开发**。此方式将使用系统默认账号进行登录数据库。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_c9dfe46.png)

连接 Redis for Memcached 实例

- 1. 创建云服务器 BCC 实例，具体操作请参见[创建 BCC 实例](https://cloud.baidu.com/doc/BCC/s/wkap3s48k)。
- 2. 登录云服务器 BCC，使用 telnet 通过域名访问 Memcached 实例。

操作指南

Redis操作指南

功能概览

以下将介绍 Redis 各版本支持的功能，便于您根据需求进行了解和查询。

Redis 功能概览									
分类		功能详情		社区版标准版				企业版集群版	
实例版本		引擎版本		>7.0		>6.0		>5.0	
>6.0		>5.0		>4.0		>7.0		>6.0	
实例管理	创建实例								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
	修改实例名称								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
	重启实例								
实例变更	支持	支持	支持	支持	支持	支持	支持	支持	支持
	释放实例								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
清除数据		支持	支持	支持	支持	支持	支持	支持	支持
实例变更	节点规格变更								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
实例续费		分片数量变更							
实例续费		支持	支持	支持	支持	支持	支持	支持	支持
实例续费		自动续费							
实例续费		支持	支持	支持	支持	支持	支持	支持	支持
实例续费		手动续费							
实例续费		支持	支持	支持	支持	支持	支持	支持	支持
实例续费		计费变更							
实例续费		支持	支持	支持	支持	支持	支持	支持	支持
实例升级	升级小版本								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
实例升级		升级大版本							
实例升级		不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
账号管理	创建账号								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
	修改账号权限								
	支持	支持	支持	支持	支持	支持	支持	支持	支持
重置密码		支持	支持	支持	支持	支持	支持	支持	支持
删除账号		支持	支持	支持	支持	支持	支持	支持	支持
连接管理		连接实例							
连接管理		支持	支持	支持	支持	支持	支持	支持	支持

<p></p>

云数据库 Redis 管理控制台用于管理 Redis 实例，您可以通过控制台上用户界面进行创建实例、管理实例等操作。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_c2ac8de.png)

48

1. ****地域****：选择云数据库 Redis 所在区域。
2. ****实例****：实例列表页面，可以查看已创建的所有实例。
3. ****热活实例组****：提供热活实例架构管理的功能，用户可以管理热活实例组。
4. ****多活实例组****：提供多活实例架构管理的功能，用户可以管理多活实例组。
6. ****硬件待维护实例****：当实例所在的物理机达到了服务器的时使用年限时，需要将服务器上的相关服务进行迁移。比如RDS和SCS，首先TS会提前通知用户，然后OP会在后台帮用户迁移实例；在用户侧的感知是，业务可能有闪断。
7. ****回收站****：用户释放的实例会进入回收站，进入回收站的实例可以进行重新开机或者立即删除，用户保护用户的实例安全。
8. ****任务列表****：用户记录和管理定时执行的后台任务。
9. ****参数模板****：可以查看和管理保存的参数模板，并且应用到对应的实例。
10. ****标签管理****：标签管理功能，通过给每个云资源创建标签，从而快速分类和识别管理这些资源。您可以根据项目和场景需求，为云数据库 Redis 实例创建、编辑或删除标签，便于对云数据库 Redis 实例进行分类和识别管理。
11. ****资源管理****：通过给每个云资源分配到对应的资源，从而快速分类和识别管理这些资源。
12. ****配额管理****：用户可以根据实际的情况申请修改实例数量的配额，用于创建更多的实例。
13. ****创建实例****：创建实例页面的入口，您可以通过 Redis 管理控制台创建云数据库 Redis 实例和云数据库 Redis for Memcached 实例。
14. ****释放****：您可以手动释放实例，支持批量释放。
15. ****计费变更/取消****：计费变更/取消计费变更的入口，支持批量变更计费/取消变更计费。
16. ****编辑标签****：标签编辑入口，支持修改实例标签。
17. ****实例搜索栏****：支持按照不同的实例属性进行搜索。
18. ****刷新****：对实例列表页面进行刷新。
19. ****下载****：资源下载按钮，方便下载实例列表进行管理。下载的资源文件为 CSV 格式，编码格式为 UTF-8，文件命名规范为“产品名地域日期_list.CSV”。所下载的实例与当前实例列表的筛选条件保持一致，内容包含云数据库 RDS实例的所有字段。
20. ****自动续费****：续费管理入口，支持开通自动续费。
21. ****实例名称****：点击实例名称，进入实例详情页面。
22. ****续费****：续费入口，可以对单个预付费实例进行续费。
23. ****清空数据****：清空数据缓存，请谨慎操作。

实例生命周期

创建实例

概述

本文为您介绍通过云数据库 Redis 管理控制台创建实例的基本方法。

创建 Redis 实例

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。

2. 点击****创建实例****，在实例创建页面，选择计费方式与地域：

* ****预付费****：包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。

* ****后付费****：按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_d038e6b.png)

3. 设置以下参数。

| 参数 | 说明 |

| --- | --- |

| 数据库类型 | 数据库引擎的类型。支持Redis、Redis容量型（PegaDB）、Memcache。 |

| 版本与框架 | 分标准版、集群版两种类型，不同的实例架构类型适用于不同场景。 创建后支持标准版变更为集群版，不支持集群版变更为标准版。 |

| 数据库版本 | 不同版本数据库内核支持的功能存在一定差异，Redis建议使用6.0版本。 |

| 存储类型 | 分为高性能内存型、容量存储型。 |

| 实例规格 | 用于保存数据的缓存节点的规格，不同规格的缓存节点提供不同容量和性能。 |

| 存储磁盘 | 只有Redis容量型（PegaDB）有该选项，用于存储冷数据。 |

| 分片数量 | 分片是一组缓存节点的集合，当实例包含多个分片时，数据切分到多个分片中，每个分片拥有完整数据集的子集。实例的总容量等于单个缓存节点的容量与分片数量的乘积。 |

| 副本数量 | 单分片中只包含1个节点，该节点充当可读写的主节点。单副本的实例支持读写高并发，但不支持备份或持久化，不提供数据可靠性。单分片中包含2个节点，其中一个节点充当可读写的主节点，其余节点充当不可读写且用户不可见的备节点，当主节点故障时，备节点自动升级为主节点。双副本的实例支持秒级故障切换，支持备份和持久化，提供高可靠、高可用特性，适用于各类应用场景。 |

| 副本只读 | 开启副本只读后，主节点提供读写，从节点提供只读。 |

| 网络配置 | 可用区地域内的独立物理区域。部分地域支持多可用区：不同节点可选择不同的可用区。 |

| 实例名称 | 由大小写字母、数字以及_ / 特殊字符组成，必须以字母开头，长度1-64，名称不可重复。 |

| 是否自动备份 | 自动备份启用后，系统将在指定时间自动对数据集执行RDB快照备份。 |

| 实例密码 | 暂不设置：创建实例后可随时对密码进行设置或修改。立即设置：在下方设置密码。 |

| 购买数量 | 最多可购买 50 个实例。如需申请更多配额可点击 对应链接申请。 |

5. 确认配置信息后，点击**下一步**按钮进入付款页面，可以在这里看到选择的实例的套餐和价格，进行确认。
6. 确认后点击右下角的**去支付**按钮，跳转进入“支付”页面，确认付款后订单完成。**预付费实例会跳转支付订单页面，需要立即支付。后付费实例不会跳转新的支付订单页面，账户有余额即可。**

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_d0d8380.png)

8. 购买成功后，返回“云数据库 Redis-实例列表”页面，可看到已创建的实例，以及实例状态、位置、数据库类型等信息。

****常见问题****

- 创建实例需要多长时间？

答：创建实例消耗的时间与实例的分片数成正相关。实例的分片数越多，需要的调配的资源越多，创建需要的时间则越长。

****相关 API****

-[创建实例](https://cloud.baidu.com/doc/SCS/s/hk0qhwxom)

重启实例

****概述****

云数据库 Redis 提供手动重启实例的功能。本文为您介绍通过管理控制台进行重启实例的通用方法。

****注意事项****

<div style="background-color: rgb(229, 239, 250); padding: 10px;">

- 重启会有分片式*5秒内的闪断，超过16分片的实例，由于串行化处理，时间较长，请耐心等待，确保应用具有自动重连操作。
 - 对于单副本实例，实例重启将导致数据丢失，请谨慎操作。
 - 重启无法解决业务异常（如业务流量、请求量异常、存在[大Key](https://cloud.baidu.com/doc/SCS/s/Akb1qmozd)、[热Key](https://cloud.baidu.com/doc/SCS/s/Lllykaixn)等引发的 Redis 问题，建议优先在业务侧进行排查。
- </div>

****重启实例****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。
2. 在控制台页面，点击需要重启的**实例名称/ID**。
3. 在实例详情页的标签页导航栏中“实例详情页”，点击标题栏的**重启实例**按钮。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_5a2fb81.png)

4. 在弹出窗口中，选择重启执行时间，勾选风险说明后，单击**确定**按钮，即可执行重启任务，具体任务进度可在[任

务列表](https://cloud.baidu.com/doc/SCS/s/BI3bt1ct9)查看。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_6430dbe.png)

****相关API****

[重启实例](https://cloud.baidu.com/doc/SCS/s/jkx9uau8)

释放实例

****概述****

本文为您介绍通过管理控制台进行释放实例的通用方法，已释放的实例可在回收站查看。

****释放实例****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。

2. 在控制台页面的“云数据库 Redis 实例列表”页面，选择需要释放****实例名称/ID****。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_8482f66.png)

3. 点击****释放****按钮，弹出释放确认界面。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_f95763b.png)

> 说明：

> 预付费的Redis实例无法提前释放。

4. 点击****确定****按钮，弹出“安全验证页面”，进行手机验证码二次验证。

<p></p>

5. 输入****短信验证码****，点击****确认****按钮，完成释放操作。

****相关 API****

[释放实例](https://cloud.baidu.com/doc/SCS/s/Pk0rv2nlf)

续费实例

手动续费

****概述****

****手动续费****：默认续费方式，适用于初期使用且有运维人员按时进行续费操作的场景。

- 缺点：续费周期即将结束前需要手动续费，操作较为繁琐，可能遗忘续费而导致实例停止服务或数据丢失。

- 解决方案：设置自动续费续费周期，设置一次永久生效。

> 说明：

>

> - 预付费实例：有到期时间。若到期未续费，会导致业务中断甚至数据丢失。您可以根据业务需要，对“预付费”实例进行续费。

> - 后付费实例：无到期时间，无须进行续费相关操作。

****Redis 控制台续费实例****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)，进入“云数据库 Redis 实例列表”面

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)，进入云数据库 Redis 实例列表页面。

2. 选择需要续费的实例，点击此实例名称/ID，支付信息中点击续费，进入订单确认页面。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_e040be3.png)

3. 订单确认页面，确定续费时间、周期后点击下一步。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_ec5974a.png)

4. 继续点击下一步确认订单信息，完成支付后即续费成功。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_91df820.png)

****续费管理续费实例****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。

2. 在控制台右上方，点击财务跳转到新的页面后，点击续费管理，通过“**到期时间、产品筛选、地域选择**”筛选栏，找到目标续费实例。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_3bd7a25.png)

3. 勾选一个或多个实例，点击**续费**按钮。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_63a835d.png)

4. 先选择**按月**或**按年**进行续费，再确定具体续费时间长度，点击下一步。

!image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_2c18751.png)

> **注意**：

> 选择按年续费，最多续费3年；按月续费，最多续费9个月。

5. 确定好续费时间和续费金额后，点击右下角**下一步**按钮，进行订单付费，订单支付后，正式完成续费操作。

****相关API****

[续费实例](https://cloud.baidu.com/doc/SCS/s/Dkoe8ylw0)

自动续费

****概述****

自动续费：您可在控制台上设置预付费实例进行自动续费。若预付费实例到期未续费，会导致业务中断甚至数据丢失。

- 使用场景：适用于**长期使用**云数据库 Redis 实例的场景。

- **优点：操作简便，避免因遗忘续费导致数据删除业务中断的情况，安全性更高**。

****注意事项****

<div style="background-color: rgb(229, 239, 250); padding: 10px;">

- 开通自动续费后，系统会在到期日前7天自动为其续费。

- 请保证百度智能云账号的余额充足，避免续费失败。续费成功或失败都给予短信和邮件提醒。

- 自动续费将按照您最近一次设置的续费周期执行。

- 在一个周期内，资源的到期时间由于其他操作而延长，自动续费将在下个到期日再执行。

- 自动续费执行时，会针对某一个实例资源生成一个续费订单。

</div>

****创建实例时开通****

您可以在购买预付费实例时开通****自动续费****，自定义续费周期，选择按月（1~9个月）或按年（1~3年）进行自动扣费。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_1ec28b2.png)

****创建实例后开通****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)，进入“云数据库 Redis-实例列表”页面。
2. 在控制台右上方，选择****财务****，进入新的财务页面后点击****续费管理****。
3. 在续费管理页面选择****到期时间****，点击****云数据库Redis**** 页签，选择实例所在地域，找到目标实例。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_2d24e58.png)

4. 勾选目标实例，点击实例名称上方的****设置自动续费****按钮。
5. 在弹出的设置自动续费窗口中选择****续费周期****（按年或按月），点击****确定****按钮。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_de7afa1.png)

6. 自动续费开通成功，系统会发送短信和邮件通知。

****相关API****

[续费实例](https://cloud.baidu.com/doc/SCS/s/Dkoe8ylw0)

实例回收站

****概述****

本文为您介绍通过管理回收站对已删除的实例执行开机、彻底销毁等操作的通用方法。

****实例回收站****

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。
2. 在控制台页面的“云数据库 Redis-回收站”页面，根据业务需求执行****目标实例名称/ID****操作列中对应的操作。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_7104539.png)

3. 点击****开机****，弹出开机确认界面。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_c7f3cb3.png)

>**** 说明：**

> 暂不支持同时开机预付费实例和后付费实例。 ******

4. 点击****立即删除****按钮，弹出删除前确认页面。

![[image.png]](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_8623b65.png)

5. 点击****确定****按钮，即可完成删除

****相关 API****

- API 接口：[从回收站中批量恢复实例](https://cloud.baidu.com/doc/SCS/s/4koe8sdq6)
- API 接口：[释放回收站中的实例实例](https://cloud.baidu.com/doc/SCS/s/kkoe81r79)
- API 接口：[获取回收站内的实例列表](https://cloud.baidu.com/doc/SCS/s/Dkoe8ay72)

清空实例

您可以在控制台清除云数据库 Redis 版实例中的所有过期数据或所有数据。

操作步骤

1. 登录[云数据库 Redis 管理控制台](https://console.bce.baidu.com/scs/#/scs/list)。
2. 在控制台页面的实例列表中，根据业务需求执行点击**目标实例名称/ID**右侧的清空数据。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_0deddd5.png)

3. 弹窗信息确认执行后，则会执行清除数据任务，任务执行进度可在[任务列表](https://cloud.baidu.com/doc/SCS/s/BI3bf1cf9)查看。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_e116e12.png)

* **所有数据**：此操作将清除该实例所有数据，清除后数据无法找回。

* **指定DB**：此操作将清除该实例指定DB的所有数据，清除后数据无法找回。

* **过期数据**：此操作将批量清除该实例的所有过期数据，清除后数据无法找回。清理过期数据会占用一定CPU资源，建议在业务低峰期进行，避免影响线上服务。

注意：通过该功能清除的数据无法找回且即刻生效，同时可能会对线上业务造成影响，请谨慎执行。推荐在执行清除前对Redis实例的数据进行备份，具体操作，请参见[自动或手动备份](https://cloud.baidu.com/doc/SCS/s/tllup7rpi)。

相关API

[清空实例](https://cloud.baidu.com/doc/SCS/s/Tk0qia7qo)

读写分离

云数据库 Redis 提供读写分离的功能。开启后，您无需修改业务代码，实例能够自动识别读、写请求并进行对应转发，满足高并发读写的业务场景。本文为您介绍通过管理控制台启用和禁用读写分离的通用方法。

- > 启用副本只读后，主节点提供读写，从节点提供只读
- > 禁用副本只读后，主节点提供读写，从节点不可读写
- > 启用/禁用副本只读操作生效过程中可能会发生少于10秒的闪断。请选择业务低峰期进行，并确保应用具有自动重连机制
- > 仅支持集群版实例，不支持标准版实例

操作步骤

启用副本只读

1. 登录云数据库 Redis 管理控制台，点击「实例名称/ID」。

在实例详情页的标题右侧快捷栏中「实例详情」页，点击「配置信息」模块中副本只读外的「启用副本只读」按钮。

4. 在标题详情页的标题页导航栏中「标题详情」页，点击「配置信息」模块中副本只读处的「启用副本只读」按钮。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_cb297e2.png)

3. 在弹出窗口中单击「确定」。

****禁用副本只读****

1. 如果副本只读处于启用状态，点击「配置信息」模块中副本只读处的「禁用副本只读」按钮。

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image_6e2a12b.png)

2. 在弹出窗口中单击「确定」。

连接指南

Redis实例登录方式

<!--1、操作指南使用对象：所有人员。-->
 <!--2、本文档是模板，操作指南是针对具体任务的操作步骤。-->
 <!--3、模板的使用：段落的临近处都有写作指导，写之前先看指导。-->
 <!--4、标题与段落、段落之间要留出一行的空行。-->
 <!--5、*必选，本节为二级任务。文档以二级标题为起始标题。-->
 <!--6、易用性写作包括：概述、前置条件、注意事项、常见问题、相关API等。-->
 <!--7、句子采用动宾结构，例如：创建BCC实例。-->
 <!--8、所有的图片符合截图规范；可以多个步骤使用一张截图。-->
 <!--9、不会使用链接时，可参见Markdown速查表，链接包括：页内链接，页间链接，其他产品链接，图片链接-->
 <!--10、最后一步必须要给出能够查看的结果。-->

****概述****

云数据库Redis实例提供多种方式登录访问，如通过账号与密码进行登录，通过密码方式进行登录，如果实例开启内网免密访问功能，则在同VPC的客户端机器上访问时无需提供密码即可登录实例。

****注意事项****

- 无论哪种访问方式，均需要确保客户端机器所在IP地址已加入实例白名单或安全组策略中，具体操作请参考[设置IP白名单](https://cloud.baidu.com/doc/SCS/s/Ejz3v637y)，[设置安全组](https://cloud.baidu.com/doc/SCS/s/tllqcva40)。

****密码登录****

- 登录方式：仅提供实例密码即可登录。如需修改密码请在实例详情页-网络连接-连接密码处进行设置。

-

特点：

但如果所有应用都使用默认用户登录集群，则不能很好地做权限控制。

- redis-cli连接示例：

```
```Shell
```

```
$ redis-cli -h ***.scs.bj.baidubce.com -p 6379
```

```
***.scs.bj.baidubce.com:6379> AUTH password
```

### 账号+密码登录

- 登录方式：需要提供账号和密码进行登录。如需修改账号、密码信息请在实例详情-账号管理页面进行账号的管理。
- 特点：可以为 Redis 实例创建多个账号，可以给不同的用户设置指定的访问权限，以更好地保障数据安全性。
- redis-cli连接示例：



```
$ redis-cli -h *.scs.bj.baidubce.com -p 6379
*.scs.bj.baidubce.com:6379> AUTH username password
```

- 使用限制：用户账号、密码的登录方式对Redis实例版本有限制，目前6.0以下的Redis社区版不支持该访问方式。 **免密登录**
- 登录方式：在实例设置密码时无需提供密码即可登录。如需开启该功能请在实例详情页-网络连接区域，开启内网免密访问开关。

**使用限制：**

- 这种场景下仅支持默认登录到 default 用户。
- 客户端和 Redis 实例必须在同一 VPC 内。
- 需要在控制台开启内网免密码登录的功能。

**特点：**

- 无需密码即可登录，但安全性较低。
- 默认允许同 VPC 内的所有客户端访问实例。
- redis-cli连接示例：

```
$ redis-cli -h *.scs.bj.baidubce.com -p 6379
*.scs.bj.baidubce.com:6379> SET foo bar
OK
```

报错信息	原因及解决方法
(error) ERR invalid password	密码错误，请用正确的格式输入账号、密码。
(error) WRONGPASS invalid username-password pair	密码错误，请用正确的格式输入账号、密码。
(error) ERR AUTH called without any password configured for the default user. Are you sure your configuration is correct?	当前集群尚未设置密码。

[查看连接地址](#)

**概述**

连接云数据库Redis实例有多种方法，如通过数据库管理工具访问，或者在代码中连接访问，或者使用redis-cli客户端访问，无论哪种访问方式都需要先获取数据库的连接地址信息，包括实例地址（称为入口）、端口号等信息，本文为您介绍如何查看云数据库Redis实例的连接地址。 **名词解释**

- 内网：和数据库Redis实例在同一个VPC内的机器，一般指客户购买的BCC实例。
- 外网：即公网(万维网)，即任意一台链接万维网的机器。
- 入口：客户访问数据库时，配置的数据库连接串(可能是域名，IPV4地址)。
- 端口：数据库开放访问端口，以Redis实例为例默认端口是6379。

**注意事项**

- 为确保业务的安全运行，请使用内网IP或者域名连接云数据库Redis，外网IP连接需要用户自己保护好账号、密码信息防止被攻击！
- 云数据库Redis实例创建后，默认允许所有IP通过便于用户连接，为了保证数据安全，请用户在安全管理-白名单、安全组功能中重新设置安全控制策略后再连接访问数据库。

操作步骤

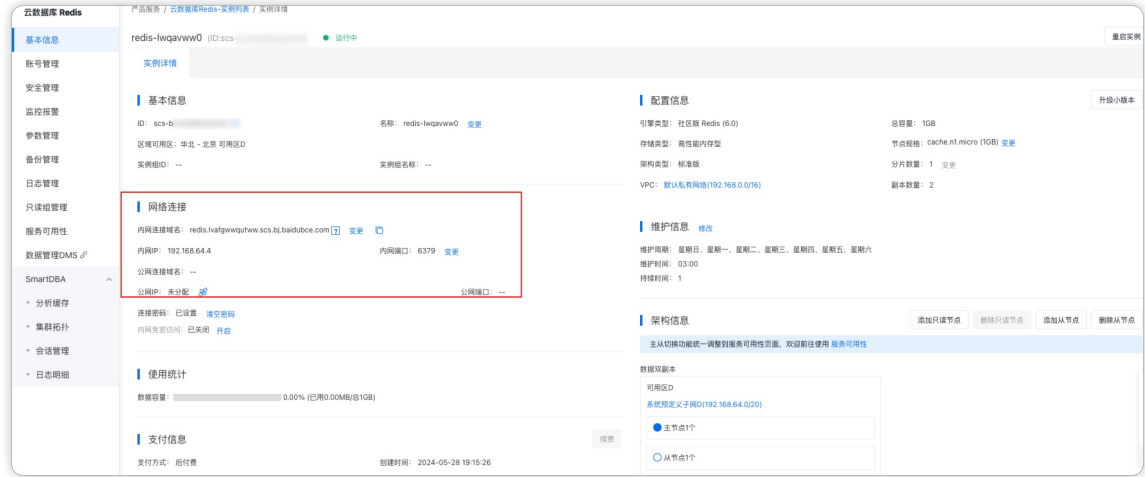
按以下步骤在控制台上查看Redis实例的连接地址：

1. 登录 [云数据库Redis管理控制台](#)。
2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



3. 在实例详情页面网络连接区域，可查看到内网、外网连接地址和端口。
- 如果使用外网连接Redis实例，请先设置实例访问密码，并在同地域[创建公网IP](#)后进行绑定，实例外网访问入口为公网IP:公网端口

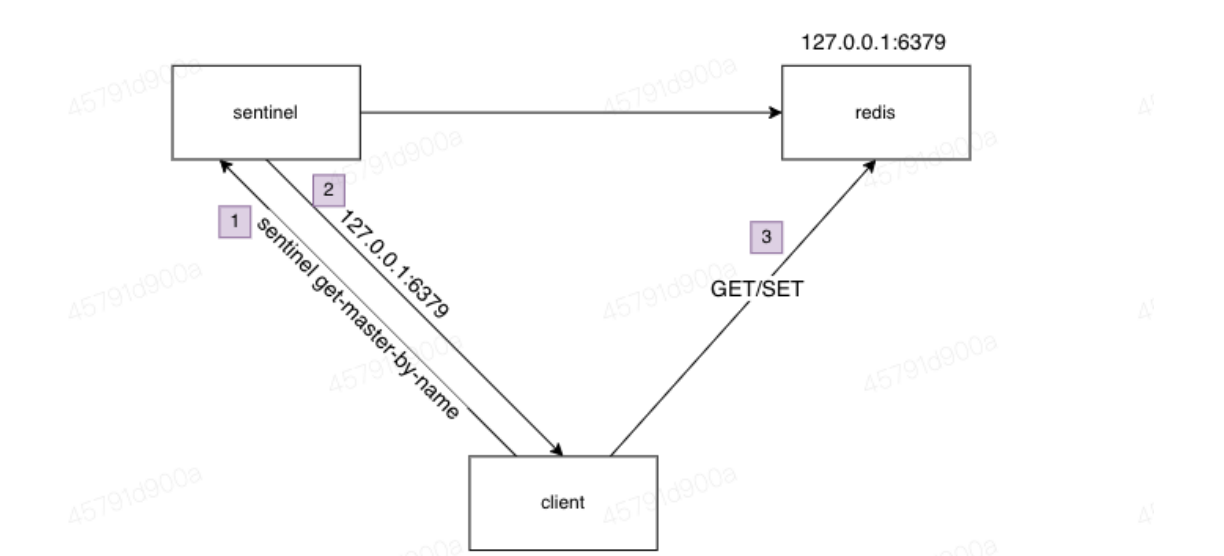
● 如果使用内网连接Redis实例，实例内网访问入口为内网连接域名:内网端口或内网IP:内网端口。



4. 接下来您可以连接Redis实例，更多信息请参考[Redis实例登录方式](#)。

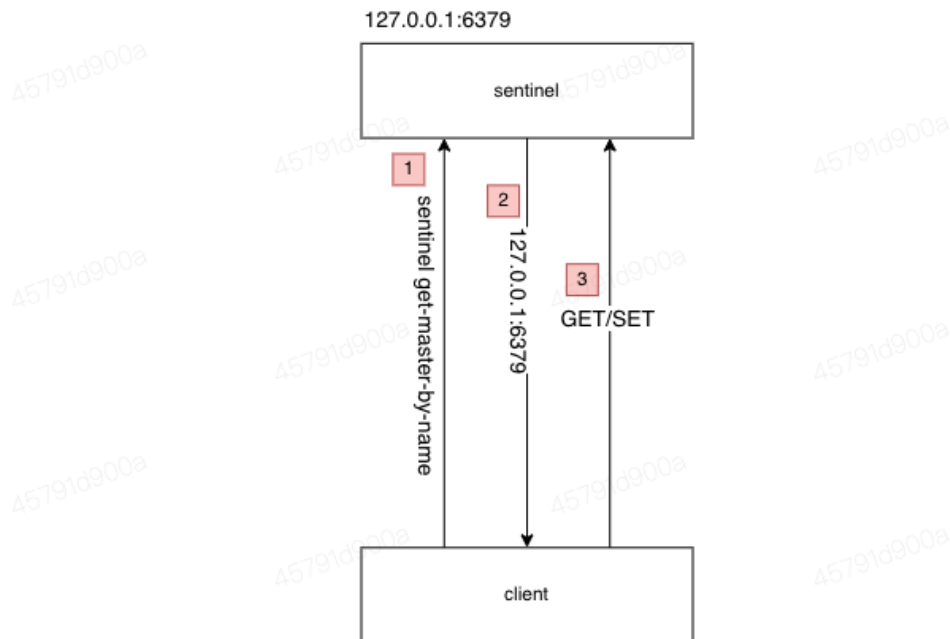
使用 sentinel 模式连接

简介 百度智能云数据库 Redis 支持基于 sentinel 模式连接，本文档将介绍如何使用 sentinel 模式连接到 云数据库 Redis 集群。背景信息 使用 sentinel 模式连接集群时，客户端先从 sentinel 上获取 Redis 节点的地址，然后连接 Redis 节点，流程如下：



上图中，客户端先给 sentinel 节点发送 `sentinel get-master-by-name` 获取到对应 master 的地址，而后连接 master 并执行读写命令。

百度智能云数据库 Redis 单机版和集群版均提供了 sentinel 系列的命令，将自己伪装为一个 sentinel 节点，此前使用 sentinel 模式的客户端，无需变更代码即可接入百度云Redis集群。其原理如下图所示：



上图中 Redis 节点兼任 sentinel 节点的功能，因此在使用 sentinel 模式时接入云数据库 Redis 时，需要明白 sentinel 节点和 Redis 节点实际上是同一个。

**主流客户端示例** 下面介绍各主流客户端使用 sentinel 模式接入的方式。**go-redis** `redis.NewSentinelClient`用来创建一个连接到 sentinel 节点的客户端，可以直接给 sentinel 节点发送命令。

```

sentinel := redis.NewSentinelClient(&redis.Options{
 Addr: "127.0.0.1:6379",
 Password: "sentinel-password",
})
sentinel.GetMasterAddrByName(ctx, "master-name")

```

通常我们需要使用 `redis.NewFailoverClient`，这个客户端会从 sentinel 中或许 redis 节点的信息，然后创建连接到 redis 节点的客户端，用户可以直接发送命令读写 redis。

```

cli := redis.NewFailoverClient(&redis.FailoverOptions{
 MasterName: "master-name",
 SentinelAddrs: []string{"127.0.0.1:26379"},
 Password: "redis-password",
 SentinelPassword: "sentinel-password",
})
client := cli.Get(ctx, "key")

```

Jedis

```
String host = "redis.*****.scs.bj.baidubce.com" ;
int port = 6379;
String password = "*****";
String masterName = "master-name";
JedisClientConfig clientConfig = DefaultJedisClientConfig.builder().password(password).build();
JedisClientConfig sentinelConfig = DefaultJedisClientConfig.builder().password(password).build();
Set<HostAndPort> sentinels = new HashSet<HostAndPort>();
sentinels.add(HostAndPort.from(host + ":" + Integer.toString(port)));
JedisPoolConfig poolConfig = new JedisPoolConfig();
JedisSentinelPool pool = new JedisSentinelPool(masterName, sentinels, poolConfig, clientConfig, sentinelConfig);
Jedis jedis = pool.getResource();
jedis.set("foo", "bar");
```

这里实际是将 redis 实例作为了 sentinel 节点，因此 sentinel 的地址和密码和云数据库 Redis 集群的地址是一样的。注意：

- JedisSentinelPool 有多个构造函数，如果集群有密码，在传入密码的时候，需要同时传入 sentinel-password 和 password。
- 使用 sentinel 模式时，如果出现如下错误，这是正常的，可以将 JedisClientConfig 的 blockingSocketTimeoutMillis 设置为一个很大的值即可。

```
ERROR 2024-05-30 11:31:01.604 redis.clients.jedis.JedisSentinelPool$MasterListener: Lost connection to Sentinel at
127.0.0.1:26379. Sleeping 5000ms and retrying.
redis.clients.jedis.exceptions.JedisConnectionException: java.net.SocketTimeoutException: Read timed out
at redis.clients.jedis.util.RedisInputStream.ensureFill(RedisInputStream.java:251)
at redis.clients.jedis.util.RedisInputStream.readByte(RedisInputStream.java:47)
at redis.clients.jedis.Protocol.process(Protocol.java:135)
at redis.clients.jedis.Protocol.read(Protocol.java:221)
at redis.clients.jedis.Connection.readProtocolWithCheckingBroken(Connection.java:350)
at redis.clients.jedis.Connection.getUnflushedObject(Connection.java:316)
at redis.clients.jedis.JedisPubSubBase.process(JedisPubSubBase.java:115)
at redis.clients.jedis.JedisPubSubBase.proceed(JedisPubSubBase.java:92)
at redis.clients.jedis.Jedis.subscribe(Jedis.java:7941)
at redis.clients.jedis.JedisSentinelPool$MasterListener.run(JedisSentinelPool.java:377)
```

这个错误的原因是，sentinel 客户端启动后会发送一个 subscribe 给 redis 实例，但默认阻塞命令的超时时间的 5 秒，而 subscribe 本就需要阻塞等待后端响应，所以每隔 5 秒就会断开一次。而这个 5 秒是由 JedisClientConfig 的 blockingSocketTimeoutMillis 参数决定的。

## redis-py

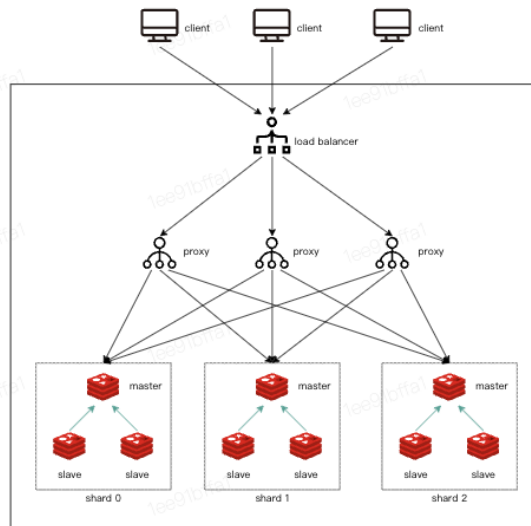
```
host = "redis.*****.scs.bj.baidubce.com"
port = 6379
password = "*****"
sentinel_password = password
这里 password 是 Redis 节点的密码，sentinel_kwargs 中的 password 是 sentinel 的密码
sentinel = Sentinel(host, port, password=password, sentinel_kwargs={'password': sentinel_password})
```

若访问不通时，请自查是否漏掉了 sentinel password，形如下面这样的配置是不对的：

```
sentinel = Sentinel(..., password='123456')
```

## 使用单机模式连接

**简介** 百度智能云数据库 Redis 集群版与原生 Redis 连接方式完全兼容，支持所有主流 Redis 支持的客户端。本文档将介绍如何使用不同编程语言的主流 SDK 连接到 Redis 集群。**背景信息** 云数据库 Redis 采用基于代理的 Redis 集群方案，客户端连接经由负载均衡器连接到 Proxy 上，后续客户端的请求将有 Proxy 负责路由到对应的 Redis 分片上。其结构如下：



集群版中所有的数据被放置在不同的分片上，每个分片由主从节点构成，包含同一份数据的多个副本，配合故障自动切换，可实现服务的高可用，提高数据的安全性。

因此在使用云数据库 Redis 服务时，无论购买的集群版本还是标准版本，均可将其视为单个 Redis 节点，因此可以使用与连接单机版 redis 一样的方法连接云数据库 Redis 集群。

#### 主流客户端示例 redis-cli

```
$./redis-cli -h redis.*****.scs.bj.baidubce.com -p 6379
> AUTH <password> # 使用默认用于 default 认证
```

go-redis 安装客户端：github:<https://github.com/redis/go-redis>

```
$ go get github.com/redis/go-redis/v8
```

#### 连接示例代码：

```
host := "redis.*****.scs.bj.baidubce.com"
port := 6379
password := "*****"
rdb := redis.NewClient(&redis.Options{
 Addr: fmt.Sprintf("%s:%d", host, port),
 Password: password,
 PoolFIFO: true,
})
var ctx = context.Background()
_ = rdb.Set(ctx, "key", "value", 0).Err()
val, _ := rdb.Get(ctx, "key").Result()
if val != "value" {
 panic("")
}
```

如果开启了连接池，需要注意 PoolFIFO 这个参数。这里 PoolFIFO 指定连接池的工作方式，FIFO 意思是 First In First Out，这里 PoolFIFO 默认值为 false，因此在从连接池中拿连接时，默认会拿最近放入的连接。如果 PoolFIFO 保持为默认值，即 false，则会优先使用最近使用过的连接。但为了负载更加均衡，我们希望能轮流地使用所有的连接，因此这里建议将 PoolFIFO 设置为 true。

如有想要使用 Cluster 模式连接云数据库 Redis 集群，下面是一个例子：

```

host := "redis.*****.scs.bj.baidubce.com"
port := 6379
password := "*****"
rdb := redis.NewClusterClient(&redis.ClusterOptions{
 Addrs: []string{fmt.Sprintf("%s:%d", host, port)},
 Password: password,
})
var ctx = context.Background()
_ = rdb.Set(ctx, "key", "value", 0).Err()
val, _ := rdb.Get(ctx, "key").Result()
if val != "value" {
 panic("")
}

```

Jedis 安装 Jedis 客户端 在 pom.xml 中添加如下内容：

```

<dependency>
 <groupId>redis.clients</groupId>
 <artifactId>jedis</artifactId>
 <version>4.3.0</version>
</dependency>

```

## 连接示例代码

```

String host = "redis.*****.scs.bj.baidubce.com" ;
int port = 6379;
String password = "*****";
JedisPoolConfig config = new JedisPoolConfig();
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
try (JedisPool pool = new JedisPool(config, host, port, 3000, password)) {
 Jedis jedis = pool.getResource();
 jedis.set("client", "jedis");
 assert jedis.get("client").equals("jedis");
}
catch (Exception e) {
 e.printStackTrace();
}

```

你可以在 JedisPoolConfig 对象上配置连接池的细节：

```

config.setMaxTotal(10); // 设置最大连接数
config.setTestOnBorrow(false); // 从连接池中拿连接是测试其可用性
config.setTestOnReturn(false); // 放回连接池中是测试其可用性
config.setTestWhileIdle(false); // 连接空闲时测试其可用性
config.setLifo(false); // 设置连接池是否为后进先出

```

这里 setTestOnBorrowsetTestOnReturnsetTestWhileIdle 用于连接保活，连接池可以快速发现并剔除错误连接。如果高频地从连接池获取连接，开启后集群会增加不少探测流量。通常连接不会无故断开，另外用户在执行命令时也会做异常的检测，所以这里建议设置为 false。

setLifo 用于配置连接池获取连接的策略，和前文描述的 golang 客户端中 PoolFIFO 含义类似，这里建议设置为 false，以保证 Proxy 上流量更均衡。lettuce 安装 lettuce 客户端 推荐使用最新版本（目前为 6.3.2.RELEASE 以上版本）连接云数据库 Redis 集群。因为旧版本中存在已知问题，如在 lettuce 5.3.2 之前版本响应解析部分存在bug，在 lettuce 5.2 中新建连接是会发送 COMMAND 命令，这会返回几十KB的响应，大量新建连接会严重降低性能。



```
<dependency>
 <groupId>io.lettuce</groupId>
 <artifactId>lettuce-core</artifactId>
 <version>6.3.2.RELEASE</version>
</dependency>
```

### 连接示例代码

```
String host = "redis.*****.scs.bj.baidubce.com";
int port = 6379;
String password = "*****";
RedisURI uri = RedisURI.builder().withHost(host).withPort(port).withPassword(password).build();
RedisClient client = RedisClient.create(uri);
StatefulRedisConnection<String, String> connection = client.connect();
RedisStringCommands<String, String> sync = connection.sync();
sync.set("client", "lettuce");
String value = sync.get("client");
assert value.equals("lettuce");
```

使用同步接口：下面的示例中使用的是同步接口：

```
RedisStringCommands<String, String> sync = connection.sync();
String value = sync.get("client");
```

在执行 `sync.get` 后，lettuce 会将 GET client 命令放入后台待发送的命令队列中，并构造一个 future 并等待其完成，此时该线程被阻塞。lettuce 的工作线程收到该请求的响应后，会将结果传递给对应的 future，此前被阻塞在 `sync.get("client")` 上线程可继续执行。使用异步接口：使用异步接口，可以在执行命令后不被阻塞地等待，通常是使用回调的方式来处理结果，下面是一个例子：

```
RedisAsyncCommands<String, String> async = connection.async();
async.set("client", "lettuce");
RedisFuture<String> future = async.get("client");
future.thenRun(new Runnable() {
 @Override
 public void run() {
 try {
 System.out.println("Got value: " + future.get());
 } catch (Exception e) {
 e.printStackTrace();
 }
 }
});
```

需要注意的是，这里传递给 `thenRun` 的回调是在 lettuce 工作线程的 `EventLoop` 中执行的，需要保证该回调函数能快速地执行完成，否则将会阻塞事件循环，并影响其他使用该连接的线程。

在 lettuce 中，支持单个链接被多处使用，但部分 Redis 命令（比如 `blpop`）需要独占连接，使用这类命令时需要使用单独的连接。使用连接池 因为 lettuce 默认不使用连接池，而是使用单连接发送请求。如果使用云数据库 Redis 集群版，集群如果有多个 Proxy 且客户端数量较少，则只有少量的连接连到 Proxy 上，这会出现负载均衡的问题，这种场景下可以配置使用连接池，使用方法如下：[https://lettuce.io/core/release/reference/#\\_connection\\_pooling](https://lettuce.io/core/release/reference/#_connection_pooling) redis-py 安装客户端：

```
$ pip install redis
```

### 连接示例代码

```
import redis
host = "redis.*****.scs.bj.baidubce.com"
port = 6379
password = "*****"
pool = redis.BlockingConnectionPool(host=host,
 port=port,
 password=password,
 max_connections=10,
 socket_connect_timeout=0.1,
 socket_timeout=2)
client = redis.Redis(connection_pool=pool)
client.set("client", "redis")
assert client.get("client").decode() == "redis"
```

node-redis 安装客户端：

```
$ npm install redis
```

连接示例代码：

```
import { createClient } from 'redis';
let host = "redis.*****.scs.bj.baidubce.com";
let port = 6379;
let password = "*****";
const client = createClient({
 url: `redis://${host}:${port}`,
 password: password,
});
client.on('error', err => console.log('Redis Client Error', err));
await client.connect();
await client.set('key', 'value');
const value = await client.get('key');
console.log(value) // "value"
await client.disconnect();
```

php-redis 示例代码：

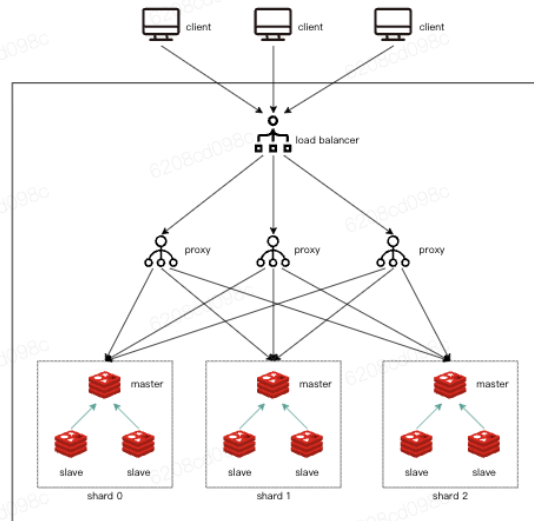
```
<?php
$host = 'redis.*****.scs.bj.baidubce.com';
$port = 8108;
$password = "*****";
$redis = new Redis();
$redis->connect($host, $port);
$redis->auth("12345");
$redis->set("abc", "ABC");
assert($redis->get("abc") == "ABC");
$cluster = new RedisCluster(NULL, Array($host . ':' . $port),
1.5, 1.5, true, $password);
$cluster->set("def", "DEF");
assert($cluster->get("def") == "DEF");
?>
```

以上示例展示了使用不同编程语言的主流 SDK 连接到 Redis 集群的方式，每个客户端均有大量配置项，上文中无法完全罗列，还请用户参考对应 SDK 的官方文档获取更全面的帮助信息。

## 🔗 数据分片规则

在集群版架构中，虽然 Proxy 为客户端屏蔽了集群的实现细节，但目前尚不能做到和使用单机版完全一样，本节我将描述集群

版的一些限制及其原因，并给出一些替代方案。**数据分片规则** 云数据库 Redis 采用基于代理的 Redis 集群方案，客户端连接经由负载均衡器连接到 Proxy 上，后续客户端的请求将有 Proxy 负责路由到对应的 Redis 分片上。其结构如下：



集群版中，将全部的数据空间划分为 16384 个子空间，工程上我们称其为 slot，即一共有 16384 个 slot，这些 slot 分散在多个分片中。 对一个 KV 对，按如下规则将其映射到某个分片上：

```
slot = crc16(key) % 16384
shard = slot_to_shard_mapping[slot]
```

对于每一个集群，都存在一个与之关联的 slot 到 shard 的映射表，这个表由集群管控组件维护。用户执行命令后，Proxy 会拿到这个命令的 key，并计算 slot，而后使用 slot 查表得到对应分片。 客户端无需关注集群的结构，通过 Proxy 即可访问云数据库 Redis 集群，就像是在访问单机版 Redis 一样。但因为数据被分散在了多个分片中，在命令执行上和单机版 Redis 尚存在一些差异。 **如何保证多个 key 落在同一个分片？** 在设计业务方案时候，用户明确某些 key 之间存在关联，希望其放置在同一个分片中。但总不能选择一组 key 保证它们执行 crc16 后得到的结果相同吧。当然不用，此时可以使用 hashtag。

hashtag 就是在 key 中使用 {} 花括号包含一个字符串，Proxy 在对此 key 计算 hash 值的时候，只会使用 {} 中的内容，如此就可以保证不同的 key 计算出的 哈希值完全一样了，下面是个例子：

```
key_000000000000 ---> crc16 ---> % 16384 ---> 5480
key_111111111111 ---> crc16 ---> % 16384 ---> 15248
{key}_000000000000 ---> crc16 ---> % 16384 ---> 12539 # 将 key 包含在 {} 中作为 hashtag
{key}_111111111111 ---> crc16 ---> % 16384 ---> 12539
```

对存在关联的 key 使用相同的 hashtag，可以保证这一组 key 映射到相同的 slot 中，进而保证其落在同一个分片上。

注意事项：

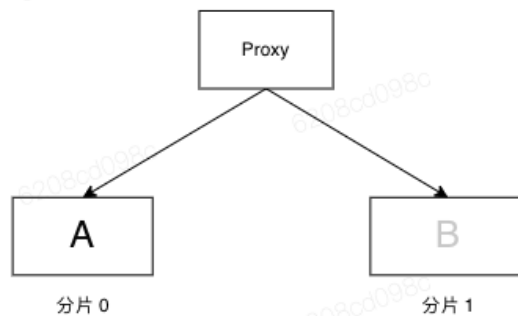
- hashtag 不限定出现的位置，可以在 key 中任何位置。
- 如果key出现了多个 hashtag，只使用第一个出现的 hashtag 来计算哈希值。
- 不要滥用 hashtag，如果有大量的 key 使用了相同的 hashtag，那么某个分片中就会比其他分片多很多 key，那么这会导致各分片的数据量不均衡，进而导致某些 Redis 实例的压力偏大。

**多 key 命令的使用限制** 对于 DEL、MGET、MSET 这类常用的涉及多个 key 的操作，我们在 Proxy 层做了处理，会自动将命令进行拆分，并将拆分后的子命令发送给对应的分片，而后对结果进行汇总。这类命令包含下面这些：MGET、MSET、DEL、UNLINK、EXISTS、TOUCH，使用这些命令时，即便这些命令涉及的 key 落在不同的分片上，用户也无需特殊处理。

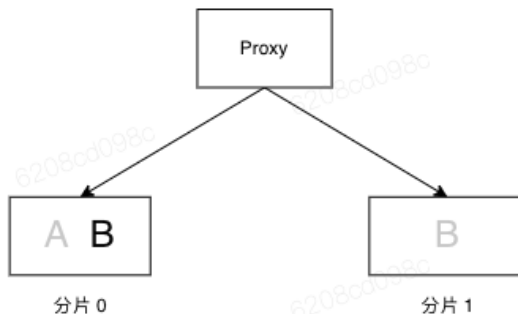
其他部分Redis 命令涉及到同时处理多个 key，比如 RENAME 命令，此命令用于将一个 key 重命名，比如下面的命令将 A 重命名为 B：

## RENAME A B

对于 Redis 集群版而言，假如 A 和 B 这两个 key 落在不同的分片上。想完美地实现 RENAME 同等功能，Proxy 需要把 A 取出来，然后将其写入到 B 中，这将是一个很复杂的操作，而且会引发很多其他的问题。



实际上，Proxy 只是基于 A 来执行转发逻辑，将 RENAMEAB 命令发送给 A 对应的分片。当 RENAME 执行完毕后，在分片 0 中就会多出一个 B key：



当用户尝试读取 B 时候，会将命令发送到分片 1 中，但 B 实际上在 0 分片，而这个 key 则不会被读取到。对于这种涉及到多个 key 的命令，用户需要保证多个 key 落在同一个分片中。

### 🔗 使用 redis-cli 连接 Redis 集群

**简介** redis-cli 是 Redis 自带的命令行工具，可用于连接和管理 Redis 数据库。本文将介绍如何在不同操作系统上安装 redis-cli 并使用它来连接 Redis 集群。

**安装 redis-cli** 接下来分别介绍 Linux、macos、windows 三种环境下安装 redis-cli 的方法。**Linux** 您可以使用 Linux 发行版对应的包管理工具安装 redis，安装 redis 时，会同时安装 redis-cli 工具。

1. #Ubuntu/Debian
2. `sudo apt install redis`
3. #centos
4. `sudo yum install redis`

**macOS** 您可以使用 brew 来安装 redis：

1. `brew install redis`

**Windows** Redis 并保证能在 Windows 平台正常运行，但如果你使用的是 Windows 10 及以上版本，推荐使用 WSL(<https://learn.microsoft.com/en-us/windows/wsl/install>) (Windows Subsystem for Linux) 来安装 redis，使用 WSL 的安装方式与前文中在 Ubuntu 上安装 redis 一致。开发开发者工具，并进入 Linux 子系统，而后输入如下命令：

1. `sudo apt-get install redis`

连接 Redis 集群

安装了 redis-cli 后，可使用以下命令格式连接到 Redis 集群：

```
1. $ redis-cli -h hostname -p port
2. # 可以使用 -h
3.$ redis-cli -h
```

示例

假设 Redis 集群的连接地址为 redis.\*\*\*.scs.bj.baidubce.com，端口号为 6379，密码为，连接命令如下：

```
$ redis-cli -h redis.***.scs.bj.baidubce.com -p 6379
如果有密码需要先认证
redis.***.scs.bj.baidubce.com> AUTH <password>
此后可以执行命名读写 redis
redis.***.scs.bj.baidubce.com> SET A B
OK
```

常见报错处理

如果在连接过程中遇到问题，请参考下表进行故障排查：

报错信息	原因及解决方法
(error) ERR client ip is not in whitelist ，(error) ERR no permission to connect	未设置正确的白名单，或客户端 IP 未添加至白名单。
(error) permission denied for 'SET'	该客户 IP 在白名单中未设置可写权限，因此不能执行 SET 命令。
((error) ERR invalid password ，(error) WRONGPASS invalid username-password pair	密码错误或格式不正确。请确保使用正确的账号和密码格式。

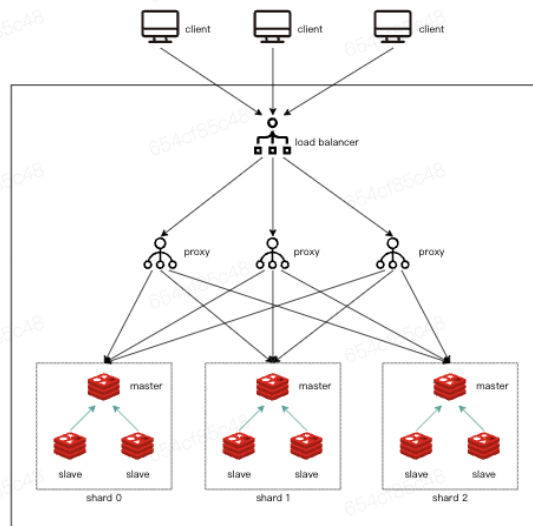
🔗 使用集群模式连接

简介

智能云 Redis 集群版支持使用 Redis clutser 模式连接，部分此前使用 redis cluster 客户端的用户在迁移至云数据库 Redis 后，可无需修改代码，仅将入口修改为云数据库 Redis 集群入口即可。对应新业务，或可以对已有代码做改动的客户，我们推荐使用单机版的连接方式接入云数据库 Redis 集群。

背景信息

云数据库 Redis 采用基于代理的 Redis 集群方案，客户端连接经由负载均衡器连接到 Proxy 上，后续客户端的请求将由 Proxy 负责路由到对应的 Redis 分片上。其结构如下：



为了兼容 redis cluster 连接模式，Proxy 对外将自己伪装为一个单片片的 redis cluster 集群，以此实现对 redis cluster 连接模式的兼容。

**使用限制** 目前集群模式连接不支持公网访问。

**主流客户端示例 go-redis 连接示例代码：**

```
host := "redis.*****.scs.bj.baidubce.com"
port := 6379
password := "*****"
rdb := redis.NewClusterClient(&redis.ClusterOptions{
 Addrs: []string{fmt.Sprintf("%s:%d", host, port)},
 Password: password,
})
var ctx = context.Background()
_ = rdb.Set(ctx, "key", "value", 0).Err()
val, _ := rdb.Get(ctx, "key").Result()
if val != "value" {
 panic("")
}
```

**Jedis** 下面是使用 Redis cluster 连接云数据库 Redis 集群的示例：

```
String host = "redis.*****.scs.bj.baidubce.com";
int port = 6379;
String password = "*****";
HostAndPort hostAndPort = new HostAndPort(host, port);
Set<HostAndPort> nodes = new HashSet<>();
nodes.add(hostAndPort);
try (JedisCluster cluster = new JedisCluster(nodes, "default", password)) {
 cluster.set("client", "jedis");
 assert cluster.get("client").equals("jedis");
} catch (Exception e) {
 e.printStackTrace();
}
```

**lettuce** 使用 Cluster 模式连接实例代码：



```
String host = "redis.*****.scs.bj.baidubce.com" ;
int port = 6379;
String password = "*****";
RedisURI uri = RedisURI.builder().withHost(host).withPort(port).withPassword(password).build();
RedisClusterClient client = RedisClusterClient.create(uri);
StatefulRedisClusterConnection<String, String> connection = client.connect();
RedisStringCommands<String, String> sync = connection.sync();
sync.set("client", "lettuce");
String value = sync.get("client");
assert value.equals("lettuce");
```

redis-py 使用 Cluster 模式连接示例：

```
import redis
host = "redis.*****.scs.bj.baidubce.com"
port = 6379
password = "*****"
client = redis.RedisCluster(host=host, port=port, password=password)
client.set("client", "redis")
assert client.get("client") == b"redis"
assert client.get("client").decode() == "redis"
```

node-redis 使用 Cluster 模式连接示例：

```
import { createCluster } from 'redis';
let host = "redis.*****.scs.bj.baidubce.com";
let port = 6379;
let password = "*****";
const client = createCluster({
 rootNodes: [{
 url: `redis://${host}:${port}`
 }],
 defaults: {
 password: password,
 }
})
client.on('error', err => console.log('Redis Client Error', err));
await client.connect();
await client.set('key', 'value');
const value = await client.get('key');
console.log(value) // "value"
await client.disconnect();
```

php-redis 示例代码：

```
<?php
$host = 'redis.*****.scs.bj.baidubce.com';
$port = 8108;
$password = "*****";
$cluster = new RedisCluster(NULL, Array($host . ':' . $port),
1.5, 1.5, true, $password);
$cluster->set("def", "DEF");
assert($cluster->get("def") == "DEF");
?>
```

以上示例展示了使用不同编程语言的主流 SDK，使用 redis cluster 模式连接到 Redis 集群的方式，但我们更推荐使用单机版的连接方式接入云数据库 Redis 集群。

## 账号管理

🔗 创建账号

### 概述

创建实例时，除系统默认创建的 default 账号外，您可根据业务的需要，通过控制台创建其他的业务账号。

### 账号类型

云数据库 Redis 实例支持两种数据库账号：**默认账号**和**用户账号**。

您可以在控制台管理所有账号和数据库，账号拥有的具体权限请参见“账号权限列表”。

账号类型	说明
普通账号	• 可以通过控制台、API或者SQL语句创建和管理。    • 一个实例可以创建多个普通账号，具体的数量与实例内核有关。    • 需要手动给普通账号授予特定数据库的权限。    • 普通账号不能创建和管理其他账号，也不能断开其他账号的连接。

### 创建账号规则

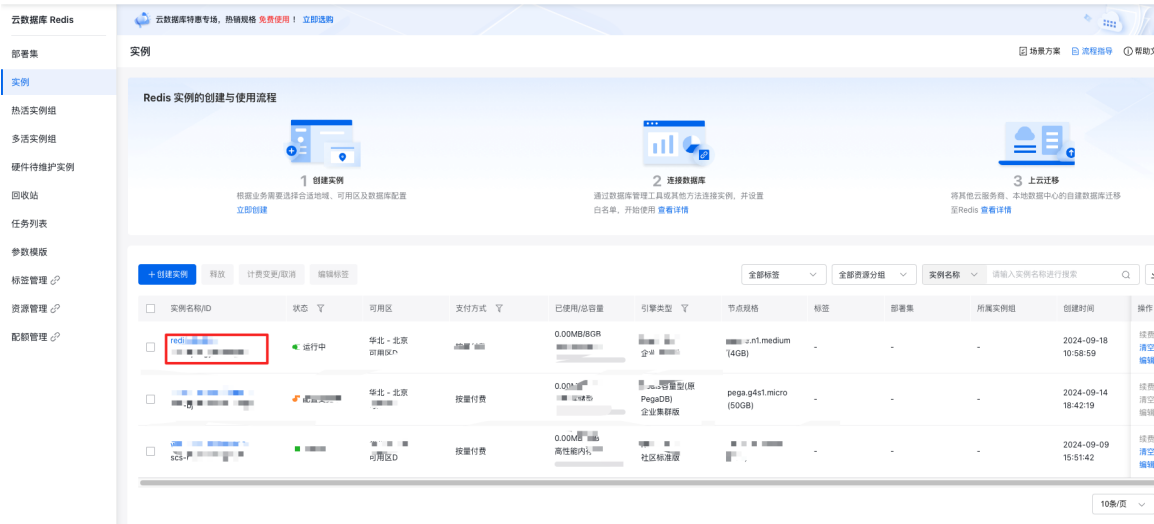
创建账号有一定规则限制，具体参考如下表格：

参数	说明
账号	填写账号名称。要求如下：     • 长度为2~16个字符。    • 以字母开头，以字母或数字结尾。    • 由小写字母、数字或下划线组成。
权限设置	该账号的数据库权限。     • 只读权限。    • 读写权限。
密码	• 长度为8~16个字符。    • 至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^*()_+=。
确认密码	再次输入密码。
备注说明	非必填，备注该账号的相关信息，便于后续账号管理。最多支持256个字符。

### 创建用户账号

1. 登录云数据库 Redis [管理控制台](#)。

- 2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
- 3. 找到目标实例，点击实例名称，进入实例详情页面。



- 4. 在左侧导航栏中单击账号管理。



- 5. 在“账号管理”子页面，点击创建账号按钮，进入创建账号页面，设置参数【参考账号规则】部分。

创建账号

1.每个用户最多可创建50个账号，请进根据需要进行账号管理及分配。

2.default账号是系统默认账号，不允许删除；允许清空密码，但有一定的风险，请谨慎操作。

3.用户账号是您创建的账号，允许删除，不允许清空密码。

\* 账号：

请输入内容

由小写字母、数字、下划线组成、字母开头,字母或数字结尾,最长16个字符

\* 设置权限：

☐只读

☐读写

\* 密码：

密码长度8-16位,至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^\*()\_+~

\* 确认密码：

备注：

请输入内容

确定

取消

6. 点击**确定**按钮，创建账号。

常见问题

1. 账号可以无限创建吗，账号数量上限是多少？

答：账号不能无限创建，最多创建50个账号。

相关API [创建账号](#)

 修改账号权限

概述

您可以根据需要在控制台的账号管理页面**修改用户账号**的权限。

修改账号权限

1. 登录云数据库 Redis [管理控制台](#)。

2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis 实例列表”页面。

3. 找到目标实例，点击**实例名称**，进入实例详情页面。

4. 在左侧导航栏中单击**账号管理**，选择目标账号，单击权限列的**设置**。

71



6. 在设置权限页面中，调整账号权限，可将权限设置为只读或读写。
7. 单击**确定**按钮，提交修改。

设置权限

×

账号：

test\_name

权限：

☐ 只读

☒ 读写

备注：

暂无

确定

取消

相关API [修改账号权限](#)

修改账号密码

概述

- 在使用实例的过程中，如果忘记数据库账号密码，可以通过控制台重新设置密码或清空密码。
- 为了数据安全，建议您定期更换密码。

修改密码规则

参数	规则	样例
密码	• 密码长度范围为8到16位字符。    • 至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^*( )_+=。	Inst12#@

修改账号密码

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中单击**账号管理**。
5. 找到要重置密码的账号，找到**密码**，然后点击**设置**。

产品服务 / 云数据库Redis-实例列表 / 账号管理

● 运行中

账号管理

+ 创建账号

账号	状态	类型	密码	权限	备注	操作
默认账号	● 可用	默认账号	已设置   设置   清空 ?	读写   设置	-	删除
test_name	● 可用	用户账号	已设置   设置	读写   设置	-	删除

6. 在弹出的对话框中，输入新密码并确认后，单击**确定**按钮，提交修改。

修改密码

!

注意：该操作将覆盖原有密码。修改密码将可能导致断连，请谨慎操作

账号：

test\_name

\* 新密码：

密码长度8-16位,至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^\*()\_+-

\* 确认密码：

确定

取消

说明：新的输入密码参考修改密码规则。

相关API [修改账号密码](#)

 修改账号备注

概述

您在使用数据库账号时，可以通过控制台为账号添加/修改**账号备注**，便于后续账号管理。

修改账号备注

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中单击**账号管理**。
5. 在“账号管理”子页面，鼠标移动到账号行，“备注”列会出现手势标记。

账号管理

+ 创建账号

账号	状态	类型	密码	权限	备注	操作
default	● 可用	默认账号	已设置   <a href="#">设置</a>   <a href="#">清空</a> ?	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs1	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs4	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs5	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs6	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs7	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>

6. 出现手势标记后，点击即可出现编辑框，输入新的备注信息，点击**确认**按钮，即可保存。

	状态	类型	密码	权限	备注	操作
	● 可用	默认账号	已设置   <a href="#">设置</a>   <a href="#">清空</a> ?	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	确认取消	<a href="#">删除</a>

相关API

账号列表

删除账号

概述

若您需要对创建的数据库账号进行停用，可以通过控制台删除数据库账号。

注意事项

- 数据库账号删除后不可恢复，请谨慎操作。
- 避免因账号误删除而影响业务的正常使用，请确认即将删除的数据库账号不再对任何应用程序提供使用。

\*\*删除账号\*\*

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中单击**账号管理**。
5. 到需要删除的账号，单击其右侧**操作**列中的**删除**。



● 运行中

账号管理

+ 创建账号

账号	状态	类型	密码	权限	备注	操作
	● 可用	默认账号	已设置   设置   清空 ?	读写   设置	-	删除
test_name	● 可用	用户账号	已设置   设置	读写   设置	-	删除

6. 在弹出的删除账号对话框中，勾选**风险提示**复选框，单击**确定**即可完成账号删除。

删除账号

!

该操作将删除已有账号。删除账号可能导致断连，请谨慎操作。

账号：

test\_name

☒ 我已了解风险并确认执行操作

确定

取消

相关API [删除账号](#)

节点变更

变更节点数量

概述

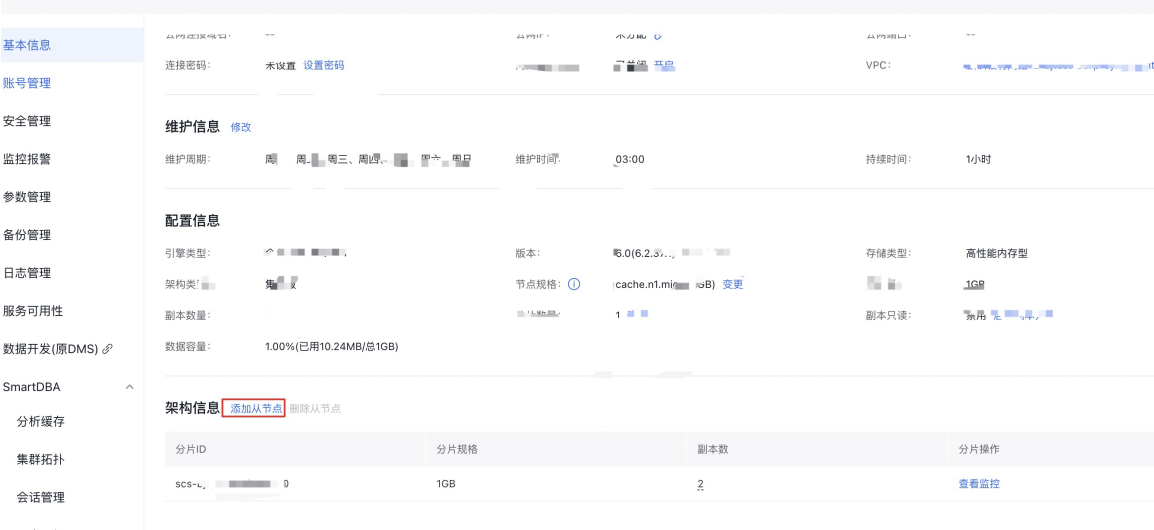
本文为您介绍通过管理控制台变更节点的通用方法，主从节点间具有复制关系。您可以根据业务量变动，灵活变更主备实例的从节点数量。本文介绍如何增加或减少单分片节点数。

前提条件

- 实例状态需为**运行中**。
- 社区版**节点数量最少为1个，**集群版**节点数量最少为2个。

增加从节点

- 登录[云数据库 Redis 管理控制台](#)。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块**添加从节点**按钮。
- 点击**添加从节点**进入“变更配置”页面，在“变更配置”模块，添加从节点信息。

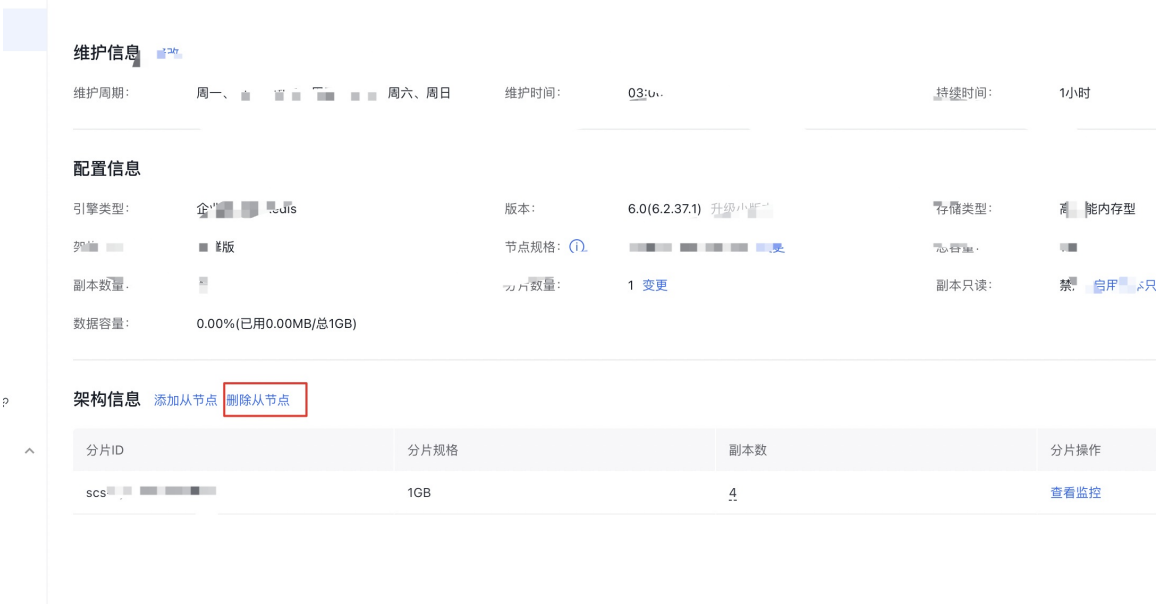


5. 确认好添加从节点的信息，点击下一步按钮，完成后续支付操作，即可完成添加从节点操作。



删除从节点

- 1. 登录云数据库 Redis 管理控制台。
- 2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 3. 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块删除从节点按钮。
- 4. 点击删除从节点进入“变更配置”页面，在“变更配置”模块，删除从节点信息。



5. 确认好删除从节点的信息，点击下一步按钮，完成后续支付操作，即可完成删除从节点操作。



变更只读节点数量

概述

本文为您介绍通过管理控制台变更只读节点的通用方法。

前提条件

- 实例状态需为运行中。
- 只有社区版实例支持。

增加只读节点

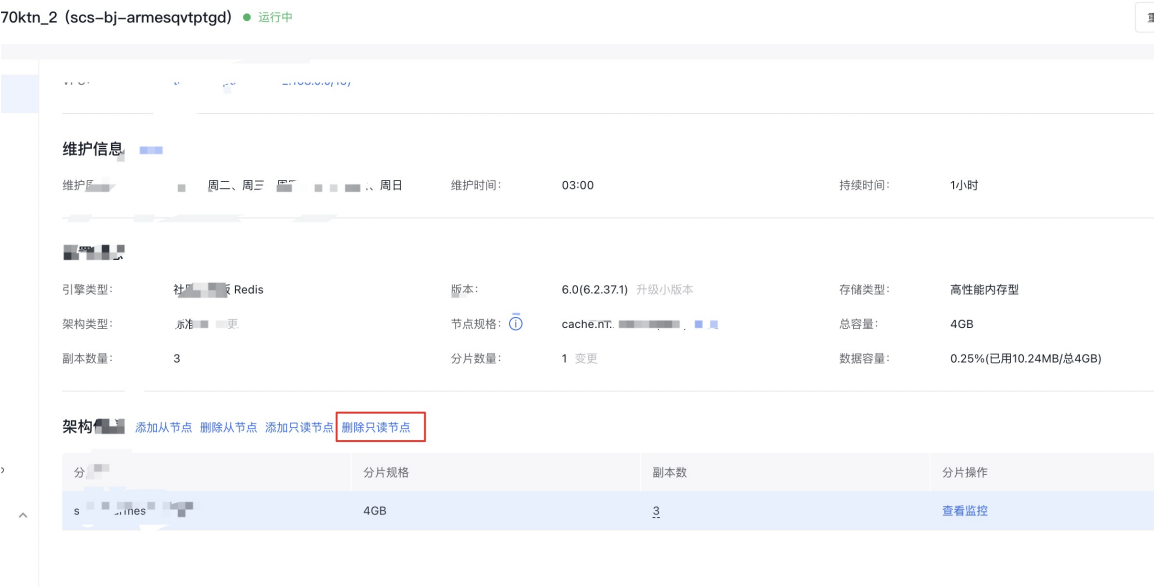
- 登录云数据库 Redis 管理控制台。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块添加只读节点按钮。
- 点击添加只读节点进入“变更配置”页面，在“变更配置”模块，添加只读节点信息。



5. 确认好添加只读节点的信息，点击下一步按钮，完成后续支付操作，即可完成添加只读节点操作。

删除只读节点

- 1. 登录云数据库 Redis 管理控制台。
- 2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 3. 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块删除只读点按钮。
- 4. 点击删除只读节点进入“变更配置”页面，在“变更配置”模块，删除只读节点信息。



5. 确认好删除只读节点的信息，点击下一步按钮，完成后续支付操作，即可完成删除只读节点操作。

安全管理

设置IP白名单

概述

云数据库 Redis 提供自定义 IP 白名单的功能，实现只允许白名单内的 IP 地址访问目标实例。本文为您介绍通过管理控制台对白名单进行设置。

说明：

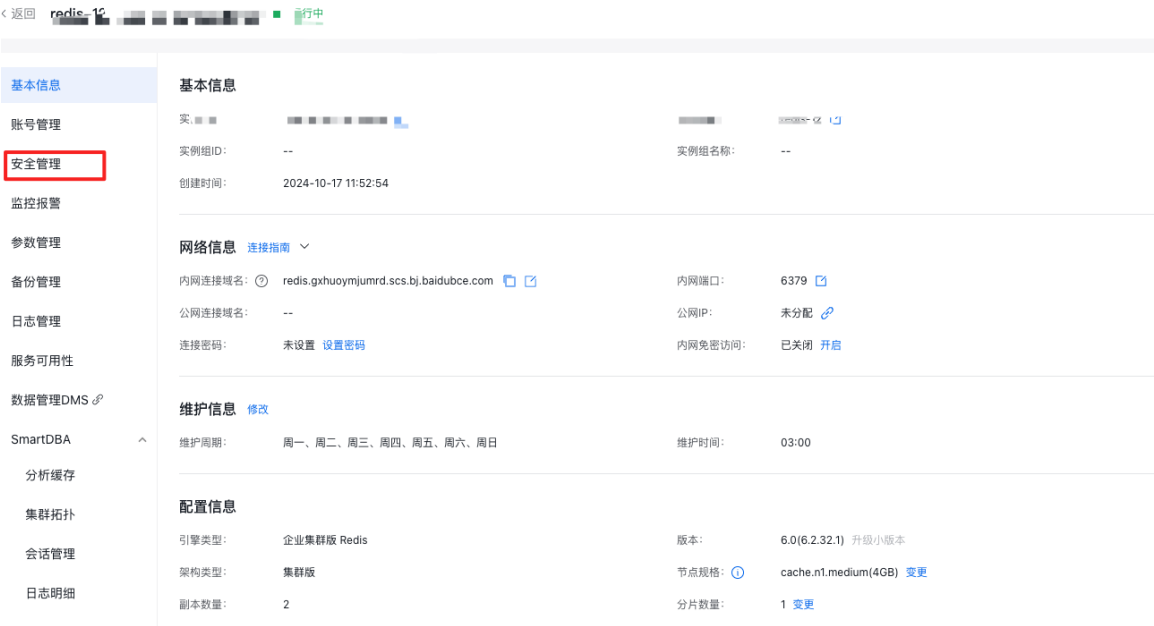
- 支持常规地址: 如 192.168.0.1。
- 支持 CIDR 地址: 如 192.168.1.0/24。
- 多个地址以英文逗号或空格分隔，如 192.168.0.1,192.168.1.0/24。
- 0.0.0.0/0 代表允许所有地址。
- 社区版本不支持该功能。
- \* 代表允许所有地址。

注意事项

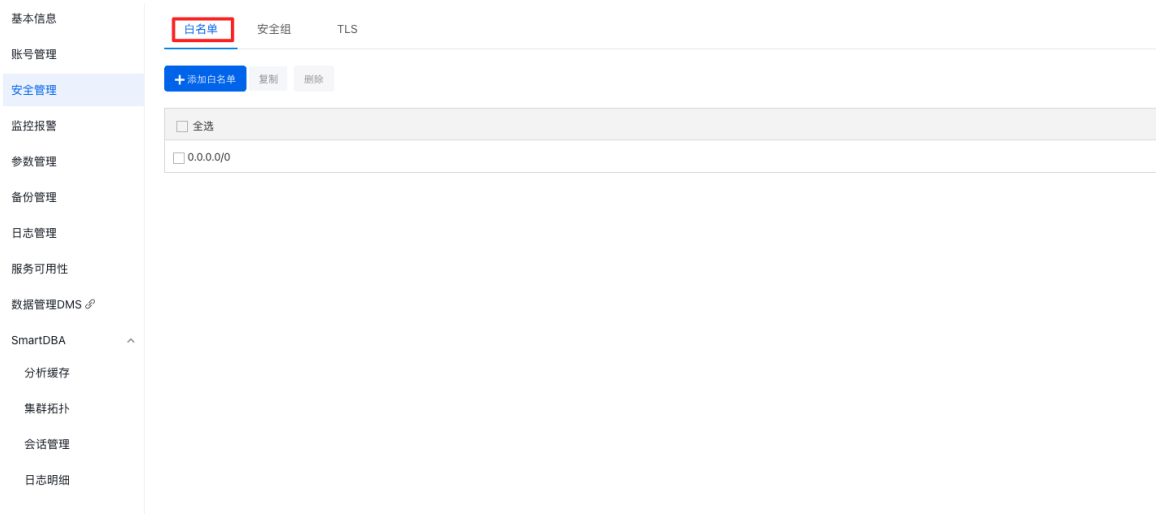
- IP 白名单功能将会逐步升级到安全组功能，提供更丰富的白名单管理功能。
- 对于已经设置了 IP 白名单的用户，可以把 IP 迁移到安全组。
- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。

\*\*操作步骤\*\*

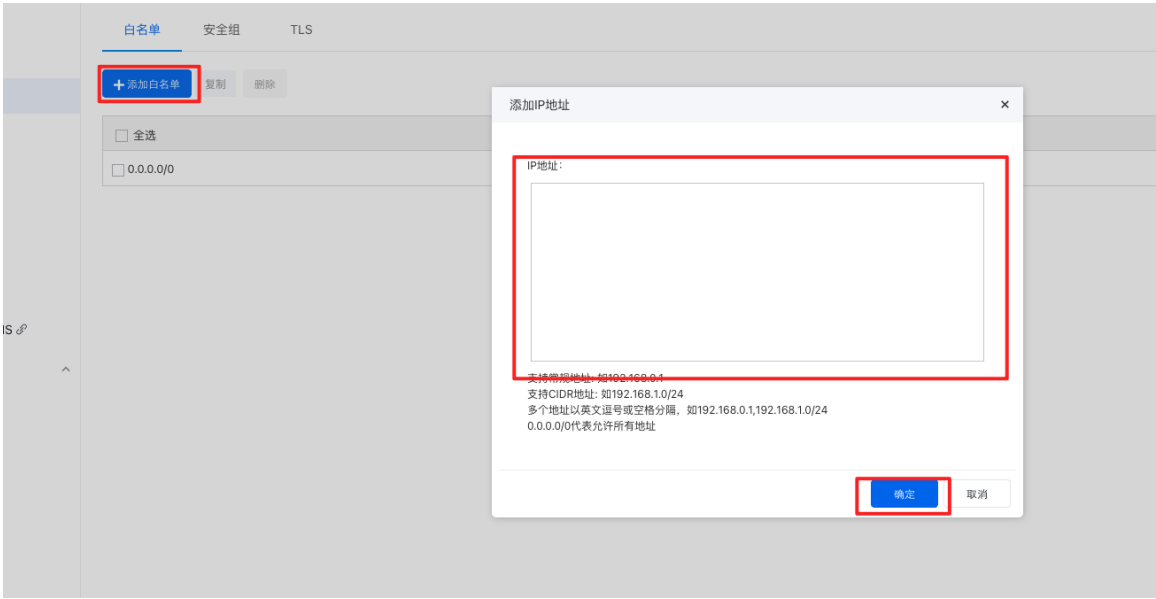
1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。



- 3.在左侧导航栏，单击安全管理，找到IP白名单。



4. 点击添加白名单，输入对应的IP地址，确认后即可添加成功。也可以选中对应白名单进行复制、删除等操作。



相关API [查询IP白名单](#)

[增加IP白名单](#)

[删除IP白名单](#)

🔗 开启免密访问

概述

云数据库 Redis 提供开启免密访问的功能（即清空密码），实现无须密码，即可访问目标实例。本文为您介绍通过管理控制台如何设置免密访问。

说明：

- 若进行清空密码，该操作将清空原有连接密码。允许无密码连接实例将可能存在安全风险，请谨慎操作。

前提条件

- 实例的网络类型为私有网络（VPC）。
- 实例已设置连接密码，方可进行清空操作。
- 标准版和集群版实例均支持此功能。

开启免密访问（清空密码）

1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。在实例详情的网络连接模块，可以看到清空密码操作。

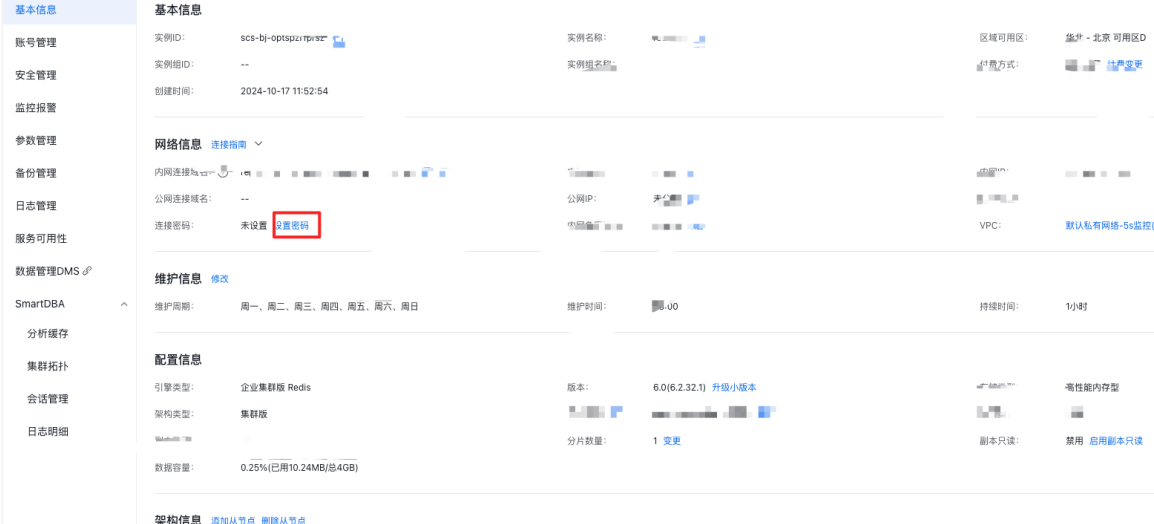


3. 点击 清空密码按钮，在清空密码页面中，勾选我已了解相关操作的风险复选框之后，继续点击【确定】按钮。



关闭免密访问（设置密码）

1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。在实例详情页面，可以看到【设置密码】按钮。



3. 点击 设置密码，根据要求设置连接密码，点击确认按钮后，连接密码即可成功设置。





安全组

**概述** [安全组](#)是在VPC网络内为BCC实例和DCC专属实例、负载均衡实例、云数据库实例中创建的安全防火墙，定义IP + 端口的入站和出站访问策略，从而提高云服务器、负载均衡、云数据库等实例的安全性。

为了方便客户端管理、提高连接的安全性，云数据库 Redis 支持[配置安全组](#)。查看绑定当前实例相关通信协议的出入站规则列表、变更该实例的安全组列表。

注意事项

- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。

操作指南

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 [安全管理](#)，打开安全管理页面。



4. 点击[安全组](#)页签，切换至安全组页面。

产品服务 / 云数据库Redis-实例列表 / 白名单列表



5. 在安全组页面，分为当前应用的安全组规则列表、关联安全组列表两部分。

查看安全组规则列表

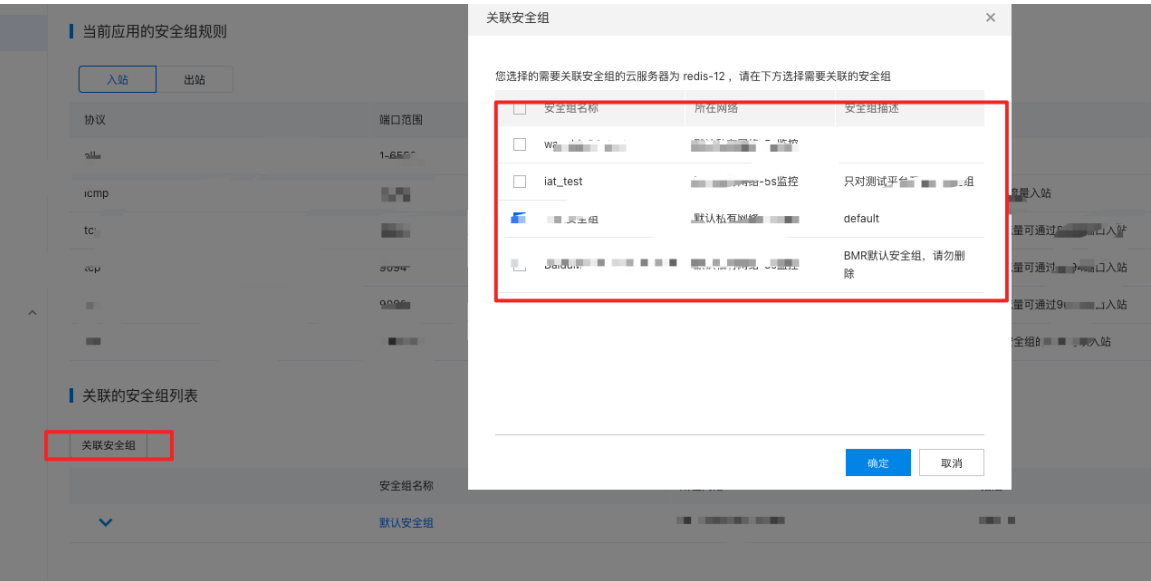
1) 在安全组页面的安全组规则列表区域，点击入站按钮，可以查看当前安全组入站的列表，具体字段包括协议、端口范围、Source、备注，详情如下图所示。



2) 在安全组页面的安全组规则列表区域，点击出站按钮，可以查看当前安全组出站的列表，具体字段包括协议、端口范围、Target、备注，详情如下图所示。

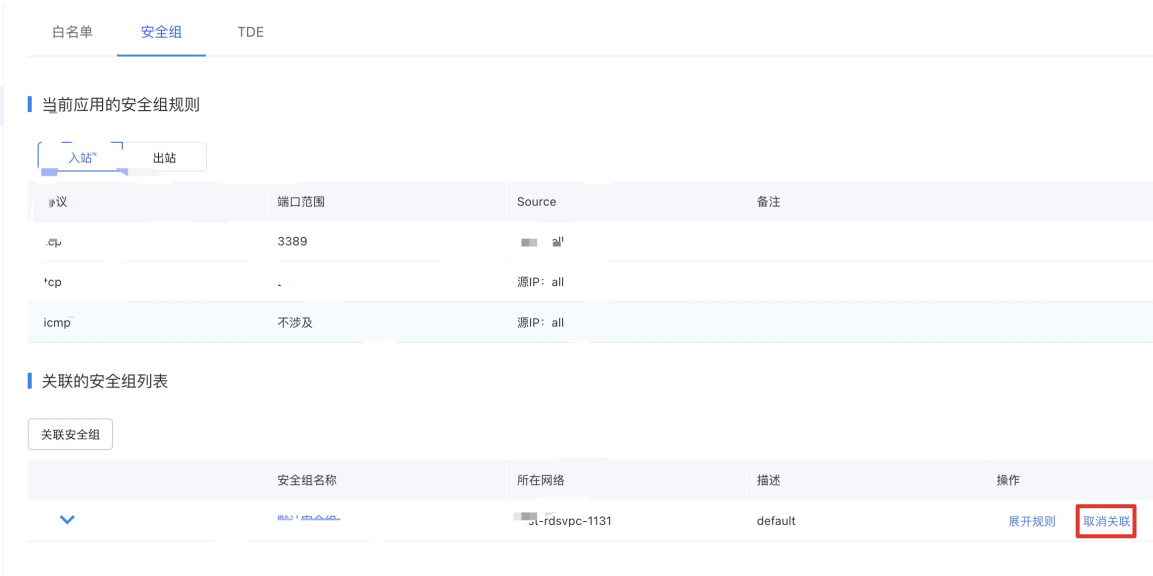
关联安全组列表

- 1) 在关联的安全组列表区域下方点击关联安全组按钮，弹出关联安全组对话框。
- 2) 在列表中、用户可以勾选多个安全组，点击确定按钮，即可完成绑定。



解绑已关联的安全组

- 1) 在关联的安全组列表区域下方选择需要解绑的关联安全组，点击取消关联后确认，即可解绑该安全组。



TLS加密访问

概述

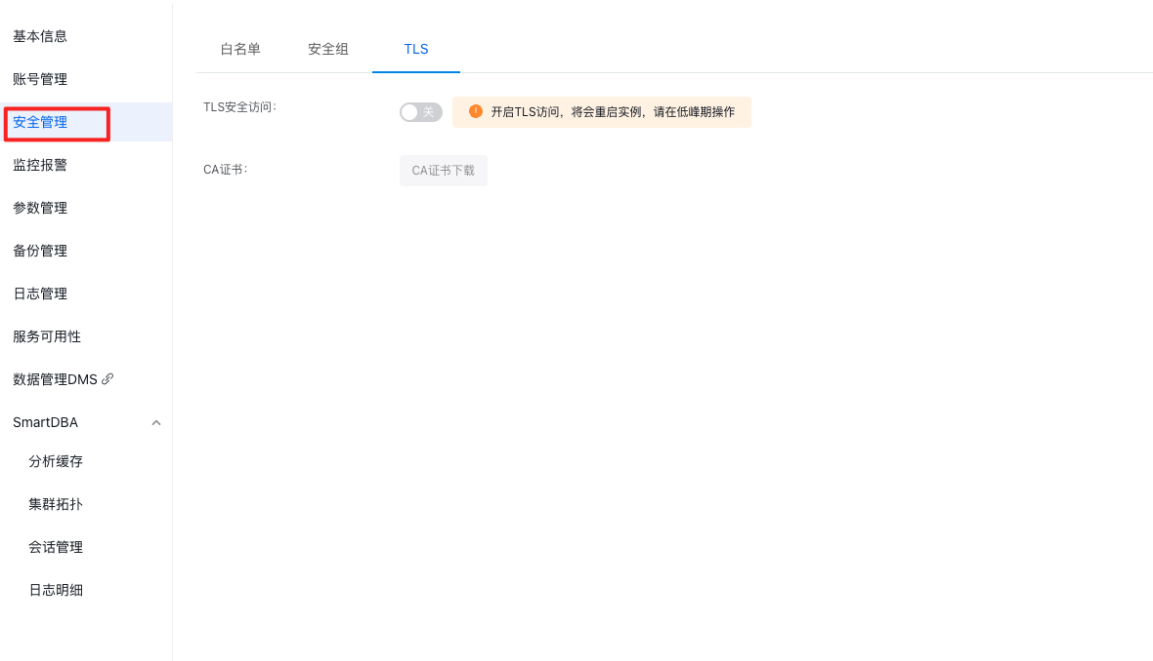
- 您可以通过控制台打开或关闭**TLS 安全访问**，还可以更新CA证书和下载CA证书。开启TLS后可以增加服务器安全访问机制。

注意事项

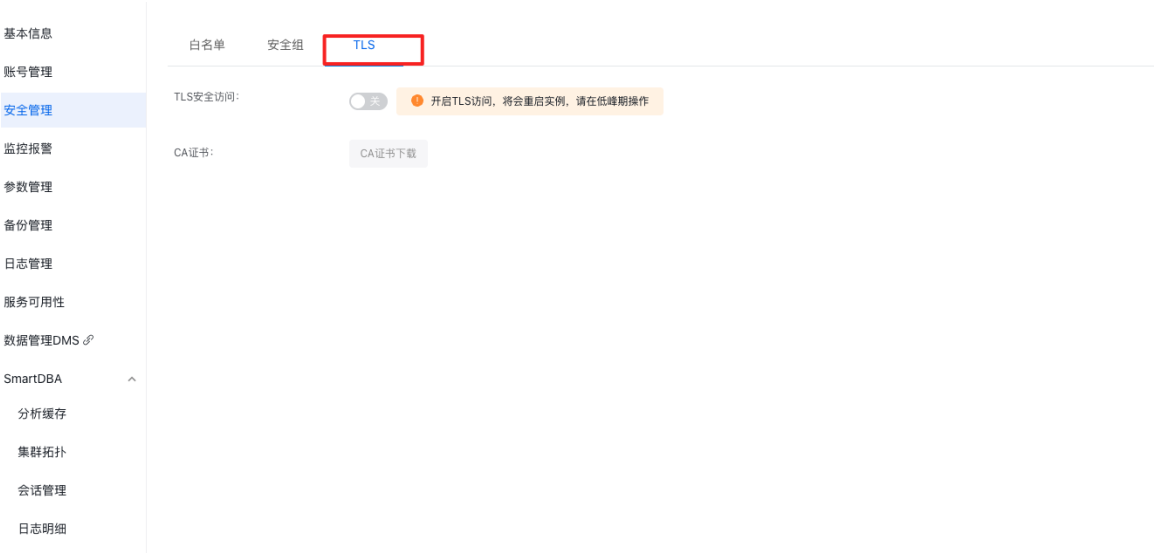
- TLS 仅支持**企业版本**。
- CA证书更新、CA证书下载必须是在**TLS打开状态**，方可操作。
- TLS 打开和关闭实例状态都将显示**配置变更中**。

开启 TLS 安全访问

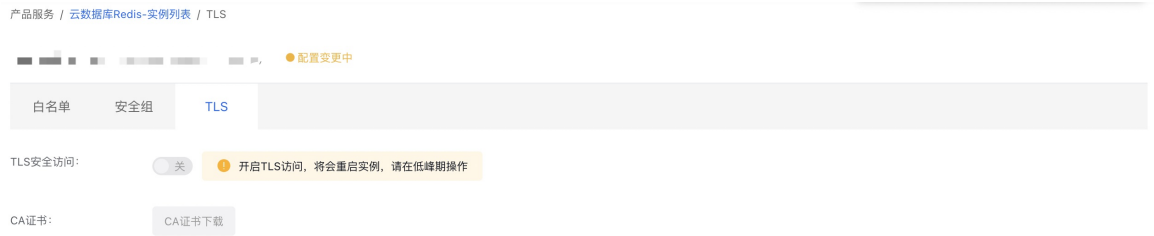
- 登录**云数据库 Redis** 管理控制台，进入实例列表页面。
- 点击目标**实例名称**，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击 **安全管理**，打开安全管理页面。



4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



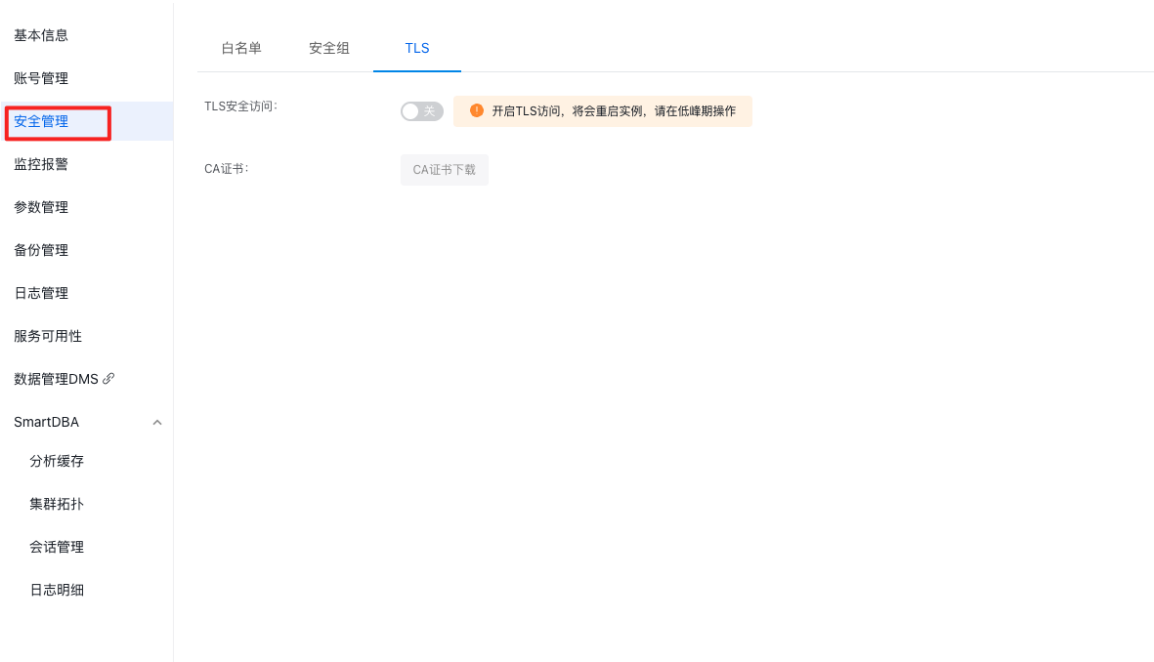
5. TLS安全访问默认是关闭状态，并且CA证书无法下载，点击右侧开/关按钮，实例状态显示为配置变更中。



6. 开通成功后，TLS 安全访问状态显示为开，并且会新增两行保护地址及证书有效期。

更新 CA 证书

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。



4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。

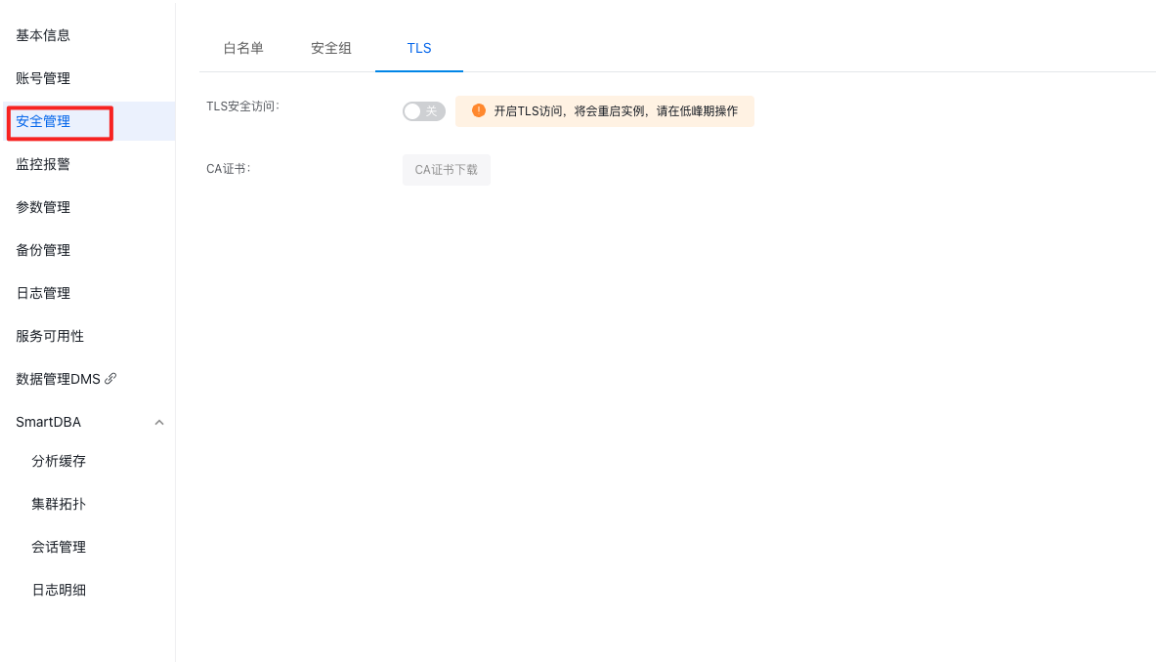


5. 点击证书右侧更新证书按钮，弹出更新TSL证书有效期，点击确定按钮。



下载 CA 证书

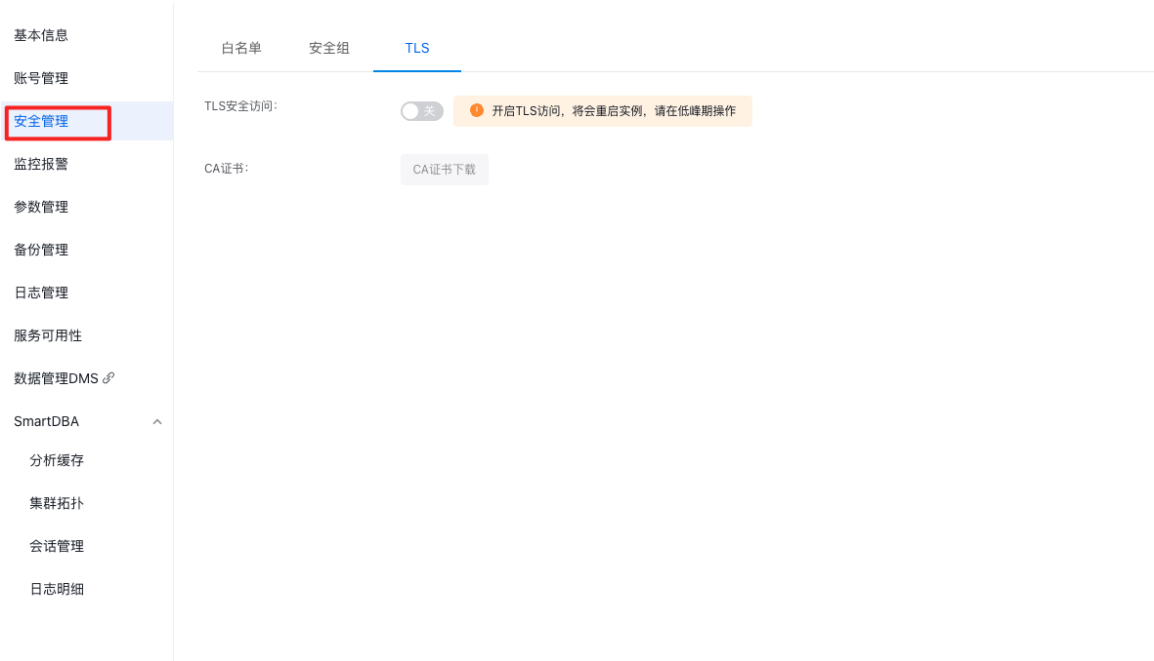
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。



- 4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。
- 5. 点击右侧下载CA证书按钮，将CA证书下载到本地。

关闭 TLS 安全访问

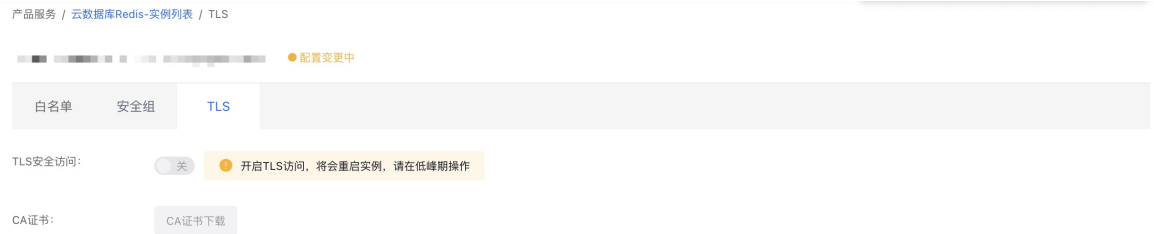
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。



4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



5. 点击TLS安全访问右侧开/关按钮，实例状态显示为配置变更中。



6. 关闭TLS安全访问后，实例状态显示为运行中。



开启TDE加密

概述

本文为您介绍开启Redis TDE透明数据加密保护的方法。

注意事项

- 仅Redis(6.0)标准版支持开启TDE加密，热活及多活实例组所属实例不支持操作TDE加密，集群状态非运行中时不允许操作TDE加密。
- TDE开启后不可关闭，功能开启后不能使用可用区迁移，无法使用大Key分析，无法使用热活架构和异地多活架构，不能使用DTS执行数据迁移和同步等操作。

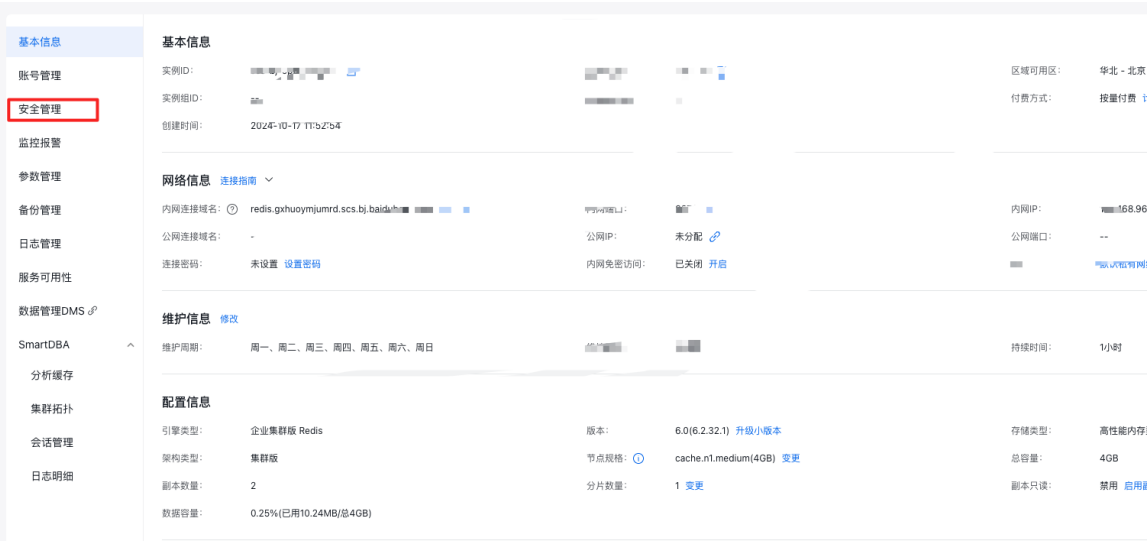
操作步骤

按以下步骤在控制台上开启TDE加密：

1. 登录 云数据库Redis管理控制台，左上角选择地域信息进行地域切换。
2. 进入指定地域的实例列表页，点击需要开启的实例名称/ID。



3. 进入实例详情页面，左侧标签页导航栏中选择安全管理，打开安全管理页面。



4. 默认选中白名单页签，点击TDE页签，切换至TDE子页面。



5. TDE加密默认是关闭状态，点击右侧开/关按钮，打开TDE加密功能，开启过程中实例状态为配置变更中状态，开启成功后TDE状态变更为开。





数据迁移

Redis数据迁移方案概览

概览

借助数据传输服务DTS（Data Transmission Service），云数据库 Redis 提供了多种数据迁移方案，包括自建 Redis 数据库迁移上云、云数据库 Redis 实例间的数据迁移等，满足多种业务场景下Redis数据迁移的需求。

**DTS介绍** DTS是百度智能云提供的实时数据流服务，集数据迁移、订阅、同步于一体，为您提供稳定安全的传输链路。DTS支持全量数据迁移和增量数据迁移，可实现不停服迁移，降低对业务的影响。

源端数据类型	目标数据类型	迁移类型
- 自建 Redis 版本：2.8x、3.x、4.x、5.x、6.x、7.x   - 云数据库 Redis	- 自建 Redis 版本：2.8x、3.x、4.x、5.x、6.x、7.x   - 云数据库 Redis	全量迁移 增量同步

迁移方案

场景	工具	配置案例
从自建数据迁移到云数据库 Redis	DTS Redis-Cli	<a href="#">自建 Redis 迁移上云</a> <a href="#">使用 AOF 文件进行迁移</a>
云数据库 Redis 间迁移	DTS	<a href="#">云数据库 Redis 间迁移</a>

使用 DTS 迁移自建 Redis

云数据库 Redis 支持用户使用百度智能云数据传输服务 DTS 将自建 Redis 数据库实例的数据迁移上云。详细操作请参考

- [自建 Redis 标准版迁移到云数据 Redis。](#)
- [自建 Redis 集群版迁移到云数据 Redis。](#)

使用 AOF 文件进行迁移

概述

redis-cli 是原生 Redis 的命令行工具，您可以使用 redis-cli 将自建 Redis 中已保存至AOF文件的数据迁移到云数据库 Redis 版。本方案具有灵活、高效与自动化等特点，可保障数据的一致性与完整性，但本方案不支持实时同步，仅支持迁移离线数据。

建议停机迁移数据，或在迁移过程中不写入新的数据。

redis-cli不是在线迁移工具，而仅是一个简单的读取、发送工具，无法保证复杂的在线迁移场景，如需进行在线实时迁移，请参见 [使用 DTS 迁移自建 Redis](#)

### 前提条件

- 将客户端所在设备的IP地址添加到云数据库 Redis 版白名单中，更多信息请参见[设置IP白名单](#)。
- 自建Redis实例已开启AOF功能。若实例未启用，请执行CONFIG SET appendonly yes命令开启AOF功能。
- 自建Redis实例已关闭混合持久化。若自建实例为Redis 5.0及以上版本，请执行CONFIG SET aof-use-rdb-preamble no命令关闭混合持久化。

### 操作步骤

- 连接自建Redis实例，手动触发AOF文件重写，保存最新的AOF文件。

```
命令：redis-cli -h <自建Redis实例IP地址> -p <端口号> BGREWRITEAOF
```

```
返回：Background append only file rewriting started
```

- 通过AOF文件将数据导入到新的云数据库 Redis 版实例中，此处以AOF文件appendonly.aof为例。

```
命令：redis-cli -h <云数据库Redis版的IP地址> -p <端口号> -a <实例密码> --pipe < appendonly.aof
```

```
返回：
All data transferred. Waiting for the last reply...
Last reply received from server.
errors: 0, replies: 90
```

当看到类似上述信息，表示迁移成功。

**后续步骤** 如果自建Redis实例不需要一直开启AOF，可在导入完成后通过以下命令关闭。

```
命令：redis-cli -h <自建Redis实例IP地址> -p <自建Redis实例端口号> CONFIG SET appendonly no
```

## 变更实例

### 变更配置

#### 概述

本文为您介绍通过云数据库 Redis 管理控制台**变更配置**的基本方法。

#### 注意事项

- 标准版实例**仅支持变更实例规格，不支持分片数量变配。
- 集群版实例**支持变更实例规格及分片数量（实例副本数量小于2时不支持变更实例规格）。

### 操作步骤

变更节点规格

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在“云数据库 Redis-实例列表”中，选择需要变更配置的实例，点击实例名称，进入实例详情。
3. 在实例详情页面的【配置信息】模块，点击节点规格后的变更，进入“变更配置”页面。

基本信息

实例ID: redis-1n7zds7

实例组ID: --

创建时间: 2024-10-17 11:52:54

网络信息

内网连接域名: redis.qw...

公网连接域名: --

连接密码: 已设置 清空密码

维护信息

维护周期: 周一、周二、周三、周四、周五、周六、周日

配置信息

引擎类型: 企业集群版 Redis

架构类型: 集群版

副本数量: 3

数据容量: 0.25%(已用10.24MB/总4GB)

版本: 6.0(6.2.32.1) 升级小版本

节点规格: cache.n1.medium(4GB) 变更

分片数量: 1 变更

存储类型: 高性能内存型

总容量: 4GB

副本只读: 禁用 启用副本只读

架构信息

分片ID

分片规格

副本数

分片操作

区域可用区: 华北 - 北京 可用区D

付费方式: 按量付费 计费变更

内网IP: 192.168.96.240

公网端口: --

VPC: 默认私有网络-56监控(192.168.0.0/)

持续时间: 1小时

4. 实例规格确认完成，然后在右侧订单栏点击下一步按钮，进行订单付费。完成订单付费后，实例配置将进行变更。待实例变配完成，实例状态更新为“运行中”，您可在实例详情查看实例的最新配置。

实例规格族

规格代码

CPU核

内存/GB

最大连接数

最大内网带宽/MB/s

☐ 独享型

cache.n1.micro

2

1

10000

10

☐ 独享型

cache.s1.micro

2

1

10000

10

☒ 独享型

cache.n1.small

2

2

10000

16

☐ 独享型

cache.s1.small

2

2

10000

16

☐ 独享型

cache.n1.medium

2

4

60000

24

☐ 独享型

cache.n1.c4.medium

4

4

60000

24

☐ 独享型

cache.s1.medium

2

4

60000

24

☐ 独享型

cache.n1.large

2

8

60000

24

☐ 独享型

cache.n1.c4.large

4

8

60000

24

☐ 独享型

cache.s1.large

2

8

60000

24

☐ 独享型

cache.n1.xlarge

2

16

60000

32

☐ 独享型

cache.n1.c4.xlarge

4

16

60000

32

☐ 独享型

cache.n1.3large

2

24

60000

40

☐ 独享型

cache.n1.2xlarge

2

32

60000

48

已选择: cache.n1.small (2GB)

分片数量: 1片

副本数量: 3

架构预览: 总容量 2GB | 单分片容量 2GB | 分片数量 1个 | 代理节点数量 2个  
单分片包含: 主节点 1个 + 从节点 2个

性能预览: 最大连接数 1万 | 最大内网带宽 16MB/s | CPU处理能力 2核 | QPS参考 60000万  
副本只读启用时，性能预览会相应调整

变更时间: ☒立即执行 ☐维护时间内

所选配置

地域: 华北 - 北京

变更配置: 节点规格: cache.n1

总容量: 2GB

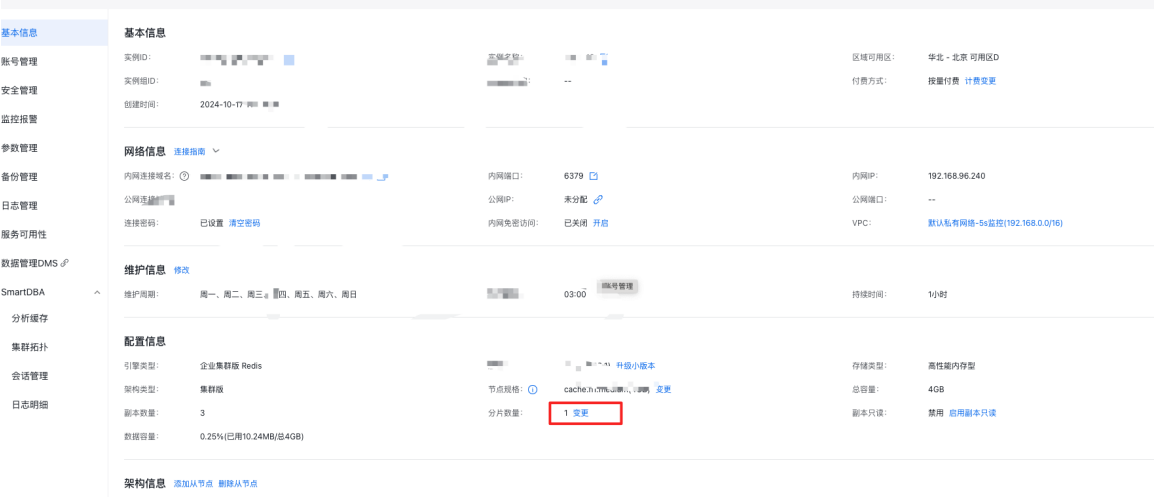
价格: ¥0.009723/分  
¥14.00/天 ¥420.0

变更配置后历史备份将失效。

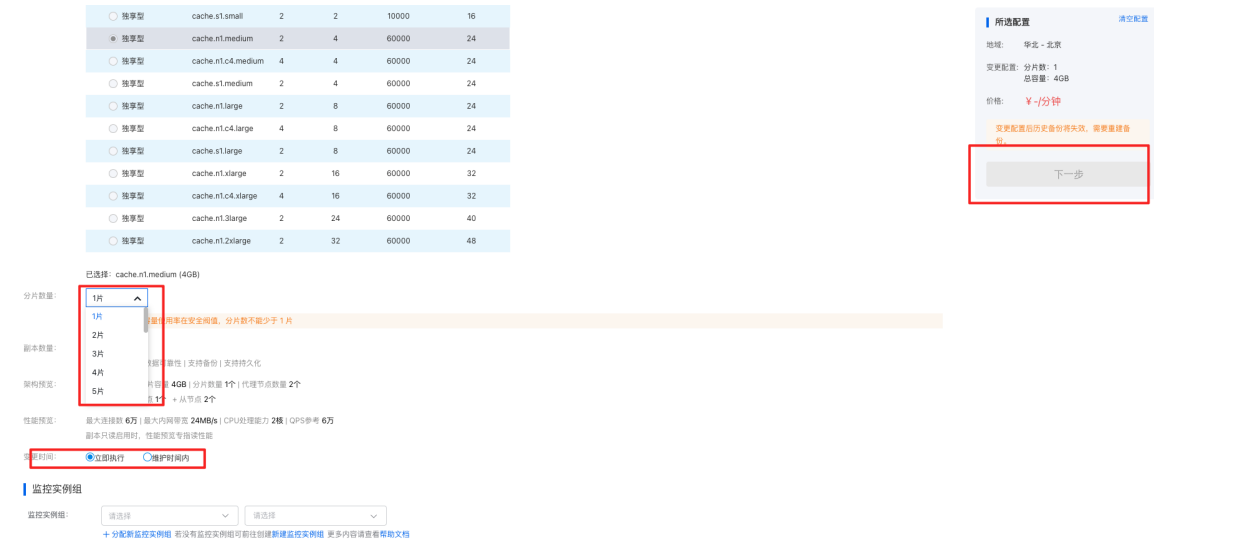
下一步

变更分片数量

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在“云数据库 Redis-实例列表”中，选择需要变更配置的实例，点击实例名称，进入实例详情。
3. 在实例详情页面的【配置信息】模块，点击分片数量后的变更，进入“变更配置”页面。



4. 分片数确认完成，然后在右侧订单栏点击下一步按钮，进行订单付费。完成订单付费后，实例配置将进行变更。待实例变配完成，实例状态更新为“运行中”，您可在实例详情查看实例的最新配置。



说明：仅集群版支持变更分片数量。

常见问题

- 资源配置发生更改后，资费是如何变更的？  
答：资源配置发生更改后，需要根据用户需要重新购买服务。

相关API

变更实例

主从切换

概述

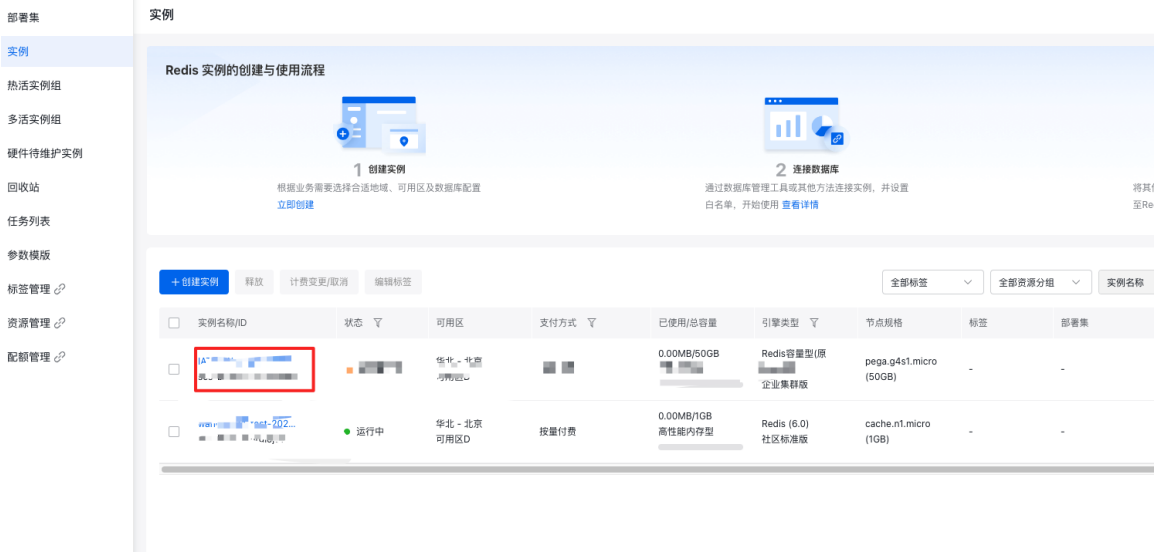
云数据库 Redis 提供手动主从切换的功能。本文为您介绍通过管理控制台进行主从切换的通用方法。

注意事项

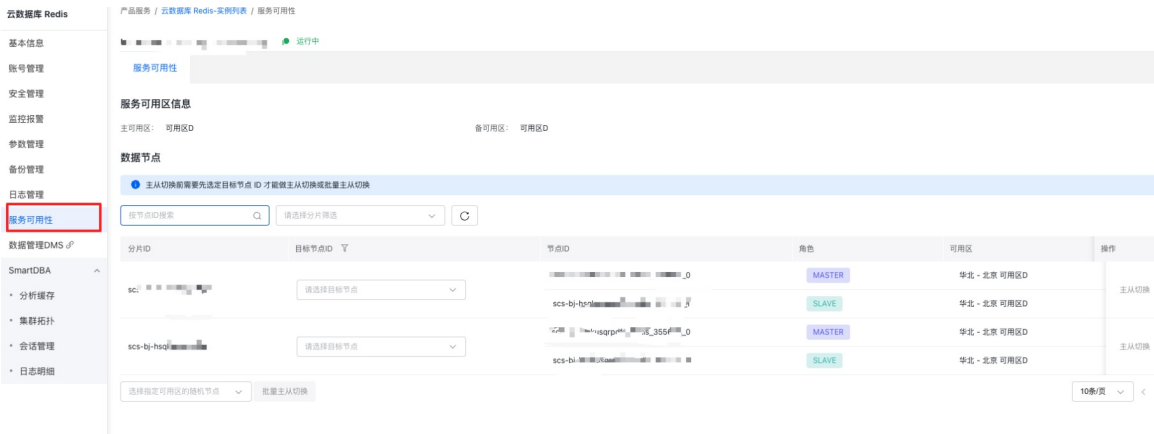
- 主节点角色将移交到其他可用区的从节点。如果所有节点都位于同一个可用区，则不支持该操作。
- 主从切换过程中可能会有1-2次闪断，请选择业务低峰期进行。

主从切换

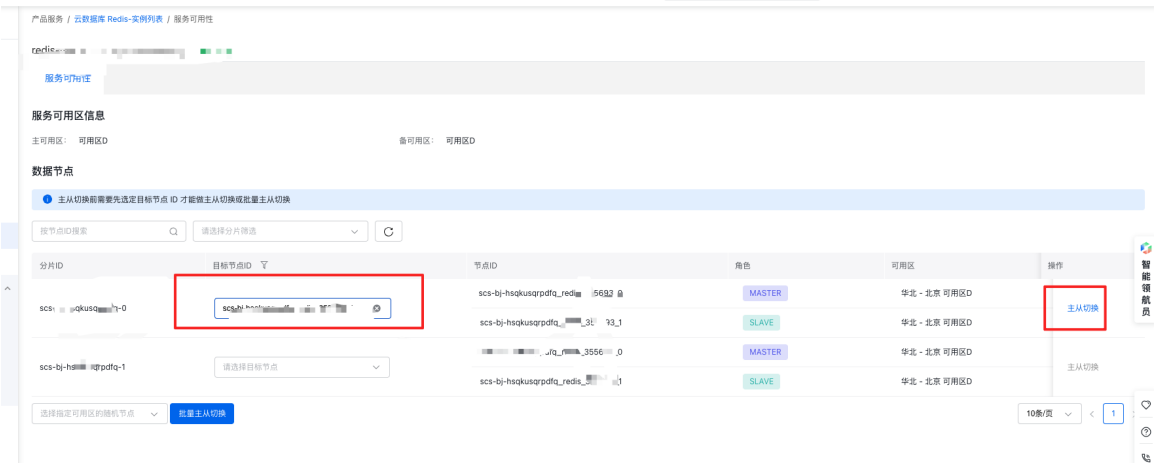
- 1. 登录云数据库 Redis 管理控制台。
- 2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



- 3. 在实例详情页的标签页导航栏中点击服务可用性。



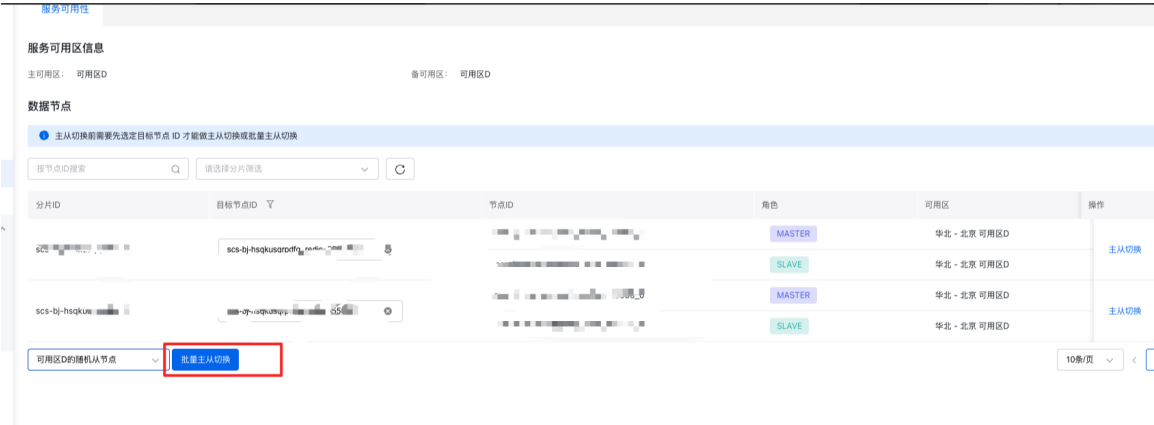
- 4. 选取目标节点后，点击主从切换。



- 5. 勾选我已了解风险并确认执行操作后点击确认。



6.也可选择批量主从切换，快捷切换到可用区随机从节点。



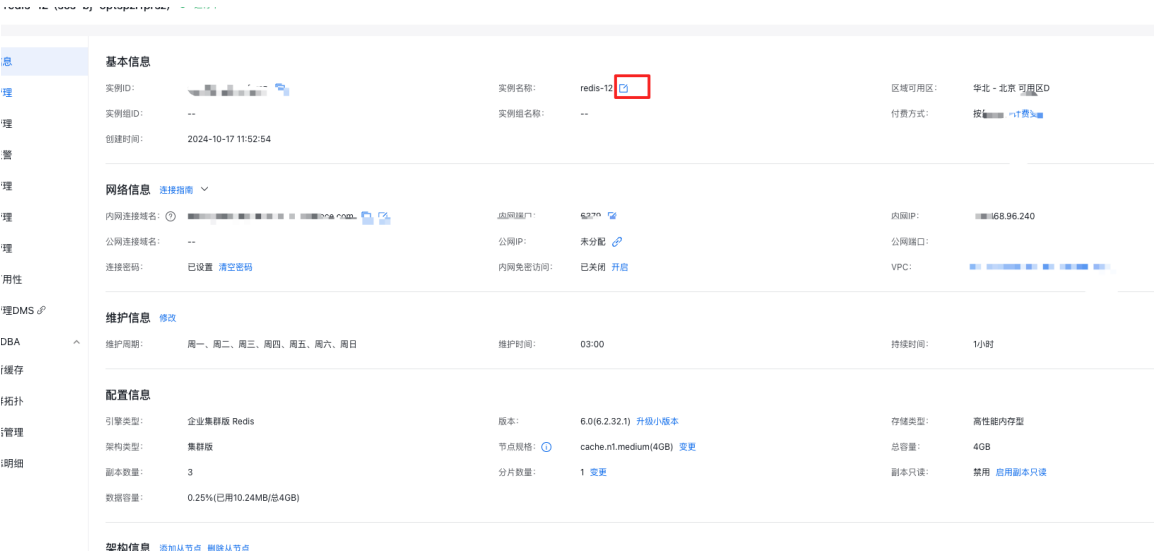
修改实例名称

概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例名称的基本方法。

操作步骤

- 1. 登录云数据库 Redis 管理控制台。
- 2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 3. 点击变更，输入变更后名称。



- 4. 点击确认按钮，用户可以完成修改实例名称的操作。



相关 API [修改实例名称](#)

[🔗 修改实例域名](#)

[🔗 设置公网访问](#)

概述

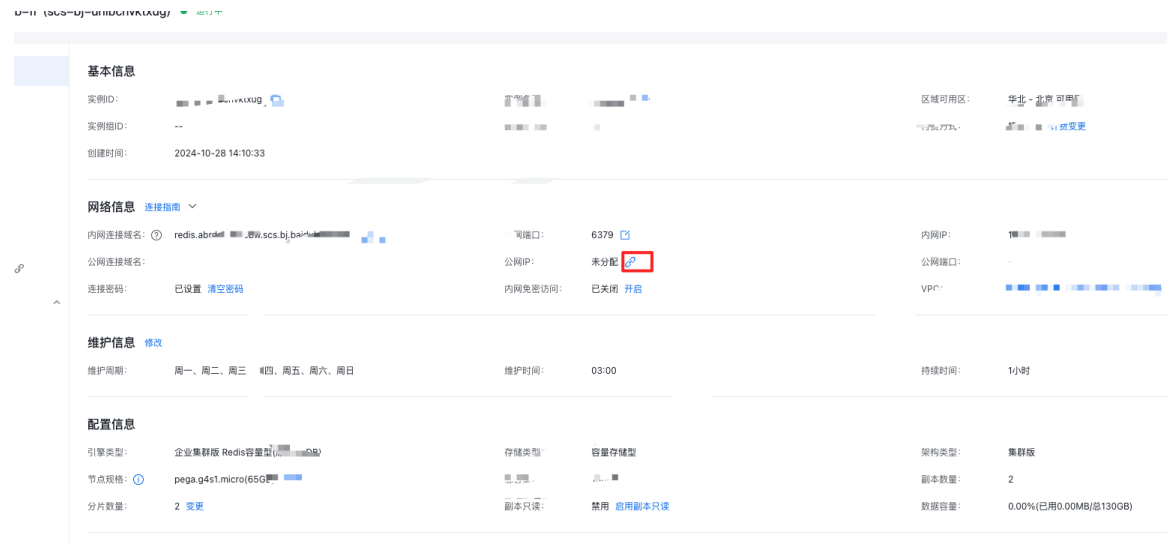
本文为您介绍通过云数据库 Redis 管理控制台绑定、解绑实例公网IP的基本方法。

说明：

- 若您当前暂无空闲的EIP，需要先创建一个EIP，用于绑定Redis实例，具体参考[创建EIP实例](#)。
- 若您的Redis实例已绑定公网IP，您可以调整EIP的带宽。
- 若您需要解绑公网IP，需要到“弹性公网IP”进行解绑操作。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 进入“云数据库 Redis-实例列表”页面，确保该实例的状态处运行中。
3. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
4. 点击公网IP右侧的设置按钮，进行公网IP绑定设置。



5. 选择您所需要的公网IP，进行绑定操作，点击确认按钮即可。





说明：绑定页面没有公网IP，点击**创建公网IP**按钮，弹出创建EIP实例页面，具体操作参考[创建EIP实例](#)，完成后重新从第4步开始操作。

6. 完成**公网IP**绑定后，您可以调整该EIP的带宽，及进行解绑操作。

修改实例端口

概述

本文为您介绍通过云数据库 Redis 管理控制台**修改实例端口**的基本方法。

说明：

- 支持修改**内网端口**和**公网端口**。
- 已绑定公网IP的实例支持**修改公网端口**。

操作步骤

- 登录[云数据库 Redis 管理控制台](#)。
- 点击“实例名称/ID”下面的目标**实例名称**，进入“实例详情”页面。
- 点击端口旁边的**变更**操作，输入变更后端口号。



4. 点击**确定**按钮，用户可以完成修改实例端口的操作。



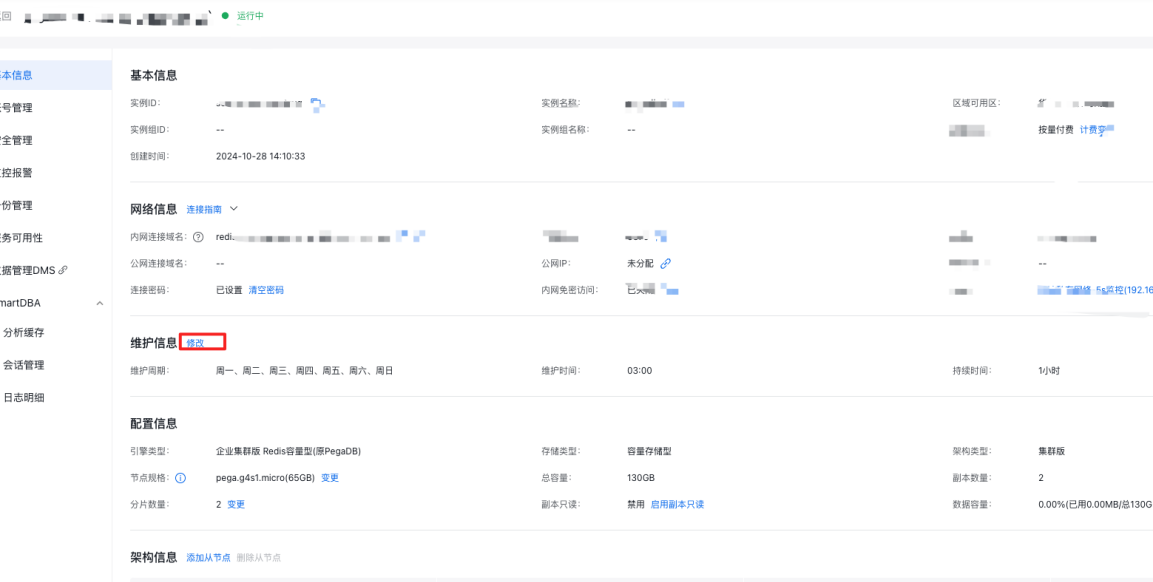
🔗 设置维护窗口

概述

本文为您介绍通过云数据库 Redis 管理控制台查看、设置维护窗口的基本方法。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 进入Redis产品服务的“云数据库 Redis-实例列表”页面。
3. 点击“实例名称/ID”下面的目标**实例名称**，进入“实例详情”页面。
4. “实例详情”页面维护信息可以查看到当前实例的**维护窗口时间**。点击维护信息右侧的**修改**按钮，进行维护周期和时间的修改。



5. 修改完成点击**确认**按钮，即可。

修改维护周期和时间

维护周期：

☒ 周一☒ 周二☒ 周三☒ 周四☒ 周五☒ 周六☒ 周日

维护时间：

开始时间：

03:00:00

持续时间：

1

小时

取消

确定

相关API [修改维护时间窗口](#)

🔗 标准版升级为集群版

概述

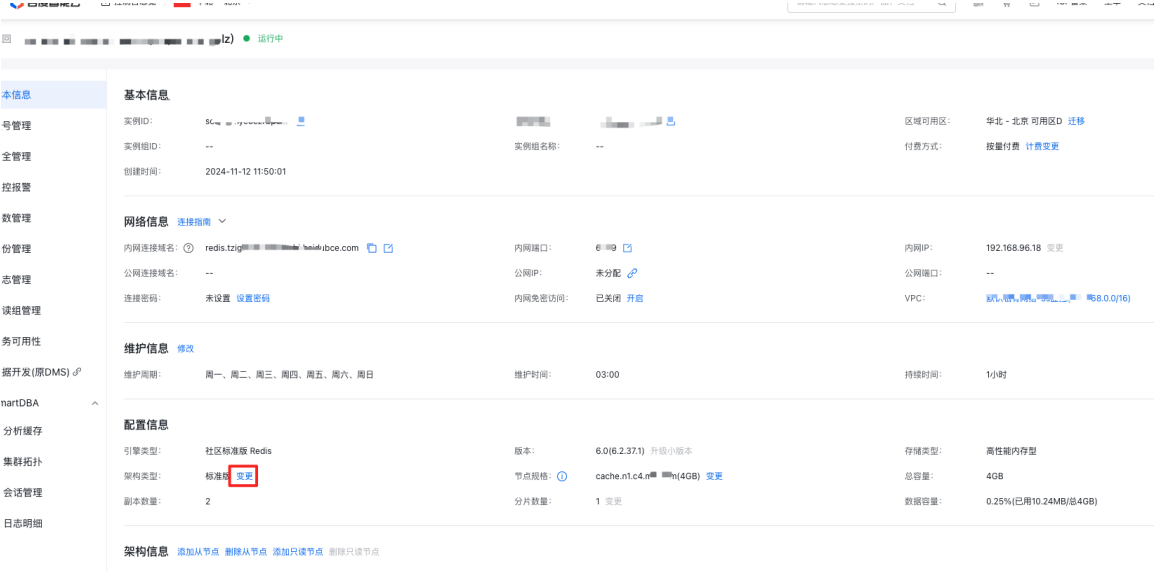
本文为您介绍通过云数据库 Redis 管理控制台**标准版实例升级为集群版**的基本方法。

注意事项

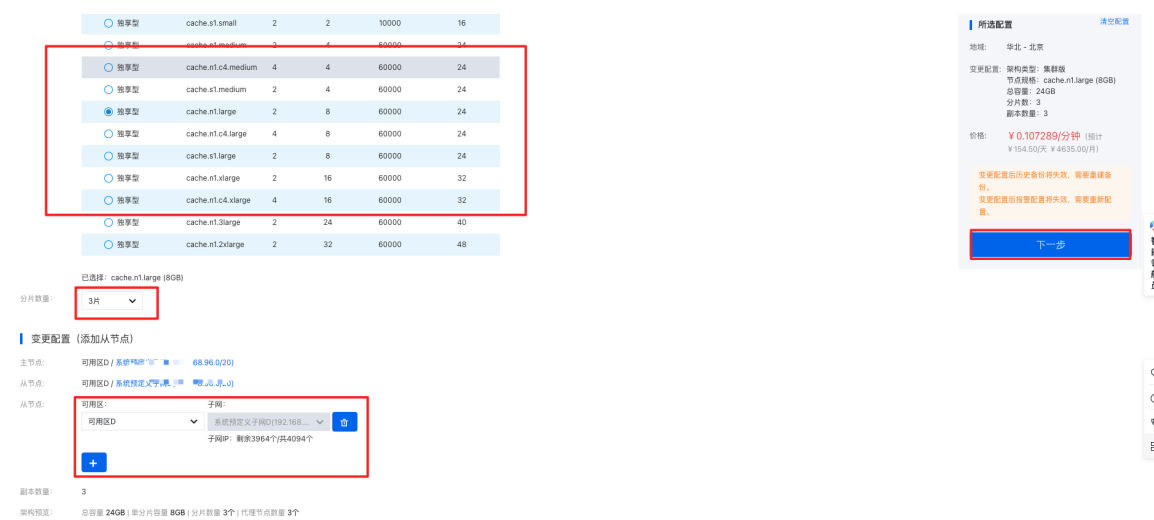
- 不支持跨版本架构升级，例如，Redis 4.0 标准架构升级至 Redis 5.0 集群架构。
- 不支持带了只读实例组的标准版实例升级到集群版，如需要升级需要先删除只读实例组才能升级。
- 热活或多活实例组中的实例不支持升级。
- 不支持单副本标准版升级架构。
- 升级后的总内存容量需要大于等于标准版总内存容量。
- 有维护时间内的任务未完成时不支持升级。
- 非运行中实例不支持升级。
- Redis7.0标准版不支持升级。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在“云数据库 Redis-实例列表”中，选择需要变更配置的实例，点击实例名称，进入实例详情。
3. 在实例详情页面的【配置信息】模块，点击架构类型后的**变更**，进入“架构升级”页面。



4. 修改实例配置信息，然后在右侧订单栏点击下一步按钮，进行订单付费。完成订单付费后，实例配置将进行变更。待实例变配完成，实例状态更新为“运行中”，您可在实例详情查看实例的最新配置。



## 版本升级

### 小版本升级

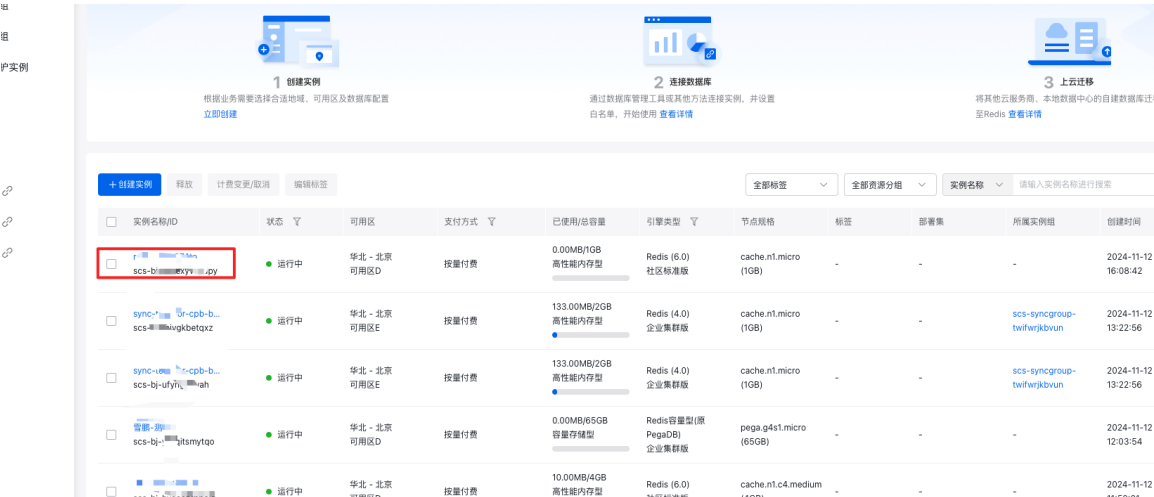
#### 概述

为提升用户体验，云数据库Redis版会不定期地发布小版本，用于丰富云产品功能或修复已知缺陷。您可以参阅本文了解Redis小版本的操作说明，在控制台进行升级小版本的操作。您也可以通过设置维护时间实现在业务低峰期进行升级任务。

- 该操作将升级实例到最新的内核版本，系统会自动检测实例的小版本，如果小版本升级按钮处于无法单击的状态，表示该实例已经是最新的小版本。
- 由于各地域版本发布进度可能有所差异，小版本发布情况以当前控制台显示为准。
- 目前云数据库Redis社区版、集群版均支持小版本升级，PegaDB、Memcache目前不支持。
- 只有运行中的实例才能进行升级小版本。
- 升级过程中实例将重启，可能会有分片数\*5秒内的闪断，请尽量选择业务低峰期进行，并确保应用具有自动重连机制。
- 对于超过16分片的实例，由于分片数较多，重启使用串型化处理，时间较长，请耐心等待。
- 单副本实例不支持升级小版本。

#### 操作步骤

1. 登录云数据库 Redis 管理控制台。
2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



3. 点击后，跳转到实例创建的基本信息界面，在配置信息的版本后点击升级小版本。



4. 点击后，会跳出弹窗提示，勾选我已了解风险并确认执行操作后，点击确认即可升级。

大版本升级

概述

云数据库 Redis 提供了升级到更高 Redis 引擎版本（大版本）的功能。新版本实例通常具备更高的性能和更完备的功能特性。用户可以通过控制台对存量实例进行大版本升级。本文将介绍如何通过控制台对 Redis 实例的大版本进行升级。

注意事项

- 版本升级方向：
- 支持低版本实例向高版本升级，不支持降级大版本。

● 支持跨大版本升级，例如，Redis 4.0 升级至6.0版本。
- 当前支持的升级路径：
- Redis 3.x 标准版可以升级到 Redis 6.x 或 7.x。

● Redis 4.x 可以升级到 Redis 6.x 或 7.x。

● Redis 5.x 可以升级到 Redis 6.x 或 7.x。

● Redis 6.x 可以升级到 Redis 7.x。
- 如果Redis 6.0 标准版 开启了 TDE（透明数据加密）功能，则不支持升级。
- 热活、多活实例组内的实例不支持升级。

- 实例副本数小于2不支持升级大版本。

升级过程

1. 升级从库：将原实例从库升级到目标版本。
2. 同步数据：新版本的从库会从原低版本的主库同步数据，确保数据一致性。
3. 主从切换：当数据同步完成后，系统会执行主从切换，将新版本的从库提升为主库。
4. 升级切换后的从库：主从切换完成后，再将低版本的从库升级到目标版本。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
3. 在实例详情页的标签页导航栏中“实例详情”页，点击配置信息模块 版本后的升级大版本按钮。



4. 在弹出窗口中，确认风险提示后，单击**确定**按钮。如果选择了可维护时间内执行，系统会在实例的可维护时间段内执行升级任务。



5. 实际过程中，实例状态将变成重启状态，**恢复运行**后内核版本升级完成。

计费变更

☞ 预付费转后付费

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍预付费转后付费的操作。

注意事项

- 已选择了自动续费的预付费实例不能进行计费变更。
- 运行状态为已到期、重启中、迁移中的实例不支持预付费转后付费。
- 用户将预付费计费方式变更为后付费时需要确保余额充足（余额>100），避免设置生效时余额检查失败。

\*\*预付费转后付费\*\*

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。



3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。
4. 点击计费变更/取消按钮，进入计费变更页面，对实例进行计费模式的变更。



5. 确认订单，点击下一步按钮，按照提示支付订单。



6. 预付费转变为后付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。
7. 在实例列表的右上角，单击 刷新列表，可查看到实例状态显示为运行中。“支付方式”显示为预付费(待变更)。实例的后付费计费方式需要等预付费到期后才会生效，未到期前可以取消计费变更。到期后支付方式会显示为后付费。

取消计费变更

用户在进行预付费转后付费后，预付费未到期前可以取消计费变更，继续使用预付费的计费方式。

1. 登录[云数据库 Redis 管理控制台](#)。



2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。



3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量取消计费变更。

4. 点击计费变更/取消按钮。



5. 在弹出的取消计费变更对话框中点击确定按钮。

取消计费变更

!

确认取消计费变更？

您已开通计费变更-预付费转后付费功能。

实例将会在到期后自动转换为后付费的计费方式。

确定

取消

6. 可以看到计费变更已取消的消息。

7. 取消计费变更成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

8. 在实例列表的右上角，单击  刷新列表，可查看到实例状态显示为运行中。“支付方式”显示为预付费。

产品服务 / 云数据库Redis-实例列表

实例列表

redis-llq1cnu2

+ 创建实例

释放

计费变更/取消

实例名称

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/> 	运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	 (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	 (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	运行中	可用区D	后付费	20.00MB/4GB (0.50%) 高性能内存型	企业版 redis (6.0)	 (4GB)	-	--	2023-08-24 14:42:14	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	运行中	可用区D	后付费	30.00MB/1GB (3.00%) 高性能内存型	社区版 redis (6.0)	 (1GB)	-	--	2023-08-17 21:42:24	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	 (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

< 1 >

后付费转预付费

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍后付费转预付费的操作。

注意事项

- 运行状态为已到期、重启中、迁移中的实例不支持后付费转预付费。

\*\*后付费转预付费\*\*

- 登录[云数据库 Redis 管理控制台](#)。
- 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

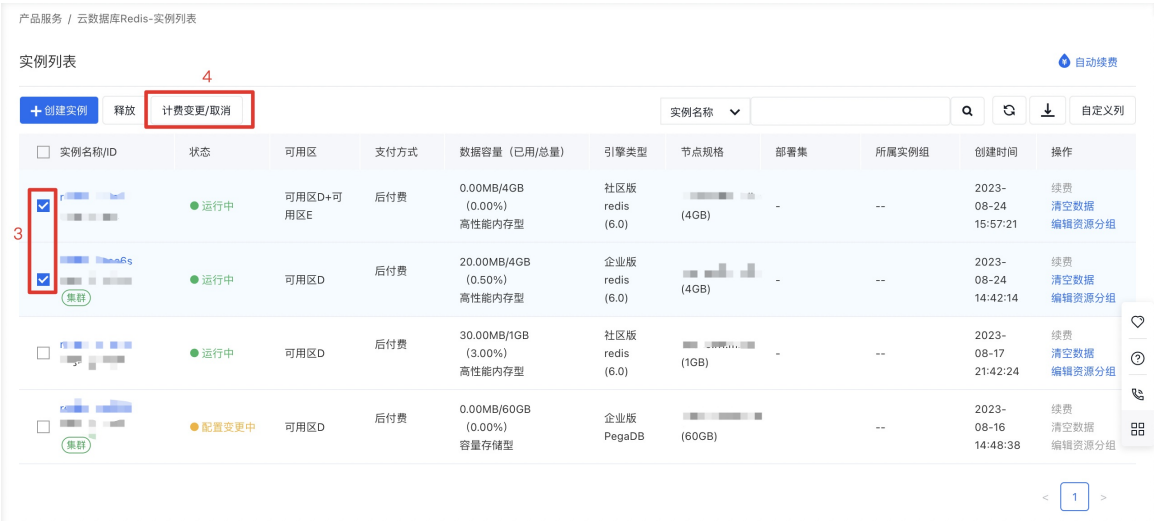
计费变更/取消

实例名称

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/> 	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	 (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	运行中	可用区D	后付费	20.00MB/4GB (0.50%) 高性能内存型	企业版 redis (6.0)	 (4GB)	-	--	2023-08-24 14:42:14	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	运行中	可用区D	后付费	30.00MB/1GB (3.00%) 高性能内存型	社区版 redis (6.0)	 (1GB)	-	--	2023-08-17 21:42:24	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> 	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	 (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

< 1 >

- 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。
- 点击计费变更/取消按钮，进入计费变更页面，对实例进行计费模式的变更。



5. 选择**预付费购买时长**，可以按月购买或按年购买，最小预付费时长为一个月。



6. 确认订单，点击**下一步**按钮，按照提示支付订单。



7. 后付费转变为预付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

8. 在实例列表的右上角，单击  刷新列表，可查看到后付费转预付费完成后，实例状态显示为**运行中**。“支付方式”显示为**预付费**。

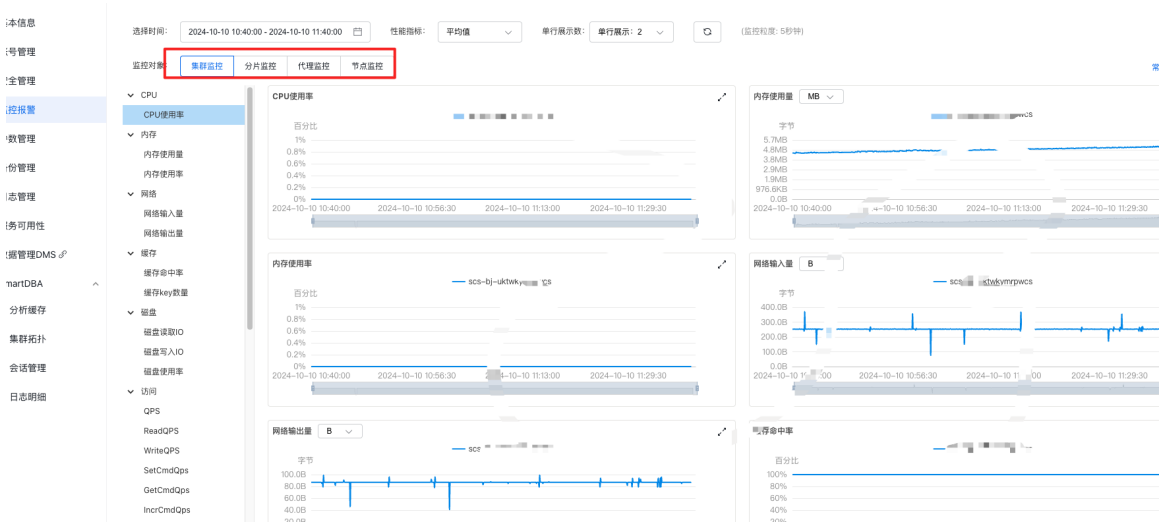
## 监控报警

### 查看监控数据

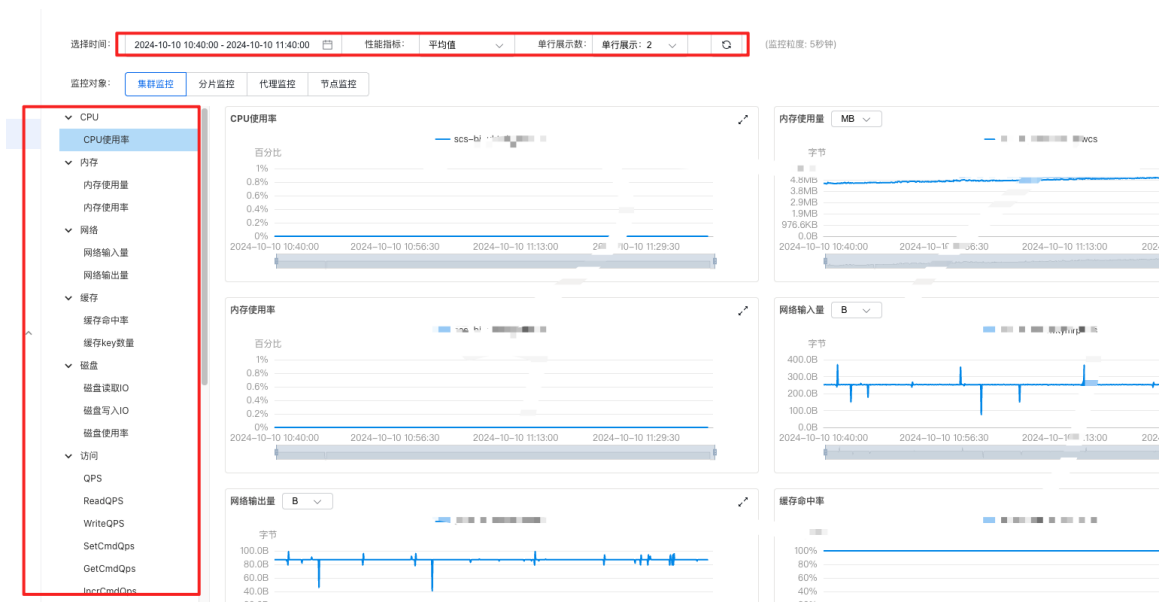
云数据库Redis版性能监控提供了**CPU使用率、内存使用率、平均时延、QPS**等性能监控指标。您可以查询过去指定时间段的监控数据，排查性能问题。

### 查看监控数据

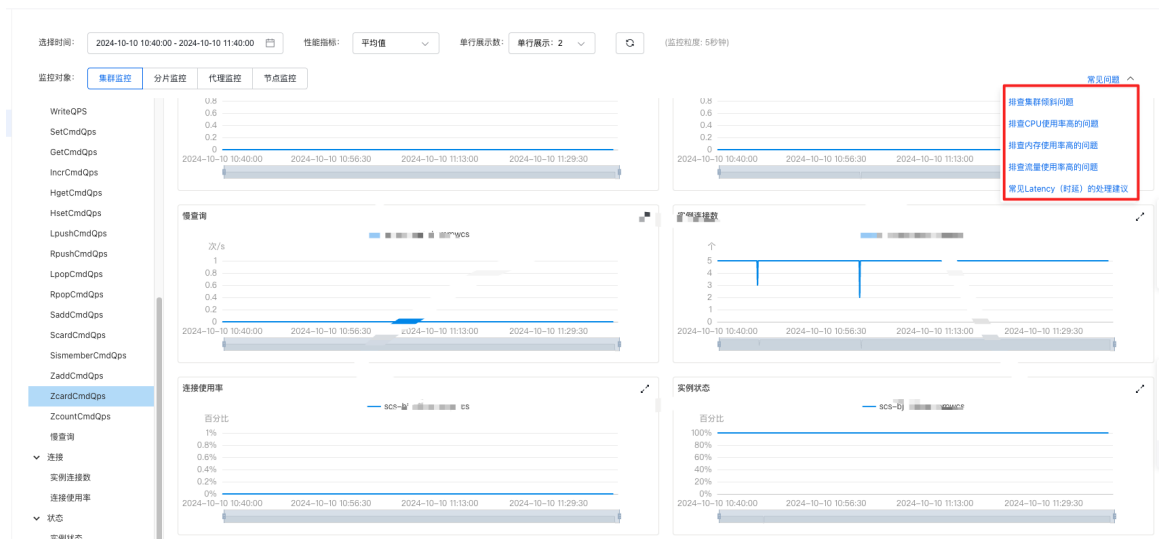
1. 登录**云数据库 Redis 管理控制台**。
2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis-实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中单击**监控报警**,进入监控报警页面。



5. 可筛选特定时间周期、性能指标计算方式后查看指标监控数据。



问题排查 针对CPU打满、内存打满、流量打满等常见性能问题，提供排查方案，可点击相关文档链接进行查看。



相关API信息 参考[BCM官网接口信息](#)

## 自定义监控项（Redis）

### 概述

云数据库 Redis 提供查看实例监控数据的功能，本文为您介绍 Redis 实例支持的监控项及其说明。

- 采样周期是指标的数据统计周期（默认为5s）。

**\*\*Redis内存型（社区版）\*\* \*\*实例监控\*\*** ||Redis控制台|单位|备注| -|-|-| |CPU|CPU使用率 |百分比|

- 常态下CPU使用率数值比主线程CPU使用率低，原因是CPU使用率=总CPU使用率/核数。| ||主线程CPU使用率 |百分比|

- 仅统计Redis主线程的CPU占用情况，单线程版本的Redis中，这基本上代表了Redis的全部CPU使用。| |内存|内存使用量|字节||内存使用率|百分比||物理内存使用量|字节||物理内存使用率|百分比||网络|节点网络输入量|字节||节点网络输出量|字节||网络输入带宽使用率|百分比||网络输出带宽使用率|百分比||缓存|命中率|百分比||缓存key数量|个||采样周期内驱逐数|个||采样周期内过期Key数量|个||内存碎片率|数值||每秒过期Key数量|个||磁盘|磁盘读取IO|字节||磁盘写入IO|字节||每秒等待IO时间|毫秒||磁盘使用率|百分比||访问|QPS|次/秒||Mget请求数|次/秒||读请求|次/秒||写请求|次/秒||其他请求|次/秒||采样周期内慢查询次数|次||每秒慢查询次数|次||平均执行时延|ms||读平均时延|ms||写平均时延|ms||实时平均执行时延|us|仅Redis 4、6、7系支持||每秒大Value请求数|次/秒|仅Redis 4系支持||每秒执行错误数|次/秒|仅Redis 4、6、7系支持||每秒Key请求数|次/秒|仅Redis 4系支持||最大执行时延|ms||P99执行时延|us|仅Redis 4、6、7系支持||其他命令平均时延|ms||处理命令总数|个||采样周期内读请求命中|次||采样周期内读请求Miss|次||客户端缓冲区利用率|百分比||连接|节点实例连接数|个||连接使用率|百分比||采样周期拒绝连接数|个||被阻塞连接数|个||最大连接数|个||状态|实例状态|百分比||复制状态|个||复制最大延迟|字节||**\*\*节点监控\*\*** ||Redis控制台|单位|备注| -|-|-| |CPU|CPU使用率|百分比|

- 常态下CPU使用率数值比主线程CPU使用率低，原因是CPU使用率=总CPU使用率/核数。| ||主线程CPU使用率 |百分比|

- 仅统计Redis主线程的CPU占用情况，单线程版本的Redis中，这基本上代表了Redis的全部CPU使用。| |内存|节点内存使用量|字节||内存使用率|百分比||物理内存使用量|字节||物理内存使用率|百分比||网络|节点网络输入量|字节||节点网络输出量|字节||缓存|命中率|百分比||节点缓存key数量|个||采样周期内驱逐数|个||采样周期内过期Key数量|个||内存碎片率|数值||每秒过期Key数量|个||磁盘|节点磁盘读取IO|字节||节点磁盘写入IO|字节||磁盘使用率|百分比||每秒等待IO时间|毫秒||访问|QPS|次/秒||采样周期内读请求命中|个||Mget请求数|次/秒||读请求|次/秒||写请求|次/秒||其他请求|次/秒||采样周期内慢查询次数|次||平均执行时延|ms||读平均时延|ms||写平均时延|ms||实时平均执行时延|ms|仅Redis 4、6、7系支持||每秒执行错误数|次/秒|仅Redis 4、6、7系支持||最大执行时延|ms||P99执行时延|us|仅Redis 4、6、7系支持||其他命令平均时延|ms||采样周期内读请求miss|个||处理命令总数|个||采样周期内处理命令数|次||每秒处理命令数|次||每秒慢查询次数|次||每秒大Value请求数|次/秒|仅Redis 4系支持||每秒Key请求数|次/秒|仅Redis 4系支持||客户端缓冲区利用率|百分比||连接|节点实例连接数|个||连接使用率|百分比||采样周期拒绝连接数|个||被阻塞连接数|个||最大连接数|个||状态|实例状态|百分比||复制状态|个||复制最大延迟|字节||

**\*\*Redis内存型（集群版）\*\* 集群监控** ||Redis控制台|单位|备注| -|-|-| |CPU|CPU使用率 |百分比|

- 常态下CPU使用率数值比主线程CPU使用率低，原因是CPU使用率=总CPU使用率/核数。| ||主线程CPU使用率 |百分比|

- 仅统计Redis主线程的CPU占用情况，单线程版本的Redis中，这基本上代表了Redis的全部CPU使用。| |内存|内存使用量|字节||内存使用率|百分比||物理内存使用量|字节||物理内存使用率|百分比||网络|网络输入量|字节||网络输出量|字节||网络输入带宽使用率|百分比||网络输出带宽使用率|百分比||缓存|命中率|百分比||缓存key数量|个||每秒过期Key数量|个||内存碎片率|数值||采样周期内驱逐数|个||采样周期内过期Key数量|个||磁盘|磁盘读取IO|字节||磁盘写入IO|字节||每秒等待IO时间|毫秒||磁盘使用率|百分比||访问|QPS|次/秒||ReadQPS|次/秒||WriteQPS|次/秒||SetCmdQps|次/秒||GetCmdQps|次/秒||IncrCmdQps|次/秒||HgetCmdQps|次/秒||HsetCmdQps|次/秒||LpushCmdQps|次/秒||RpushCmdQps|次/秒||LpopCmdQps|次/秒||RpopCmdQps|次/秒||SaddCmdQps|次/秒||ScardCmdQps|次/秒||SismemberCmdQps|次/秒||ZaddCmdQps|次/秒||ZcardCmdQps|次/秒||ZcountCmdQps|次/秒||采样周期内慢查询次数|次||采样周期内处理命令数|次||每秒处理命令数|次||采样周期内读请求命中|次||采样周期内读请求Miss|次||平均执行时延|us||实时平均执行时延|us|仅Redis 4、6、7系支持||P999执行时延|us||P99执行时延|us|仅Redis 4、6、7系支持||采样周期内慢查询次数|次||客户端缓冲区利用率|百分比||连接|实例连接数|个||连接使用率|百分比||最大连接数|个||被阻塞连接数|个||采样周期拒绝连接数|个||状态|实例状态|百分比||复制状态|个||复制最大延迟|byte| **分片监控** ||Redis控制台|单位|备注| -|-|-| |CPU|CPU使用率 |百分比|

- 常态下CPU使用率数值比主线程CPU使用率低，原因是CPU使用率=总CPU使用率/核数。| ||主线程CPU使用率 |百分比|

- 仅统计Redis主线程的CPU占用情况，单线程版本的Redis中，这基本上代表了Redis的全部CPU使用。| |内存|内存使用量|字节||内存使用率|百分比||物理内存使用量|字节||物理内存使用率|百分比||网络|节点网络输入量|字节||节点网络输出量|字节||网络输入带宽使用率|百分比||网络输出带宽使用率|百分比||缓存|缓存key数量|个||命中率|百分比||每秒过期Key数量|

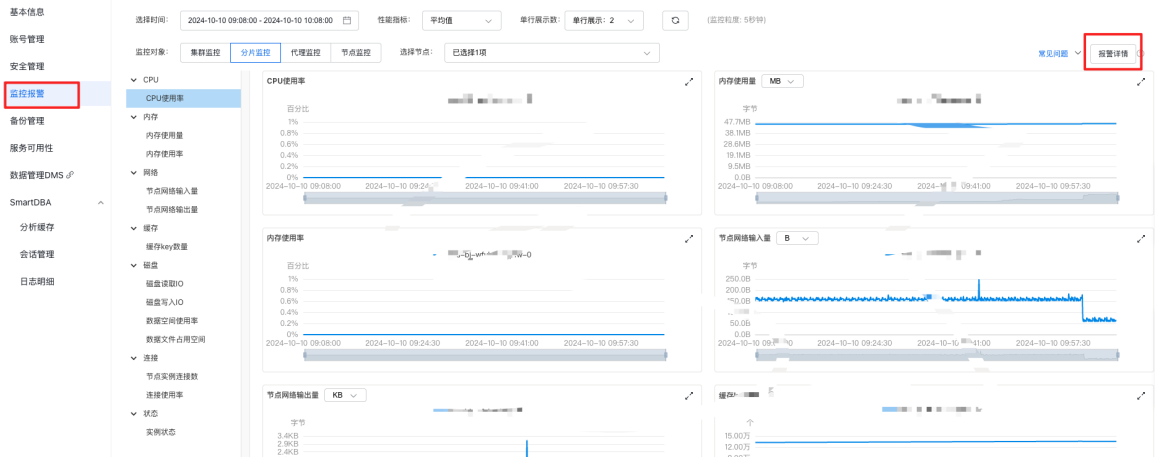
个||内存碎片率|数值||采样周期内驱逐数|个||采样周期内过期Key数量|个||磁盘|磁盘读取IO|字节||磁盘写入IO|字节||每秒等待IO时间|毫秒||磁盘使用率|百分比||访问|QPS|次/秒||采样周期内处理命令数|次||每秒处理命令数|次||Mget请求数|次/秒||读请求|次/秒||写请求|次/秒||其他请求|次/秒||采样周期内慢查询次数|次||每秒慢查询次数|次||平均执行时延|ms||读平均时延|ms||写平均时延|ms||实时平均执行时延|us|仅Redis 4、6、7系支持|每秒执行错误数|次/秒|仅Redis 4、6、7系支持|每秒大Value请求数|次/秒|仅Redis 4系支持|每秒Key请求数|次/秒|仅Redis 4系支持|最大执行时延|ms||P99执行时延|us|仅Redis 4、6、7系支持|其他命令平均时延|ms||采样周期内读请求命中|次||采样周期内读请求Miss|次||处理命令总数|次||客户端缓冲区利用率|百分比||连接|节点实例连接数|个||连接使用率|百分比||采样周期拒绝连接数|个||被阻塞连接数|个||最大连接数|个||状态|实例状态|百分比||复制状态|个||复制最大延迟|字节||**代理监控**||Redis控制台|单位|备注|---|CPU|CPU使用率|百分比||网络|网络输入量|字节||网络输出量|字节||访问|ProxyQPS|次/秒||Mget请求数|次/秒||读请求|次/秒||写请求|次/秒||其他请求|次/秒||ReadQPS|次/秒||WriteQPS|次/秒||SetCmdQps|次/秒||GetCmdQps|次/秒||IncrCmdQps|次/秒||HgetCmdQps|次/秒||HsetCmdQps|次/秒||LpushCmdQps|次/秒||RpushCmdQps|次/秒||LpopCmdQps|次/秒||RpopCmdQps|次/秒||SaddCmdQps|次/秒||ScardCmdQps|次/秒||SismemberCmdQps|次/秒||ZaddCmdQps|次/秒||ZcardCmdQps|次/秒||ZcountCmdQps|次/秒||采样周期内慢查询次数|次||平均执行时延|us||P999执行时延|us||P99执行时延|us|仅Redis 4、6、7系支持|连接|实例连接数|个||连接使用率|百分比||最大连接数|个||状态|实例状态|百分比||**节点监控**||Redis控制台|单位|备注|---|CPU|CPU使用率|百分比|

- 常态下CPU使用率数值比主线程CPU使用率低，原因是CPU使用率=总CPU使用率/核数。||主线程CPU使用率|百分比|
- 仅统计Redis主线程的CPU占用情况，单线程版本的Redis中，这基本上代表了Redis的全部CPU使用。||内存|节点内存使用量|字节||内存使用率|百分比||物理内存使用量|字节||物理内存使用率|百分比||网络|节点网络输入量|字节||节点网络输出量|字节||网络输入带宽使用率|百分比||网络输出带宽使用率|百分比||缓存|节点缓存key数量|个||命中率|百分比||每秒过期Key数量|个||内存碎片率|数值||采样周期内驱逐数|个||采样周期内过期Key数量|个||访问|QPS|次/秒||采样周期内读请求Miss|个||处理命令总数|个||Mget请求数|次/秒||采样周期内处理命令数|次||每秒处理命令数|次/秒||读请求|次/秒||写请求|次/秒||其他请求|次/秒||采样周期内慢查询次数|次||每秒慢查询次数|次||读平均时延|ms||写平均时延|ms||平均执行时延|ms||实时平均执行时延|us|仅Redis 4、6、7系支持|每秒执行错误数|次/秒|仅Redis 4、6、7系支持|每秒大Value请求数|次/秒|仅Redis 4系支持|每秒Key请求数|次/秒|仅Redis 4系支持|最大执行时延|ms||P99执行时延|us|仅Redis 4、6、7系支持|其他命令平均时延|ms||采样周期内读请求命中|次||客户端缓冲区利用率|百分比||磁盘|节点磁盘读取IO|字节||节点磁盘写入IO|字节||磁盘使用率|百分比||每秒等待IO时间|毫秒||连接|节点实例连接数|个||连接使用率|百分比||采样周期拒绝连接数|个||被阻塞连接数|个||最大连接数|个||状态|实例状态|百分比||复制状态|个||复制最大延迟|字节||

## 🔗 设置报警规则

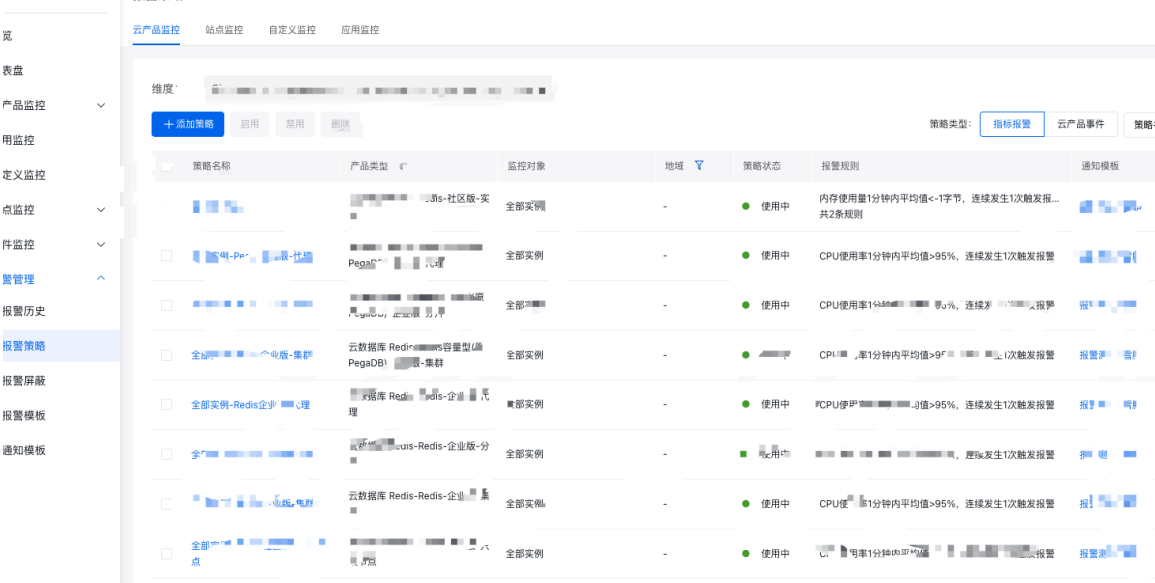
**背景** 云监控BCM是针对百度云资源和互联网应用提供监控的服务，您可以对重要监控指标设置报警规则。当云数据库 Redis 监控指标异常（超过设定阈值）或实例触发高可用切换时，云监控BCM平台会向您发送警告信息。 **操作步骤**

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中点击**监控报警**，进入监控页面。
5. 点击页面右上角的**报警详情**按钮，即可进入实例的“报警管理”页面设置报警规则。



添加报警策略

- 1. 选择报警策略页签，进入报警策略页面。



- 2. 在此页面中可以创建报警策略，具体分为：基本信息、策略规则、报警通知三部分，具体如下。

基本信息

- 基本信息包含：策略名称：产品的类型：监控对象。其中策略名称用户自定义输入。产品类型默认为云数据库Redis，监控对象为实例ID。

基本信息

策略名称：

请输入策略名称

产品类型：

云数据库 Redis

监控对象：

策略规则 策略规则包括策略类型、报警等级、报警规则，具体如下。

- 策略类型：指标报警。
- 报警等级：通知、警告、重要、严重。
- 报警规则：选择的报警模版发生的次数，设置出发规则。
- 报警重复提醒：报警发生后如未恢复正常，触发/不触发，重复报警。
- 报警恢复处理：报警发生后如数据恢复正常，触发/不触发，恢复报警。



- **数据不足报警**：监控数据不足时，**触发/不触发**，报警动作。

策略规则

策略类型：

指标报警

报警级别：

通知

警告

重要

严重

①

报警规则：已创建1个规则组

请选择

1分钟

和值

>=

请输入阈值

B

每 60 s 判断1次，连续发生 

1

 次，触发报警

添加且规则

删除

添加或规则

使用报警模板

?

报警重复提醒：报警发生后知未恢复正常，

不触发

重复报警

报警恢复处理：报警发生后知数据恢复正常，

触发

恢复报警

数据不足报警：监控数据不足时，

不触发

报警动作

报警通知

1. 点击**新建模版**，进入创建报警模版的页面，填写以下参数。

参数	说明
动作名称	创建报警动作名称
可用地域	不同地域的报警动作仅适用于各自地域的报警策略
通知方式	百度智能云目前支持“邮件通知”，“短信通知”和“邮件+短信通知”的形式
通知对象	支持“用户组”和“用户”两种形式 若您当前没有已创建的通知对象，您可以点击"添加用户组"来进行创建
回掉地址	触发该模版之后的会调路径

2. 查看已经新建的**报警记录列表**。

报警通知

新建模板

共1条，已选 0 条，还可以选 5 条

请输入通知模板名称

Q

C

<input type="checkbox"/>	通知模板名称	用户/用户组	通知渠道	报警回调	定时关闭	更新时间	操作
暂无数据							

10条/页

<

1

>

跳转至

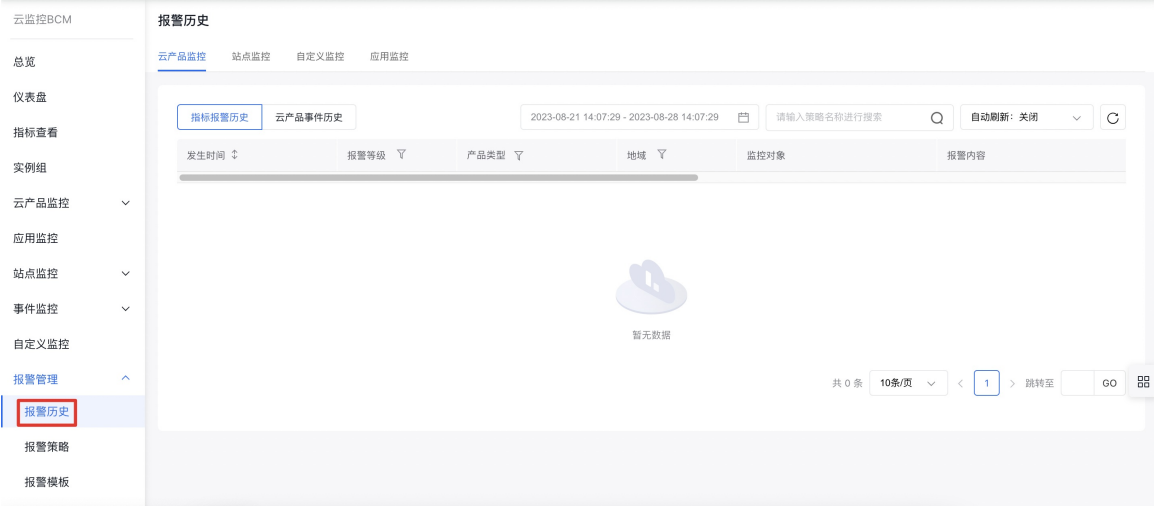
GO

3. 点击**保存**后，页面提示创建成功，报警动作列表显示新创建的报警模版。

查看报警历史

选择左侧导航栏的**报警历史**页签，进入报警历史页面，您可查看此实例产生的报警事件。





相关API

信息 参考[BCM官网](#)

参数管理

参数说明

云数据库 Redis 提供自定义部分实例参数的功能，本文为您介绍 Redis 实例支持自定义的参数及其说明。

参数说明

参数名	详细信息	说明	适用版本
append only	默认值：yes 类型：字符串 可取值：[yes   no   partial]	是否启用 AOF 持久化	除单副本
disable_commands	默认值：flushall,flushdb 类型：字符串 可取值：[flushall   flushdb   keys   hgetall] 更改生效：重新启动后	用户自定义禁用命令	全部
maxmemory-policy	默认值：volatile-ttl 类型：字符串 可取值：[volatile-lru   allkeys-lru   volatile-random   allkeys-random   volatile-ttl   noeviction]	当内存超量时的键淘汰策略。可选： volatile-lru：只从已设置过期时间的数据集中挑选最近最少使用的数据淘汰 allkeys-lru：挑选最近最少使用的数据淘汰 volatile-random：只从已设置过期时间的数据集中任意选择数据淘汰 allkeys-random：任意选择数据淘汰 volatile-ttl：只从已设置过期时间的数据集中挑选将要过期的数据淘汰 noeviction：不进行数据淘汰	全部
hash-max-ziplist-entries	默认值：512 类型：整数 可取值：[0-999999999999999]	当 hash 对象同时满足以下两个条件时，hash 对象使用 ziplist 编码： 1. hash 对象保存的键值对数量小于或者等于 hash-max-ziplist-entries 参数值 2. hash 对象保存的所有键值对的键和值的字符串长度都小于或者等于 hash-max-ziplist-value 参数值	全部
hash-max-ziplist-value	默认值：64 类型：整数 可取值：[0-999999999999999]	当 hash 对象同时满足以下两个条件时，hash 对象使用 ziplist 编码： 1. hash 对象保存的键值对数量小于或者等于 hash-max-ziplist-entries 参数值 2. hash 对象保存的所有键值对的键和值的字符串长度都小于或者等于 hash-max-ziplist-value 参数值	全部
set-max-	默认值：512 类型：整数	当 set 对象同时满足以下两个条件时，set 对象使用 intset 编码： 1. set 对象中的元素数量小于或者等于 set-max-intset-entries 参数值	全部

intset-entries	可取值：[0-999999999999999]	2. set 对象中的所有元素都是 64 位有符号十进制整数	
zset-max-ziplist-entries	默认值：128 类型：整数 可取值：[0-999999999999999]	当 zset 对象同时满足以下两个条件时，zset 对象使用 ziplist 编码： 1. zset 对象保存的键值对数量小于或者等于 zset-max-ziplist-entries 参数值 2. zset 对象保存的所有键值对的键和值的字符串长度都小于或者等于 zset-max-ziplist-value 参数值	全部
zset-max-ziplist-value	默认值：64 类型：整数 可取值：[0-999999999999999]	当 zset 对象同时满足以下两个条件时，zset 对象使用 ziplist 编码： 1. zset 对 保存的键值对数量小于或者等于 zset-max-ziplist-entries 参数值 2. zset 对象保存的所有键值对的键和值的字符串长度都小于或者等于 zset-max-ziplist-value 参数值	全部
notify-keyspace-events	默认值：空 类型：字符串 可取值：[K   E   g   \$   l   s   h   z   x   e   A]	客户端可以利用键空间通知来接收那些影响 Redis 数据集的事件通知，notify-keyspace-events 参数指定了接收通知的类型。参数值可由多个字符组成，空值表示禁用通知。各字符含义如下： K：键空间事件 E：键事件事件 g：通用命令（非特定类型），例如 DEL、EXPIRE、RENAME 等 \$：字符串命令 l：列表命令 s：集合命令 h：哈希命令 z：有序集合命令 x：键过期事件 e：键淘汰事件 A：参数 g\$lshzxe 的别名，因此 "AKE" 字符串表示所有的事件	标准版
timeout	默认值：0 类型：整数 可取值：[0-100000]	断开空闲客户端之前等待的秒数，零值表示从不断开空闲客户端	全部
hz	默认值：10 类型：整数 可取值：[1-500]	设置 Redis 后台任务执行频率，比如清除过期键任务。参数值越大，CPU 消耗越大，延迟越小，建议不要超过 100	全部
lazyfree-lazy-eviction	默认值：no 类型：字符串 可取值：[yes   no]	是否开启基于lazyfree的驱逐功能	全部
lazyfree-lazy-expire	默认值：yes 类型：字符串 可取值：[yes   no]	是否开启基于lazyfree的过期Key删除功能	全部
lazyfree-lazy-server-del	默认值：yes 类型：字符串 可取值：[yes   no]	DEL命令是否基于lazyfree异步删除数据	全部
lazyfree-lazy-user-del	默认值：yes 类型：字符串 可取值：[yes   no]	执行DEL命令时是否基于lazyfree异步删除数据	全部
slowlog-log-slower-than	默认值：10000 类型：整数	设置慢日志的记录阈值，即设置对执行时间大于多少微秒的操作进行记录，单位：微秒，参数值1000000表示1秒	全部

than	可取值：[10000-10000000]		
------	----------------------	--	--

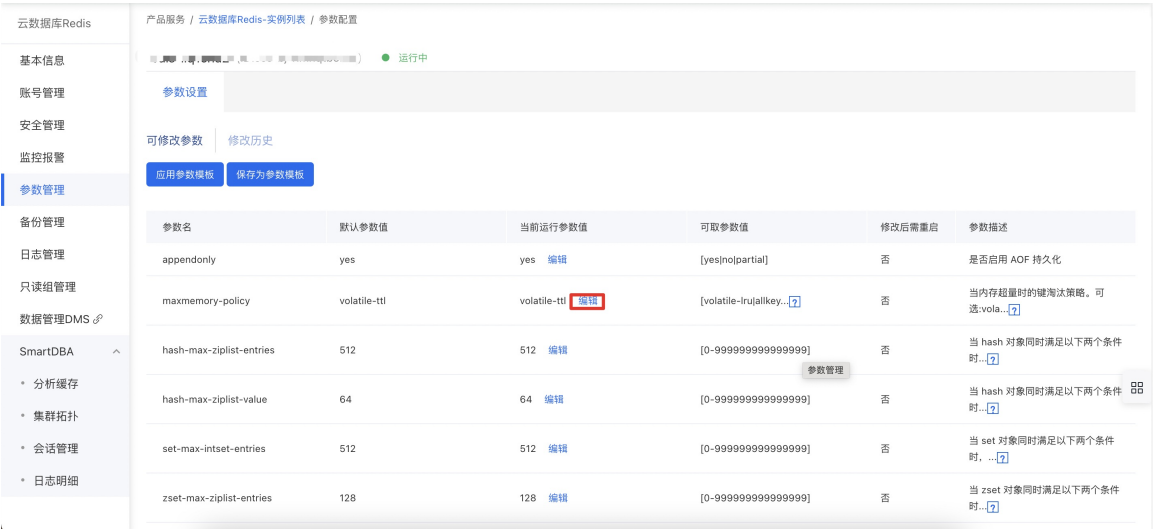
🔗 设置实例参数

概述

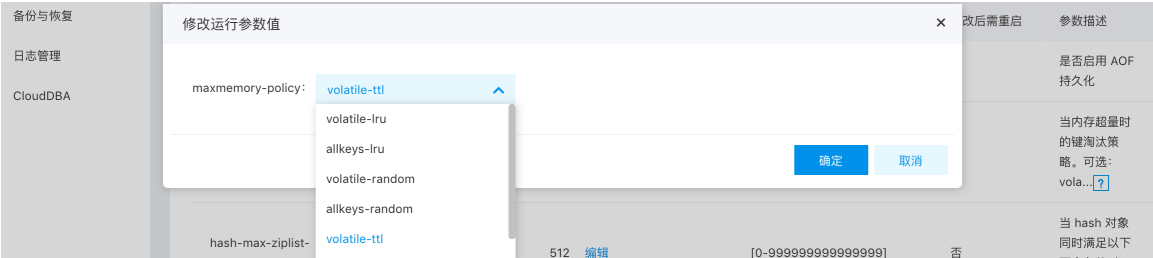
云数据库 Redis 提供自定义部分实例参数的功能，本文为您介绍通过管理控制台修改各参数值的通用方法。

设置实例参数

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **参数管理**。
4. 在**参数管理**页面中，选择**可修改参数**页签，点击目标参数“当前运行参数值”列的**编辑**。



5. 在弹出窗口中修改参数值，单击**确定**按钮。



6. 刷新页面，目标参数的“当前运行参数值”已经修改。

相关 API

- [修改参数](#)
- [获取参数列表](#)

🔗 查看参数修改历史

概述

云数据库 Redis 提供自定义部分实例参数的功能，本文为您介绍通过管理控制台查看修改参数值的历史记录。

查看参数修改历史

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **参数管理**。

4. 在参数管理页面中，选择 **修改历史** 页签，可查看参数修改历史记录。



相关 API

- [获取参数列表](#)

🔒 禁用危险命令

概述

云数据库 Redis 提供禁用危险命令的功能。用户可通过管理控制台修改 `disable_commands` 参数，实现禁用部分可能导致数据误删除、服务不稳定的危险命令。

说明：

- 支持禁用的命令包括：FLUSHALL、FLUSHDB、KEYS、HGETALL。
- FLUSHALL、FLUSHDB 命令会删除大量甚至全部数据。
- KEYS、HGETALL 命令的不合理使用会造成 Redis 阻塞、CPU 打满。
- 修改禁用危险命令参数后，需要重启服务方可生效。

禁用危险命令

1. 登录 [云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **参数管理**。
4. 在参数管理页面中，选择**可修改参数**页签，找到列表中 `disable_commands` 参数，点击其“当前运行参数值”列的**编辑**。

zset-max-ziplist-value	64	64 <a href="#">编辑</a>	[0-999999999999999]	否	当 zset 对象同时满足以下两个条件时... <a href="#">?</a>
notify-keyspace-events	-	- <a href="#">编辑</a>	[K E I g \$ l s h z x e... <a href="#">?</a>	否	客户端可以利用键空间通知来接收那些影响... <a href="#">?</a>
timeout	0	0 <a href="#">编辑</a>	[0-100000]	否	断开空闲客户端之前等待的秒数。零值表示从... <a href="#">?</a>
hz	10	10 <a href="#">编辑</a>	[1-500]	否	设置 Redis 后台任务执行频率，比如... <a href="#">?</a>
disable_commands	flushall,flushdb	flushall,flushdb  编辑	[flushall flushdb ke... <a href="#">?</a>	是 <a href="#">?</a>	用户自定义禁用命令

5. 在弹出窗口中选择需要禁用的命令，单击**确定**。

修改运行参数值

disable\_commands:

☒ flushall

☒ flushdb

☒ keys

☐ hgetall

确定

取消

6. 用户在控制台点击**重启实例**，重启完成后生效。

🔒 关闭AOF落盘

## 概述

云数据库 Redis 提供 AOF 落盘开关的功能。用户可通过管理控制台修改 `appendonly` 参数，实现启用/禁用 AOF 落盘。

**注意：** 该参数不适用于单副本的实例。

关闭 AOF 落盘

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标**实例名称**，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **参数管理**。
4. 在**参数管理**页面中，选择**可修改参数**页签，找到 `appendonly` 参数，点击其“当前运行参数值”列的**编辑**。

云数据库Redis

基本设置

账号管理

安全管理

监控报警

参数管理

备份管理

日志管理

只读组管理

数据管理DMS

SmartDBA

分析缓存

集群拓扑

会话管理

日志明细

运行中

参数设置

可修改参数

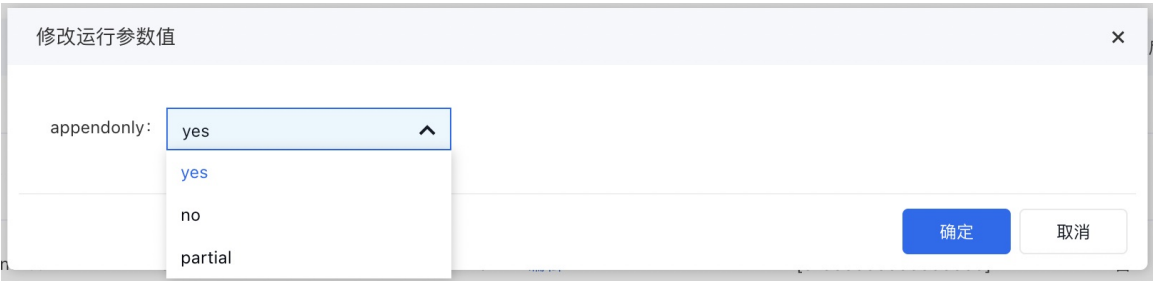
修改历史

应用参数模板

保存为参数模板

参数名	默认参数值	当前运行参数值	可取参数值	修改后需重启	参数描述
appendonly	yes	yes 编辑	[yes no partial]	否	是否启用 AOF 持久化
maxmemory-policy	volatile-ttl	volatile-ttl 编辑	[volatile-lru allkey-lru]	否	当内存超量时的键淘汰策略。可选volatile-lru
hash-max-ziplist-entries	512	512 编辑	[0-9999999999999999]	否	当 hash 对象同时满足以下两个条件时...
hash-max-ziplist-value	64	64 编辑	[0-9999999999999999]	否	当 hash 对象同时满足以下两个条件时...
set-max-intset-entries	512	512 编辑	[0-9999999999999999]	否	当 set 对象同时满足以下两个条件时...
zset-max-ziplist-entries	128	128 编辑	[0-9999999999999999]	否	当 zset 对象同时满足以下两个条件时...
zset-max-ziplist-value	64	64 编辑	[0-9999999999999999]	否	当 zset 对象同时满足以下两个条件时...

- 在弹出窗口中选择需要开关状态，单击**确定**按钮。
  - 支持的开关状态包括：YES、NO、PARTIAL。
    - YES**：主节点启用 AOF，备节点启用 AOF，适用于偏重数据可靠性。
    - NO**：主节点禁用 AOF，备节点禁用 AOF，适用于偏重服务性能。
    - PARTIAL**：主节点禁用 AOF，备节点启用 AOF，适用于平衡数据可靠性与服务性能。



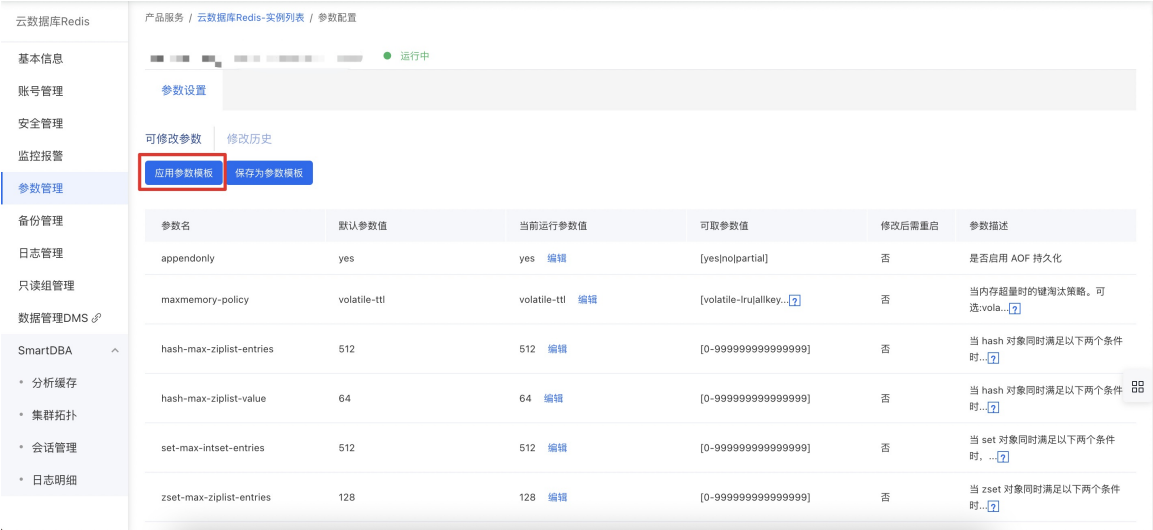
保存为参数模版

概述

云数据库 Redis 提供自定义部分实例参数的功能，本文为您介绍通过管理控制台修改各参数值的通用方法。

保存为参数模版

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击参数管理。
- 4. 在参数管理页面中，选择可修改参数页签，点击保存为参数模版按钮。



- 5. 在弹出页面中，输入参数模版名称、备注，选择集群类型、引起类型及版本。确认信息后，点击确定按钮，完成参数模版保存。

保存为参数模板

\* 模板名称:

abc\_testname

\* 集群类型:

标准版

\* 引擎类型:

redis

\* 引擎版本

6.0

备注

最多256个字符（一个汉字等于三个字符）

参数列表:

参数名	当前参数值	是否需重启	可修改参数值	参数描述
appendonly	yes	否	yes	是...

取消

确认

应用参数模版

**概述** 云数据库 Redis 提供自定义部分实例参数的功能，本文为您介绍通过管理控制台修改各参数值的通用方法。

**注意事项**

- 当您应用参数模板后，实例的原有参数将被重新设置，与当前模板的参数值保持一致。
- 若当前参数模板包含重启参数，则您在应用参数模板后，需要手动重启实例；若未重启实例，则相应参数不生效。
- 建议您在业务低峰期应用含有重启参数的模板。

**\*\*应用参数模版\*\*** 登录云数据库 Redis 管理控制台，进入实例列表页面。

1. 点击目标实例名称，进入实例详情页面。
2. 在实例详情页的左侧导航栏中，点击**参数管理**。
3. 在**参数管理**页面中，选择**可修改参数**页签，点击“应用参数模版”按钮。

云数据库Redis

产品服务 / 云数据库Redis-实例列表 / 参数配置

参数设置

可修改参数 修改历史

应用参数模板 保存为参数模板

参数名	默认参数值	当前运行参数值	可取参数值	修改后需重启	参数描述
appendonly	yes	yes 编辑	[yes no partial]	否	是否启用 AOF 持久化
maxmemory-policy	volatile-ttl	volatile-ttl 编辑	[volatile-lru allkey-lru]	否	当内存超量时的键淘汰策略。可选:volatile-lru
hash-max-ziplist-entries	512	512 编辑	[0-999999999999999]	否	当 hash 对象同时满足以下两个条件时...
hash-max-ziplist-value	64	64 编辑	[0-999999999999999]	否	当 hash 对象同时满足以下两个条件时...
set-max-intset-entries	512	512 编辑	[0-999999999999999]	否	当 set 对象同时满足以下两个条件时...
zset-max-ziplist-entries	128	128 编辑	[0-999999999999999]	否	当 zset 对象同时满足以下两个条件时...

在弹出页面中，选择参数模版名称、显示相关参数模版的信息及是否需要重启实例，确认信息后，点击**确定**按钮。

应用参数模板

1. 当您应用参数模板后，实例的原有参数将被重新设置，与当前模板的参数值保持一致。

2. 若当前参数模板包含重启参数，则您在应用参数模板后，需要手动重启实例；若未重启实例，则相应参数不生效。

3. 建议您在业务低峰期应用含有重启参数的模板。

模板名称：

abc\_testname (模板id:scs-templ-lezjnp5)

参数数量：16

是否需要重启：是

参数列表：以下仅展示应用参数模板后有变化的参数

是否需要重启：

☐ 不重启

☐ 维护时间重启

☒ 立即重启

参数名	当前参数值	是否需重启	可修改参数值	参数描述
hash-max-ziplist-entries	512	否	512	当 has...
hash-max-ziplist-value	64	否	64	当 has...
...	...	...	...	当 set ...

取消

确认

## 日志管理

### 慢日志

**概述** 云数据库 Redis 提供查询指定时间内的慢请求日志的功能，帮助您发现并解决性能问题，定位实例中的慢查询命令或潜在的性能瓶颈等问题。慢日志记录了执行时间超过指定阈值（**slowlog-log-slower-than**）的命令，默认为10毫秒，您也可以自定义该参数。本文为您介绍通过管理控制台查询慢日志的通用方法。

### 注意事项

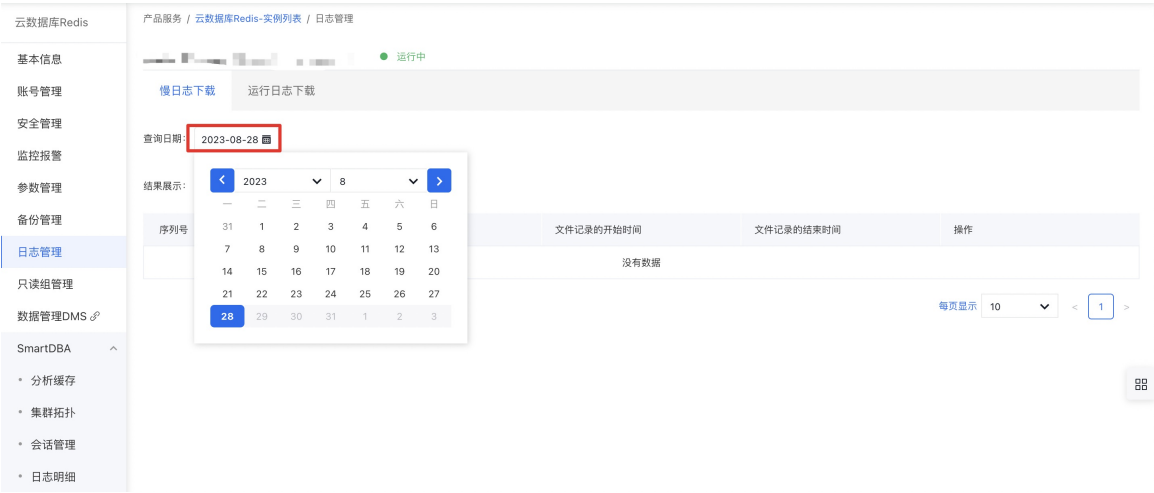
- 最大查询范围支持最近七天。
- 单次查询最多展示前 10000 条数据。

**相关参数** | 参数名 | 说明 | | ---|---| | **slowlog-log-slower-than** | 设置慢日志阈值，默认为10毫秒| 参数设置方法请参见[设置实例参数](#)。

### 查询慢日志

- 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 点击目标**实例名称**，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击 **日志管理**。
- 在日志管理页面中，选择**慢日志**下载页签，点击查询日期处的日历组件，分别选择“开始日期和结束日期”，然后点击**确定按钮**。





5. 如果实例为集群版，可以点击结果展示处的下拉菜单，选择需要查询全部分片或某个具体分片。



6.如果慢日志数量较多，可在每页展示处选择展示数量。



运行日志

概述

云数据库 Redis 提供查询指定时间内的运行日志的功能，帮助您发现并解决运维问题。本文为您介绍通过管理控制台查询运行日志的通用方法。

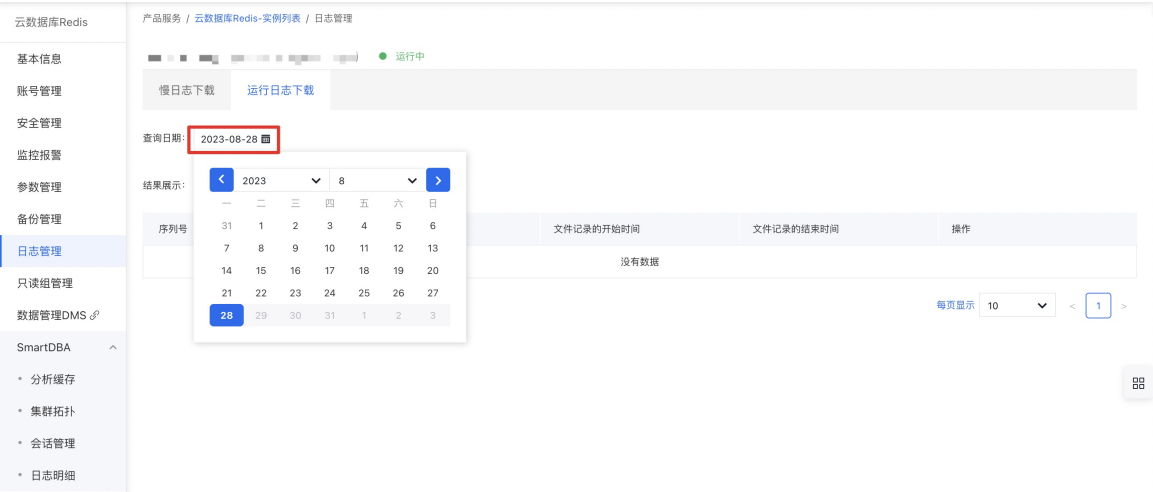
注意事项

- 最大查询范围支持最近七天。
- 单次查询最多展示前10000条数据。

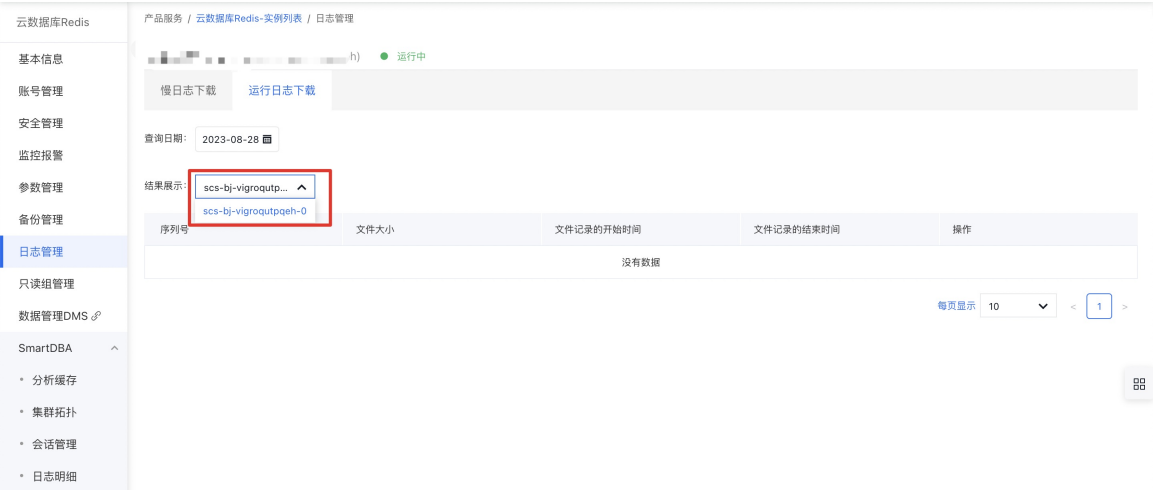
查询运行日志

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。

3. 在实例详情页的左侧导航栏中，点击 **日志管理**。
4. 在**日志管理**页面中，点击**运行日志**页签，点击查询日期处的日历组件，分别选择开始日期和结束日期，然后点击**确定**按钮，即可查询此查询日期区间的运行日志。



5. 如果实例为集群版，可以点击**结果展示**处的下拉菜单，选择需要查询的分片。



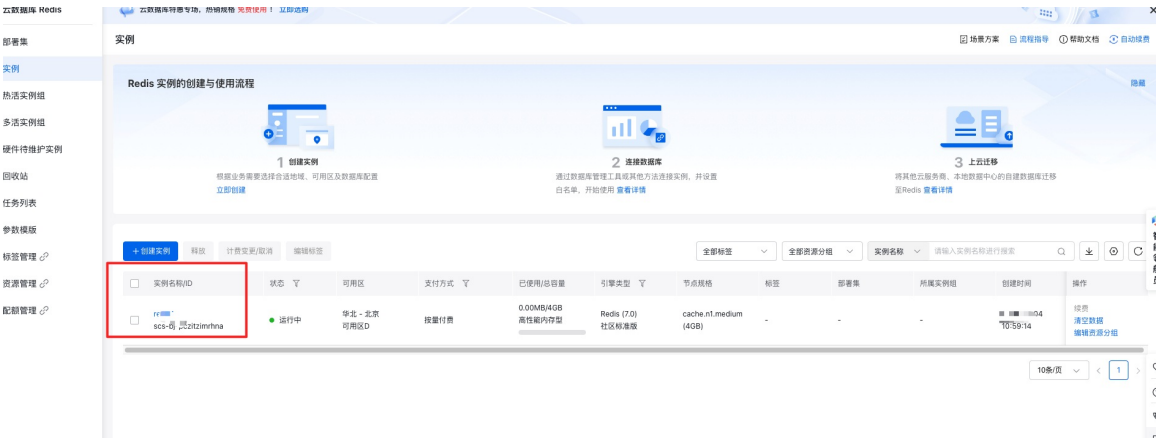
🔗 审计日志

**概述** 本文为您介绍开启云数据库Redis**审计日志**的方法，可用于帮助开发运维人员定位性能问题。

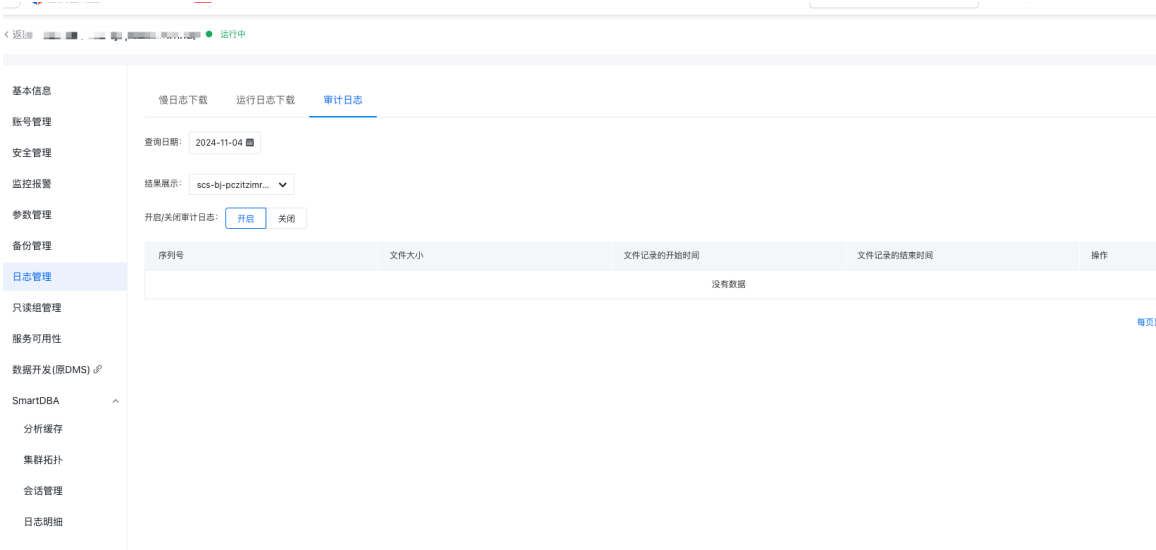
**注意事项**

- 仅**Redis 4.0、6.0、7.0**实例支持开启**审计日志**。
- 热活实例组从角色不支持**操作审计日志**相关功能。
- 开通**审计日志**后，系统将只记录写操作的审计信息，可能会对实例造成5%~15%的性能损失。
- 当命令的参数过多、参数过长或命令总长度过长时，**审计日志**中的命令不会完整显示。 **操作步骤**

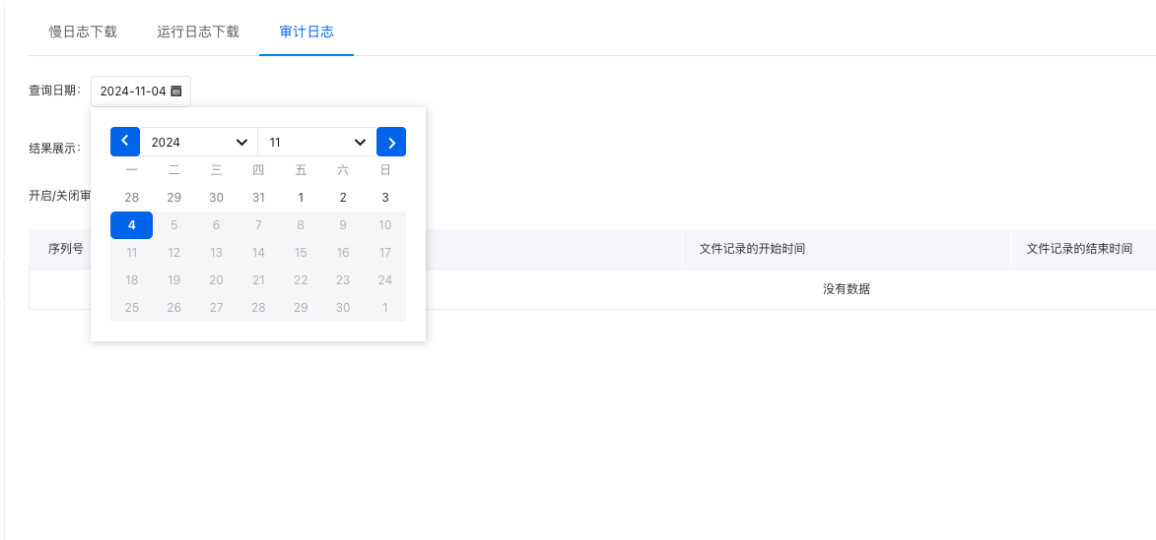
1. 登录[云数据库Redis管理控制台](#)，左上角选择地域信息进行地域切换。
2. 进入指定地域的实例列表页，点击需要开启的实例名称/ID。



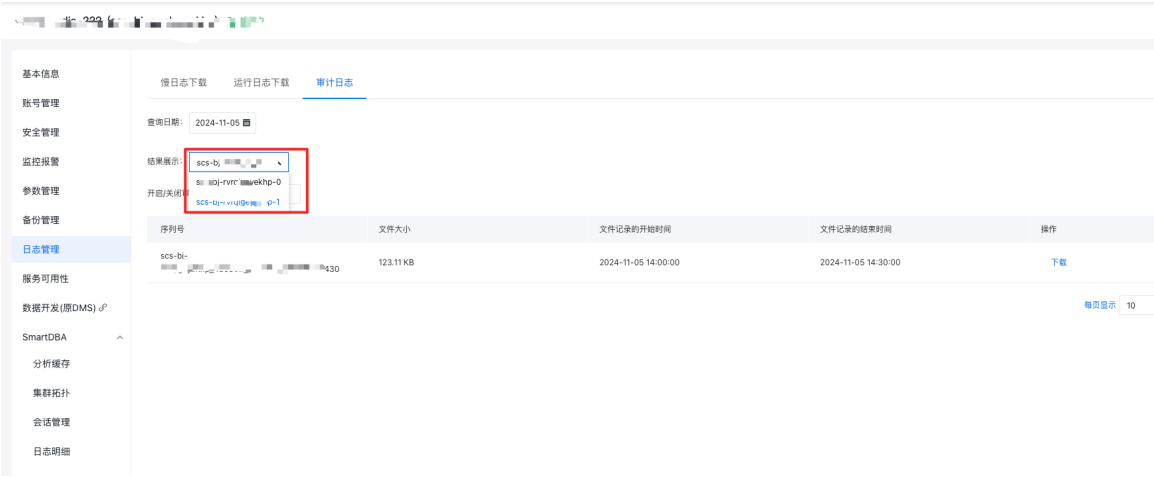
3. 进入实例详情页面，左侧标签页导航栏中选择日志管理，点击审计日志页签，切换到审计日志子页面。



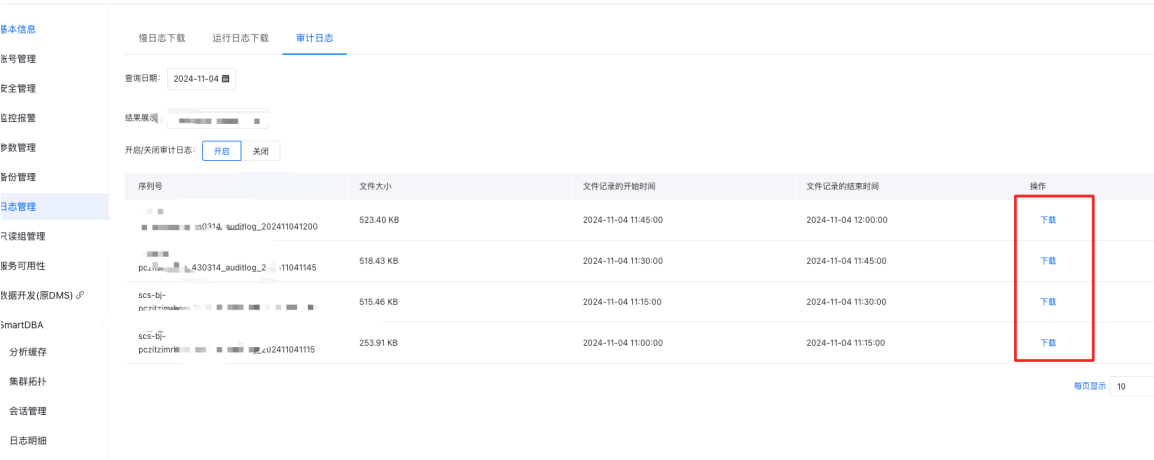
4. 开启后，点击查询日期处的日历组件，选择需要查看的日期，即可查询此日期产生的审计日志。



5. 如果实例为集群版，可以点击结果展示处的下拉菜单，选择需要查询的分片。



6. 若需要下载审计日志，点击日志文件后的下载按钮，系统会自动将日志以文稿格式下载到本地，您可以在本地使用合适的工具打开文件进行查看。



相关API接口 [日志列表](#)

[审计日志开关](#)

SmartDBA

[大Key分析](#)

概述

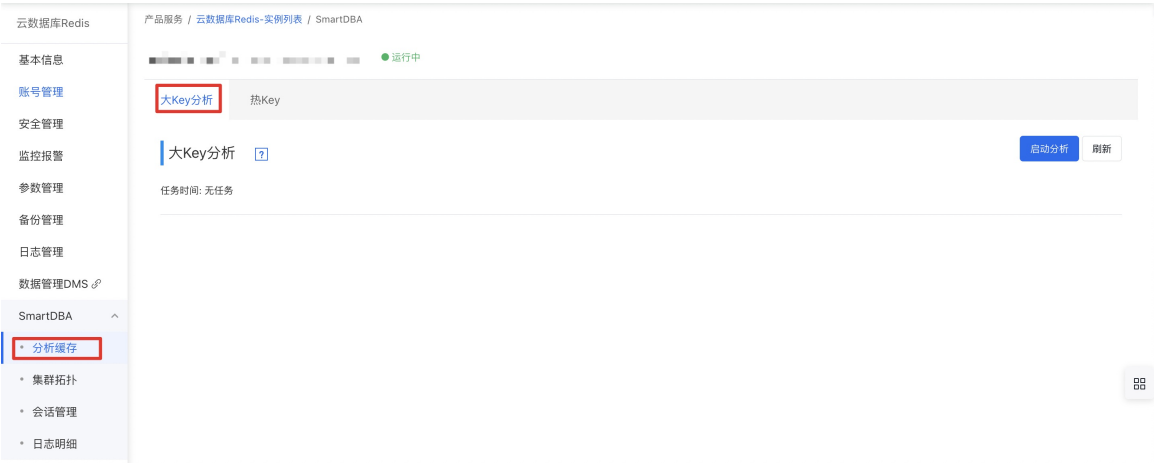
云数据库 Redis 提供分析缓存占用和大 Key 分布情况的功能。本文为您介绍通过管理控制台进行缓存分析的通用方法。

注意事项

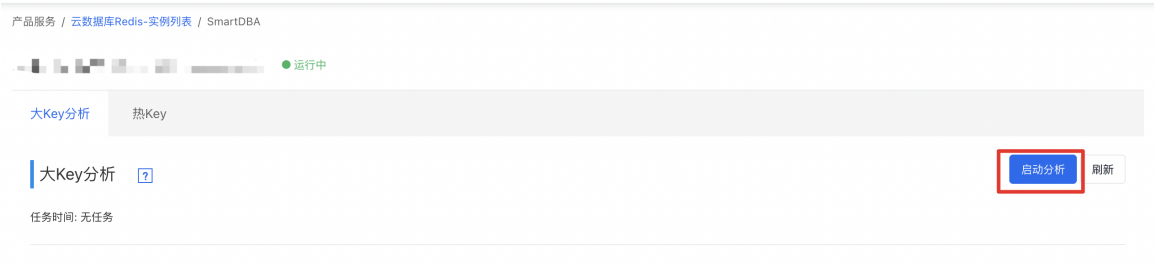
- 启动分析任务后，实例将处于缓存分析中的状态，期间控制台各项操作将被禁用。
- 分析任务完成后，将展示占用空间最大的前 100 个 Key。
- 不支持单副本的实例。

操作指南

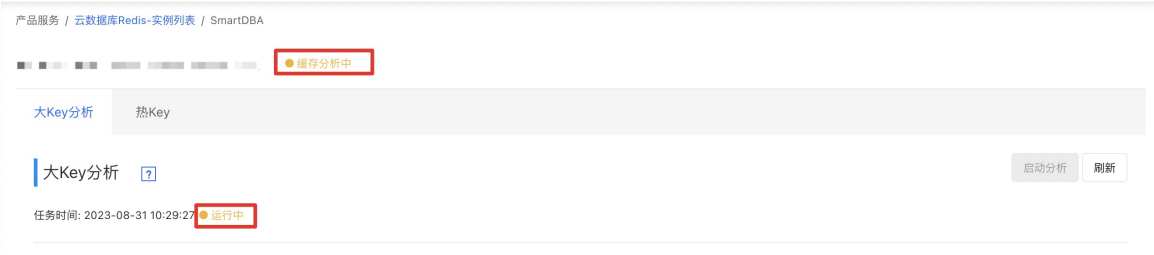
- 登录 [云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 点击目标实例名称/ID，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击 **SmartDBA**，默认选中**大Key**页签。



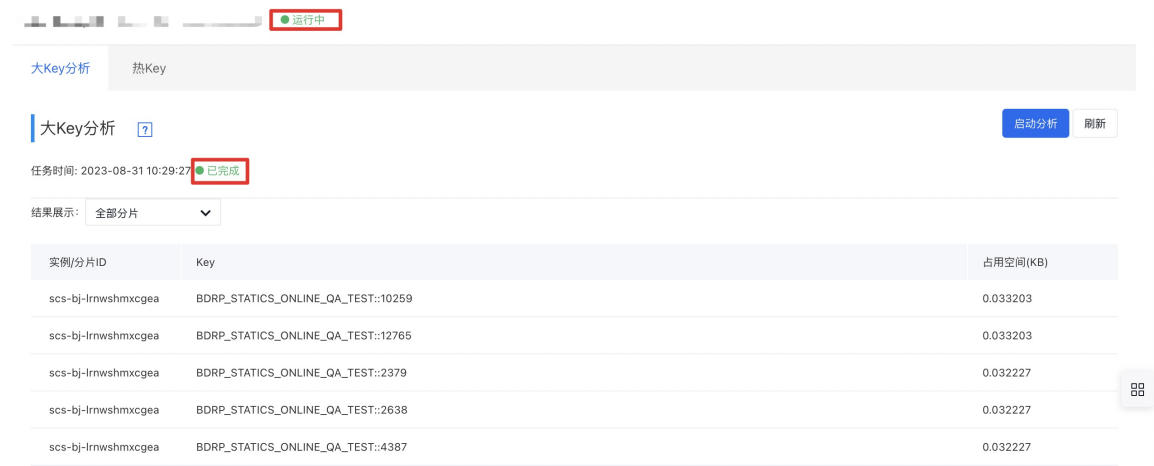
3. 在 大Key 页面中，点击“大Key”模块的启动分析按钮。



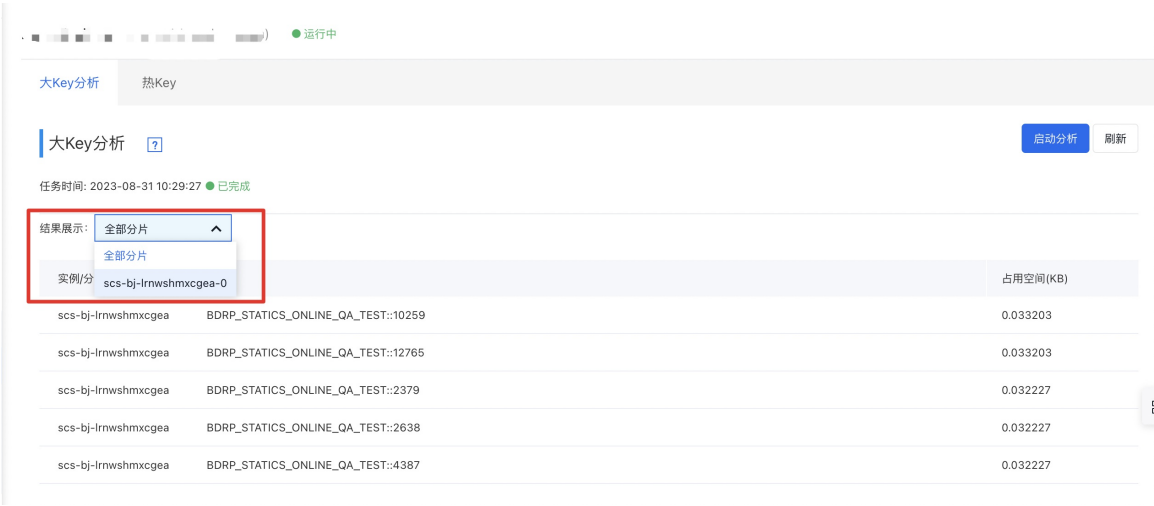
4. 分析任务状态变为运行中且实例状态显示为缓存分析中。



5. 当任务运行完毕，点击刷新按钮，分析任务状态变为已完成，分析结果将展示在下方。您可以在该页面再次启动新的分析任务，新任务将覆盖旧任务的分析结果。



5. 如果实例为集群版，可以点击结果展示处的下拉菜单，选择需要查询全部分片或某个具体分片。



热Key分析

概述

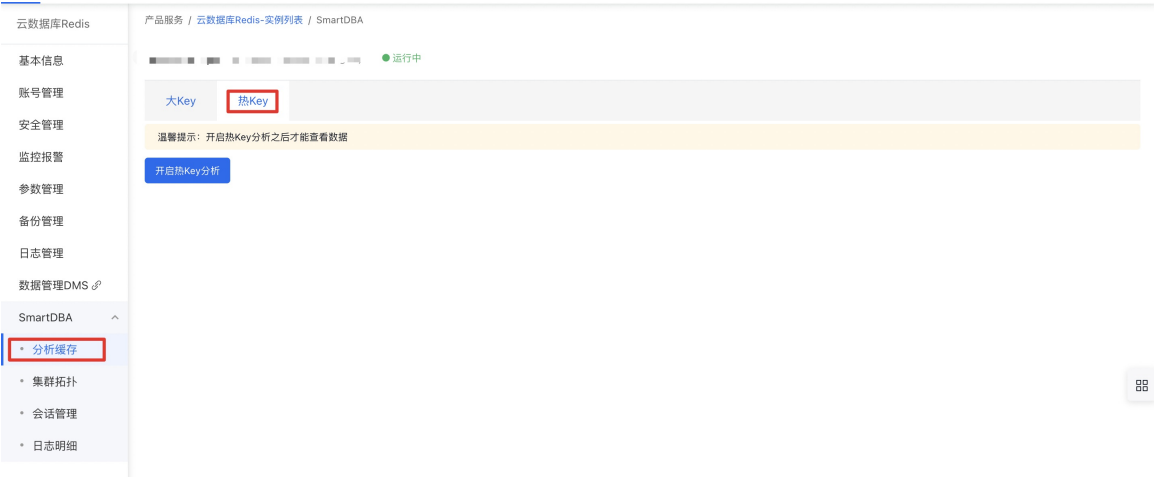
云数据库 Redis 提供热 Key 分布情况的功能。本文为您介绍通过管理控制台进行缓存分析的通用方法。

注意事项

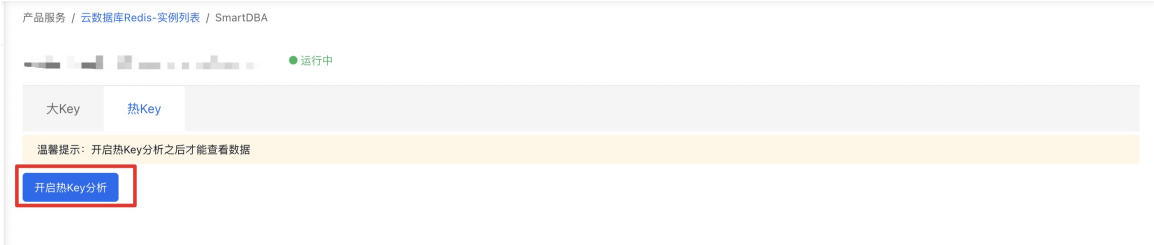
- 实时数据取最近60s的数据，当打开自动刷新后取最近30s的数据。
- 不支持单副本的实例。

开启热Key分析缓存

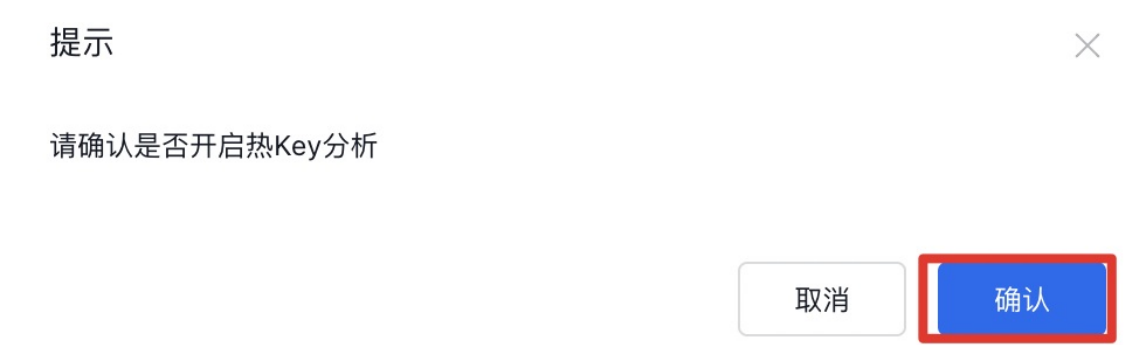
1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称/ID，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 SmartDBA，默认选中大Key页签，点击热Key页签，切换热key页面。



4. 在 热Key 页面中，点击开启热Key分析按钮。



5.弹出提示窗口，点击**确定**按钮。



6. 打开**实时数据**和**历史数据**页签。



实时数据

以下操作前提是**开启热Key分析缓存**的操作步骤：

1. 点击**热Key**页签中的**实时数据**子页签，切换到**实时数据**列表，可以看到相关提示。



2. 在**节点ID**选择列表中可以选择指定实例或**全部(全部实例)**。

3. 在这一步，如果是**自动刷新**是已打开状态，则可以忽略此步骤，否则则需要开启它。



4.打开后，系统30秒将更新次列表数据，需要用户手动点击**刷新**按钮。



历史数据

以下操作前提是开启热Key分析缓存的操作步骤：

- 1. 点击热Key页签中的实时数据子页签，切换到历史数据列表，可以看到相关提示。



- 2. 在节点ID选择列表中可以点击指定实例或全部(全部实例)。

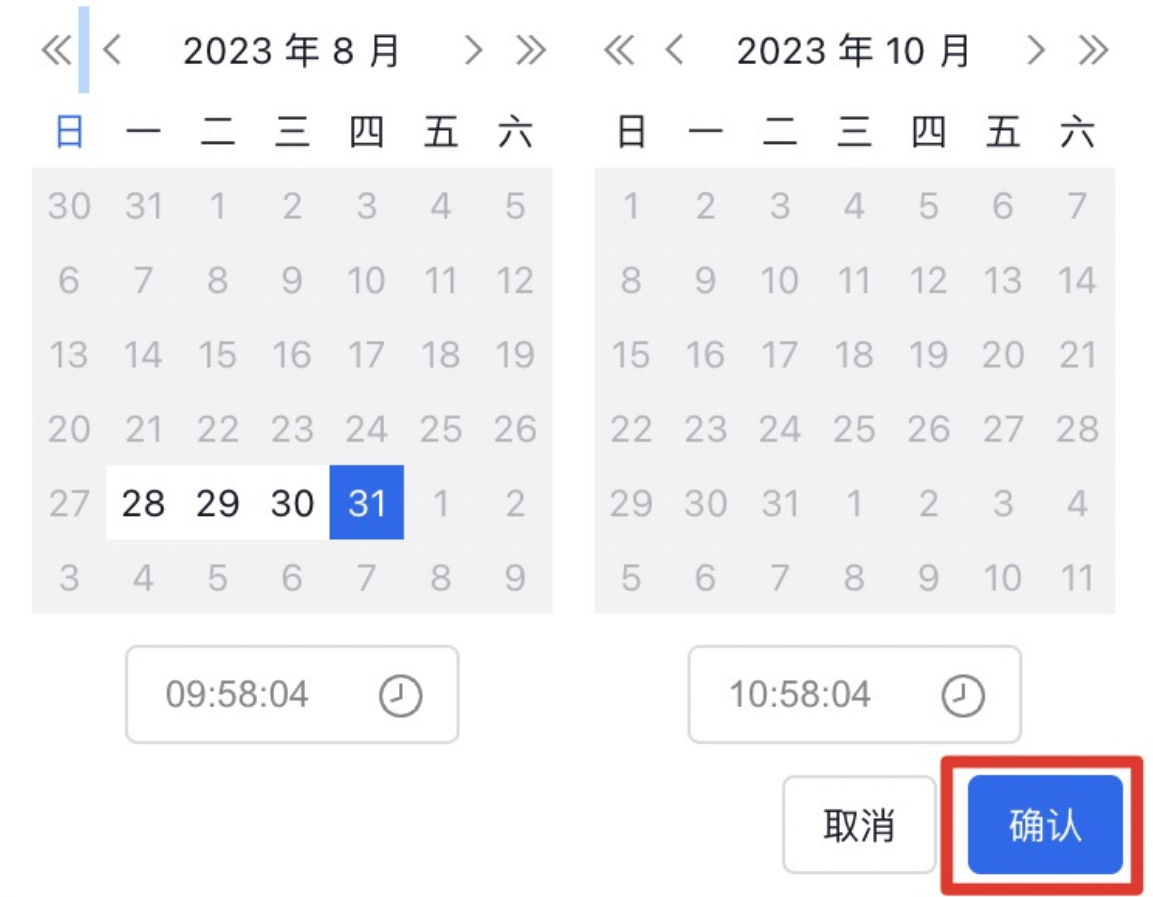


- 3. 在时间范围选择列表中可以点击框内日历图标。



- 4. 弹出日历组件，选择开始和结束范围，点击确定按钮。





5.点击列表上的刷新按钮，列表将刷新数据。



关闭热Key分析缓存

以下操作前提是开启热Key分析缓存的操作步骤：

- 1. 点击实时数据和历史数据页签最右侧关闭热Key分析按钮。



- 2. 弹出提示窗口，点击确定按钮。

提示

×

请确认是否关闭热Key分析

取消

确认

3.自动刷新后，会出现开启热Key分析按钮，表示热Key分析缓存已经关闭。



会话管理

概述

会话管理用于查看当前实例的会话状态、会话详情、会话统计等。

操作步骤

- 1. 登录云数据库 Redis管理控制台，进入实例列表页面。
- 2. 点击目标实例名称/ID，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 SmartDBA菜单下的会话管理菜单，打开会话管理页面，节点选择默认选择集群节点信息项，可查看集群会话概要。



4.在选择节点选择具体节点实例，，将出现诊断类型具体包含会话详情、会话统计、会话拓扑。例如：  
lrwshmxgea\_redis\_244827\_0。

● 运行中

选择节点：

主节点：scs-bj-lrnwshmxgea\_redis\_244827...

数据更新时间：2023-08-31 15:15:41

诊断类型：

会话详情

会话统计

会话拓扑

会话概要

会话总数：2个

活跃总数：0个

最大执行时间：71298秒

会话详情

访问来源：

支持左前缀匹配

持续时间：

请选择

单位（秒）

空闲时间：

请选择

单位（秒）

重置

查询

kill所选会话

kill会话历史

提示：

会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↑ ↓	空闲时间(idle) ↑ ↓	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	6	172.17.32.27:58279		info	71297秒	28秒	0	N	12
<input type="checkbox"/>	5	172.17.32.158:44025		setex	71298秒	5秒	0	N	13

会话详情

点击诊断类型的会话详情按钮，出现会话详情页面，页面分为会话概要、会话详情两部分。

会话概要

会话概要包含当前会话会话总数、活跃总数、最大执行时间三个部分的统计，具体如下。

会话概要

会话总数：2个

活跃总数：0个

最大执行时间：73002秒

会话详情

会话详情包含会话列表，可以对会话筛选、会话进行kill、查看会话框历史等操作，具体如下。

会话详情

访问来源：

支持左前缀匹配

持续时间：

请选择

单位（秒）

空闲时间：

请选择

单位（秒）

重置

查询

kill所选会话

kill会话历史

提示：

会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↑ ↓	空闲时间(idle) ↑ ↓	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

• 会话筛选

在列表上放输入访问来源、持续时间、空闲时间等条件，如果不加入筛选条件，则查询所有，完成并确认筛选条件后，点击查询按钮。

访问来源：

支持左前缀匹配

持续时间：

请选择

单位（秒）

空闲时间：

请选择

单位（秒）

重置

查询

kill所选会话

kill会话历史

提示：

会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↑ ↓	空闲时间(idle) ↑ ↓	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

- 访问来源是IP地址。
- 持续时间和空闲时间选择条件是大于（>）、小于（<）。
- 持续时间单位是输入条件是秒数。

kill 所选会话

1. 在会话列表选择所需要kill的会话记录，勾选当前行复选框，点击会话列表右上角上 kill所选会话按钮。

访问来源：

支持左前缀匹配

持续时间：

请选择

单位（秒）

空闲时间：

请选择

单位（秒）

重置

查询

kill所选会话

kill会话历史

提示：

会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input checked="" type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)  ↕	空闲时间(idle)  ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input checked="" type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

2. 弹出对话框kill会话窗口，确认当前会话后，点击确定按钮。

kill会话

会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)	空闲时间(idle)	数据库ID(db)	会话标签(flags)	文件描述符(flags)	文件描述事件(events)
5	172.17.32.158:44025		setex	73327	2	0	N	N	r

取消

确定

会话历史

1. 点击会话列表右上角上 会话历史按钮。

访问来源：

支持左前缀匹配

持续时间：

请选择

单位（秒）

空闲时间：

请选择

单位（秒）

重置

查询

kill所选会话

kill会话历史

提示：

会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)  ↕	空闲时间(idle)  ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	39604	172.17.32.158:30489		setex	1秒	1秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	74947秒	48秒	0	N	12

2. 在弹出的会话历史窗口中，可以接近一小时、近24小时、近7天、近30天、自定义时间范围进行筛选。

kill会话历史记录

近一小时

近24小时

近7天

近30天

2023-08-01 16:17:09

—

2023-08-31 16:17:09

📅

会话ID(id)	访问来源(addr)	命令(cmd)	持续时间(age)	任务执行状态	任务执行结果
5	172.17.32.158:44025	setex	74939	SUCCESS	

关闭

会话重置。

1. 在列表右上角，点击充值按钮，可重置所有条件进行查询。

会话详情

访问来源:

持续时间: 

请选择

 单位 (秒)

空闲时间: 

请选择

 单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示： 会话详情通过命令CLIENT LIST采集，字段具体含义请参考Redis文档：<https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ⬆	空闲时间(idle) ⬆	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

会话统计

点击诊断类型的会话统计按钮，出现回话统计页面，分别用户维度统计、访问来源维度统计、数据库维度统计三部分统计列表及图表数据。

用户维度统计

字段包含用户、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时长（秒）。

产品服务 / 云数据库 Redis-实例列表 / SmartDBA-会话管理

主节点: scs-b...

redis\_244827...

数据更新时间: 2023-08-31 16:39:57

诊断类型: 

会话详情

会话统计

会话拓扑

用户维度统计

用户	活跃会话数	总会话数	活跃会话平均时长 (秒)	活跃会话总时长 (秒)
0	2	0	0	

用户活跃会话数占比

访问来源维度统计

字段包含来源IP、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时（秒）。

访问来源维度统计

来源IP	活跃会话数	总会话数	活跃会话平均时长 (秒)	活跃会话总时长 (秒)
172.17.32.158	0	1	0	0
172.17.32.27	0	1	0	0

访问源活跃会话数占比

访问来源维度统计

字段包含数据库、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时（秒）。

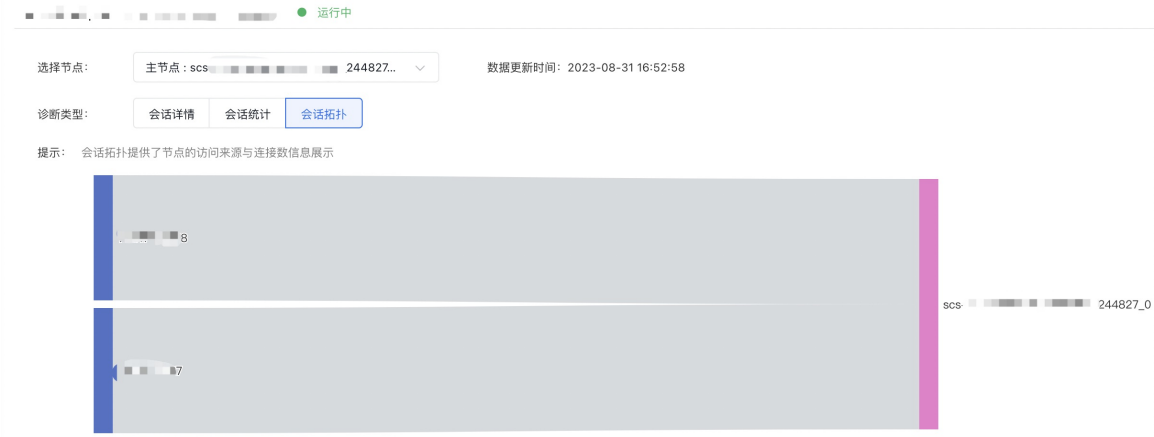
数据库维度统计

数据库	活跃会话数	总会话数	活跃会话平均时长 (秒)	活跃会话总时长 (秒)
0	0	2	0	0

数据库活跃会话数占比

会话拓扑

点击诊断类型的会话拓扑按钮，打开会话拓扑图页面，用户可以通过切换节点ID方式来查看相应实例的拓扑图。



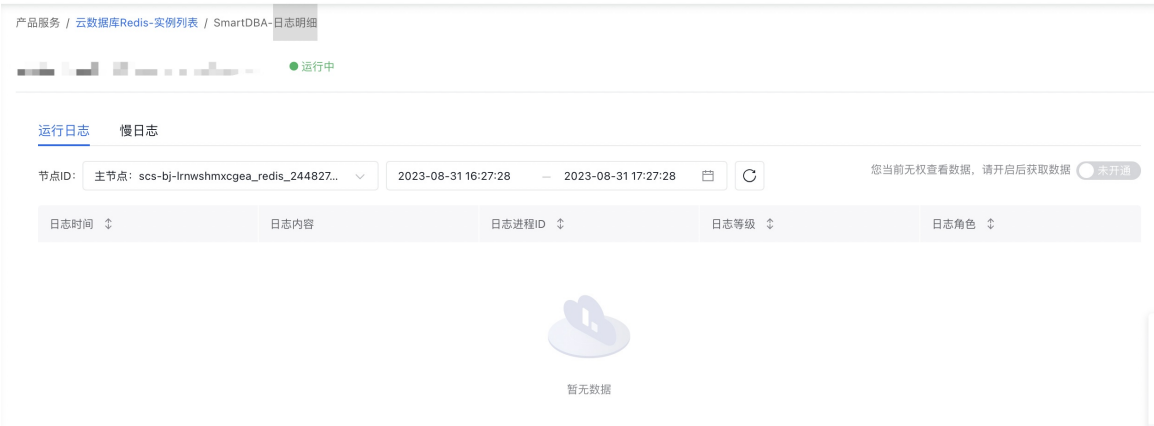
🔗 日志明细

概述

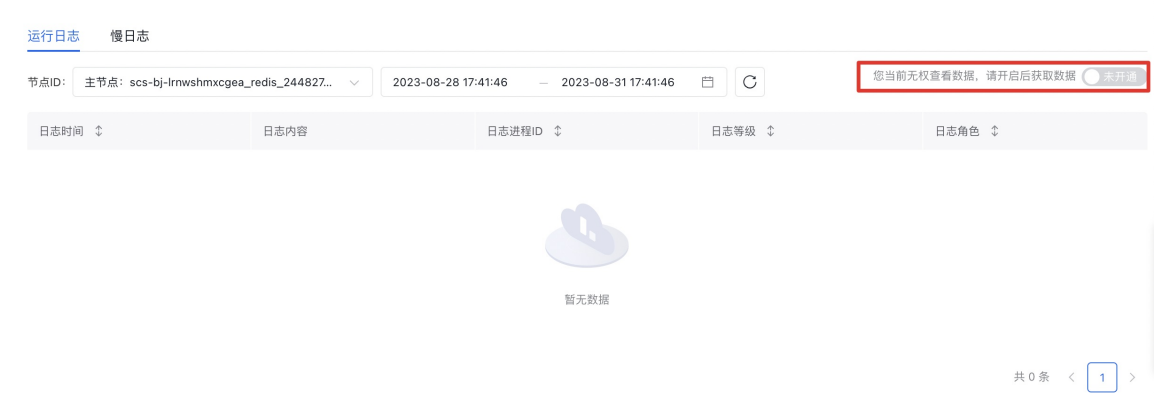
查看当前redis实例运行日志和慢日志列表数据。

运行日志-操作指南

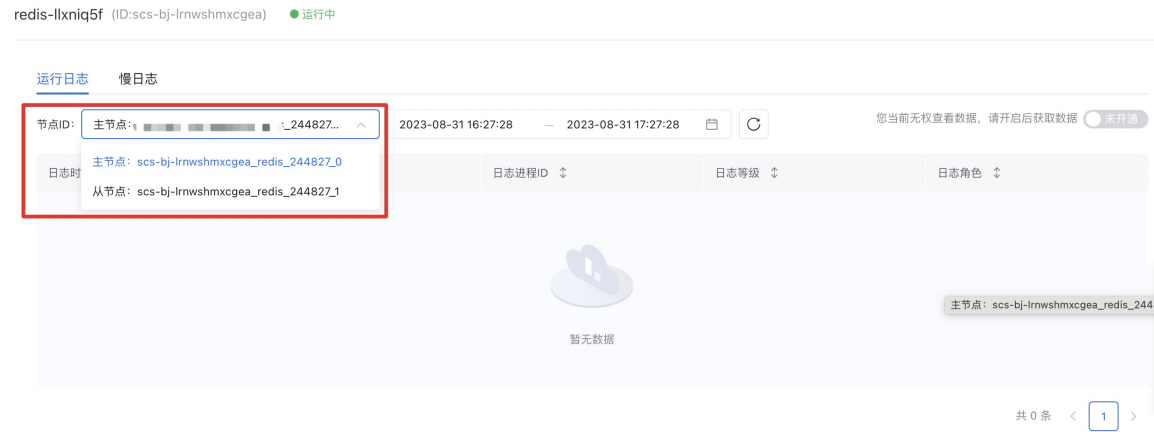
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称/ID，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 **SmartDBA**，菜单展开的**明细日志**菜单，进入明细日志页面，在页面中默认选中**运行日志**页签。



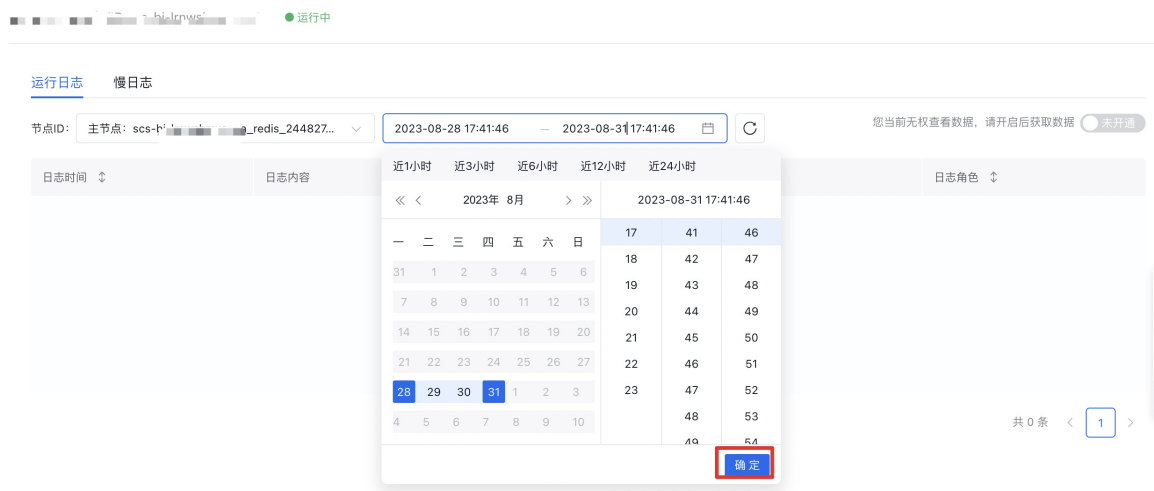
- 4. 如果你有开通运行日志的权限，点击右侧开通查看数据权限，没有则忽略下面操作步骤。



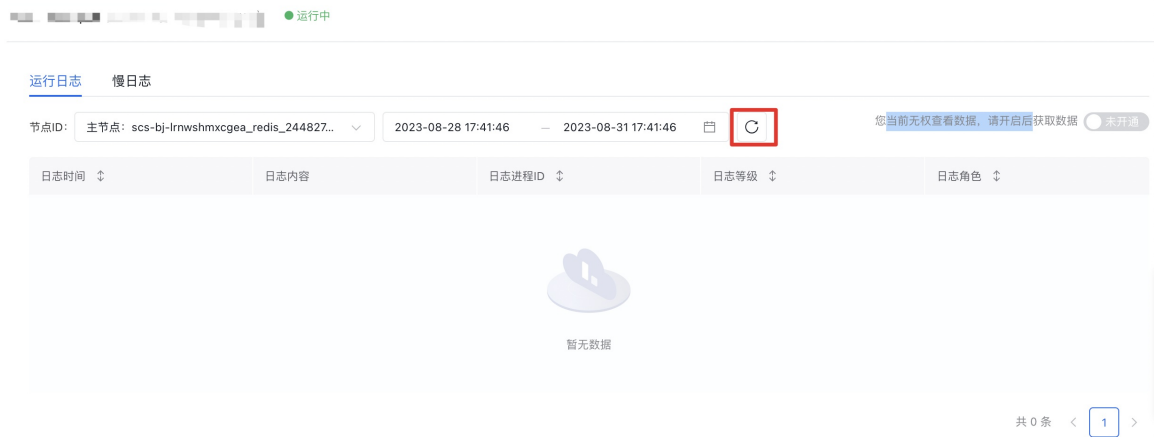
- 5. 在**节点ID**下拉列表框中，选择实例。



6. 在日历框中范围点击右侧日历图标，弹出日历组件,选择开始时间和结束时间，完成后，点击确定按钮，即可刷新列表数据。

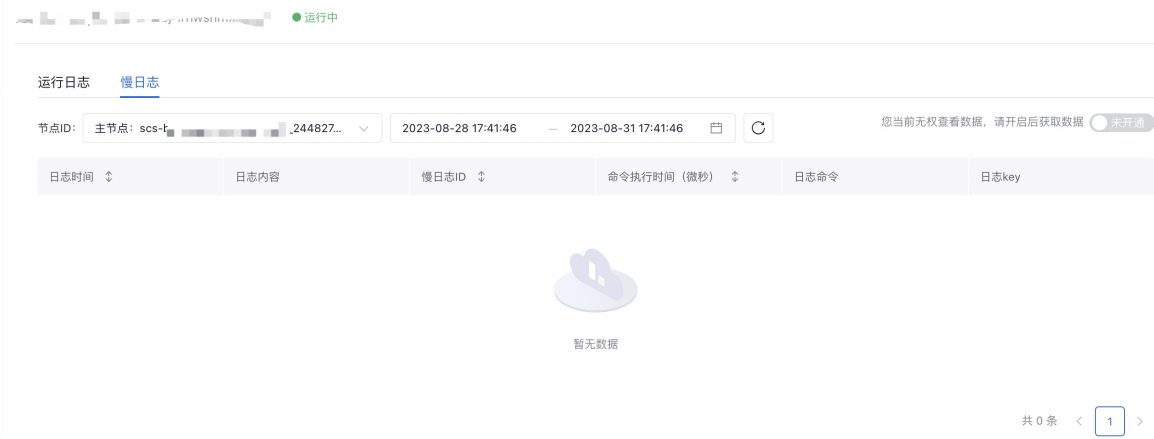


7. 若想刷新列表，则点击日历组件右侧刷新按钮，即可刷新列表。若不想刷新列表，此操作忽略。

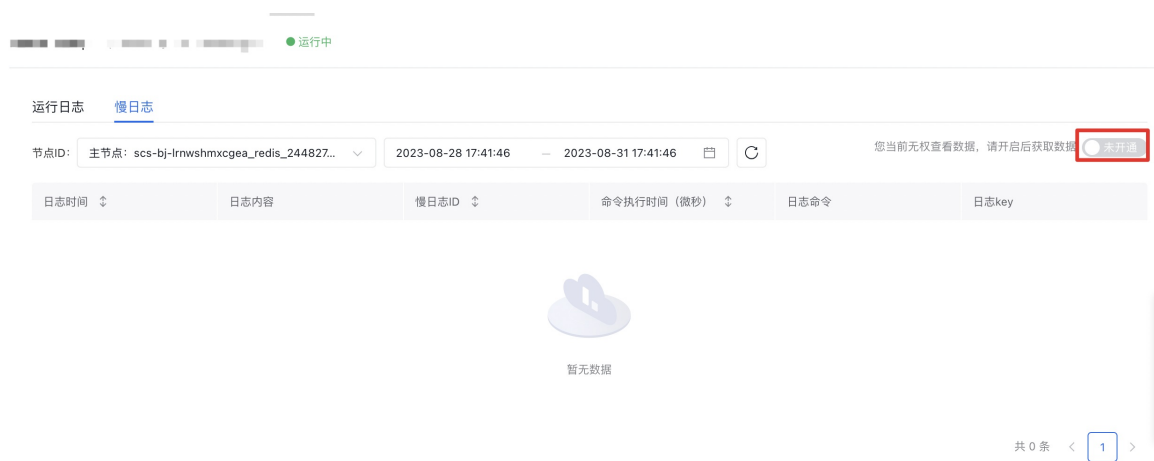


慢日志-操作指南

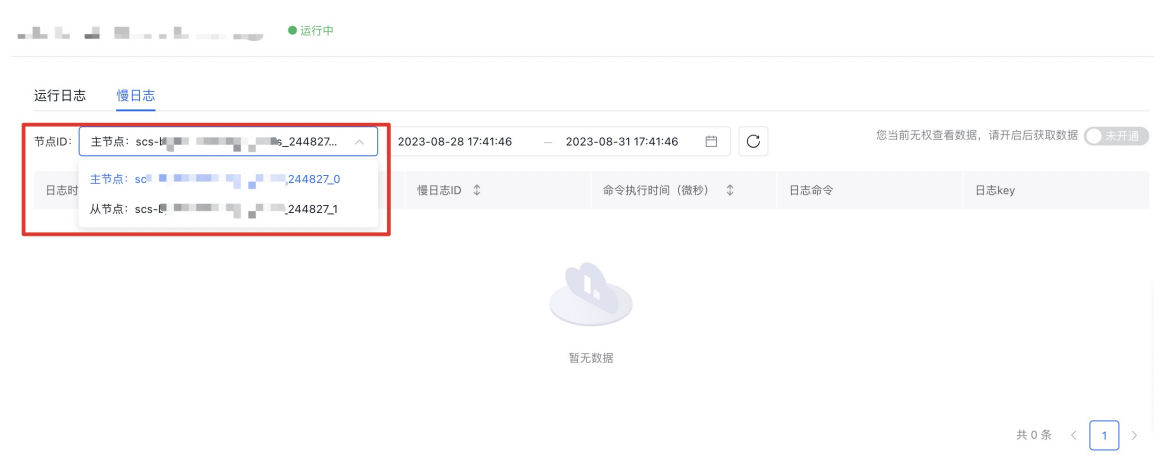
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称/ID，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 SmartDBA，菜单展开的明细日志菜单，进入明细日志页面，在页面中默认选中慢日志页签。



4. 如果你有开通慢行日志的权限，点击右侧开通查看数据权限，没有则忽略下面操作步骤。

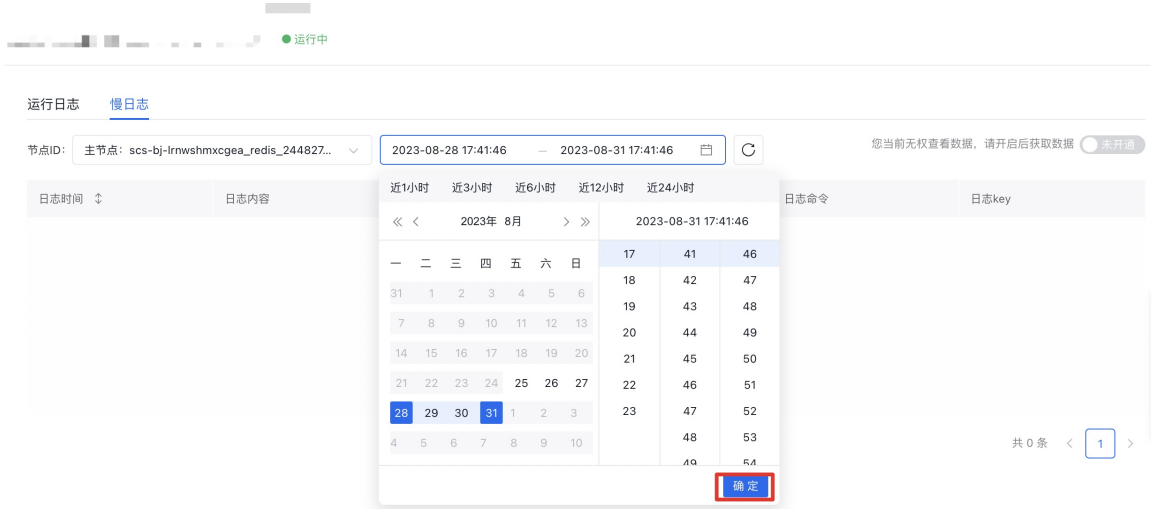


5. 在节点ID下拉列表框中，选择实例。

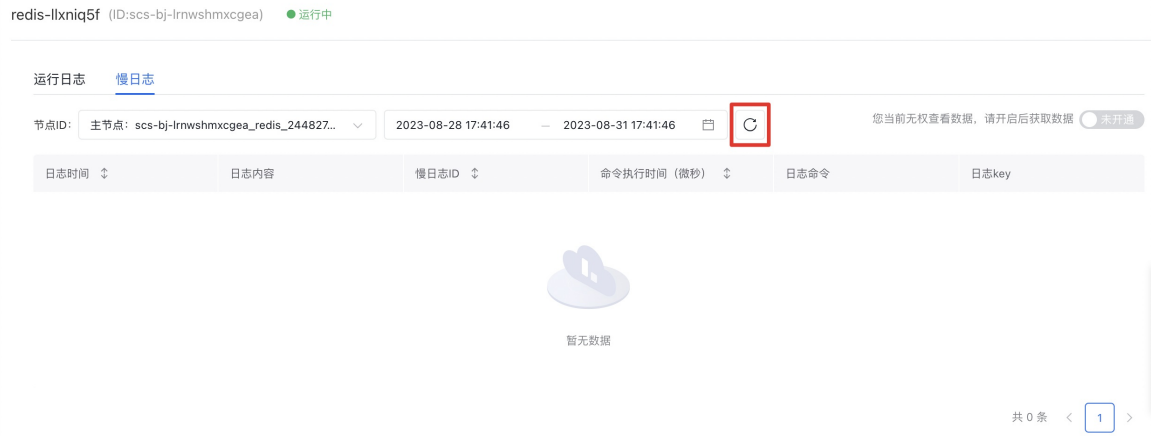


6. 在日历框中范围点击右侧日历图标，弹出日历组件,选择开始时间和结束时间，完成后，点击确定按钮，即可刷新列表数据。





7. 若想刷新列表，则点击日历组件右侧刷新按钮，即可刷新列表。若不想刷新列表，此操作忽略。



集群拓扑

概述

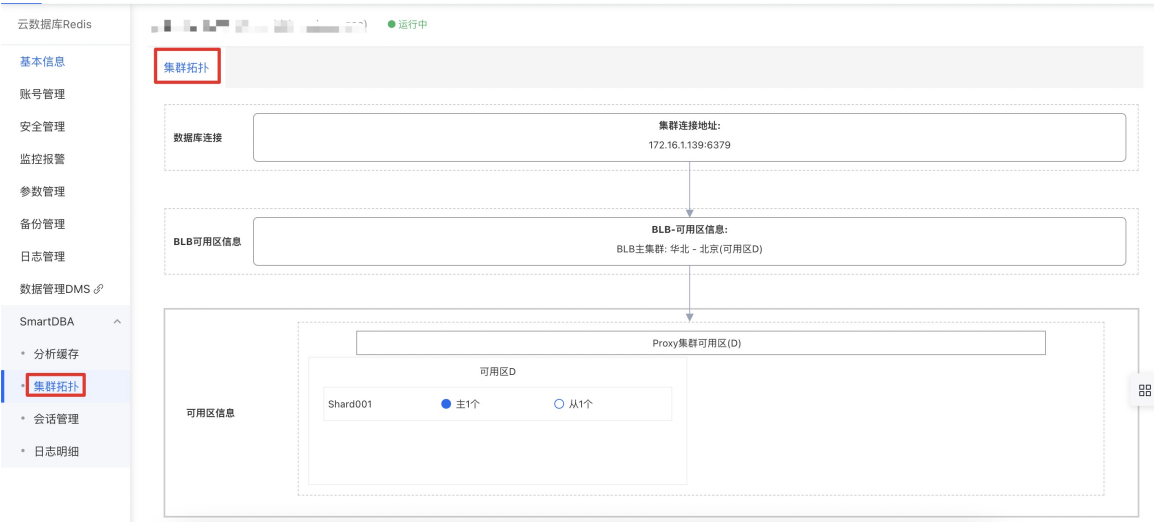
如果您查看实例部署、在网络中的体系结构可通过集群拓普结构查看。

操作步骤

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称/ID，进入实例详情页面。



3. 在实例详情页的左侧导航栏中，点击 SmartDBA，默认选中集群拓扑页签并打开页面。



多用户访问控制

概述

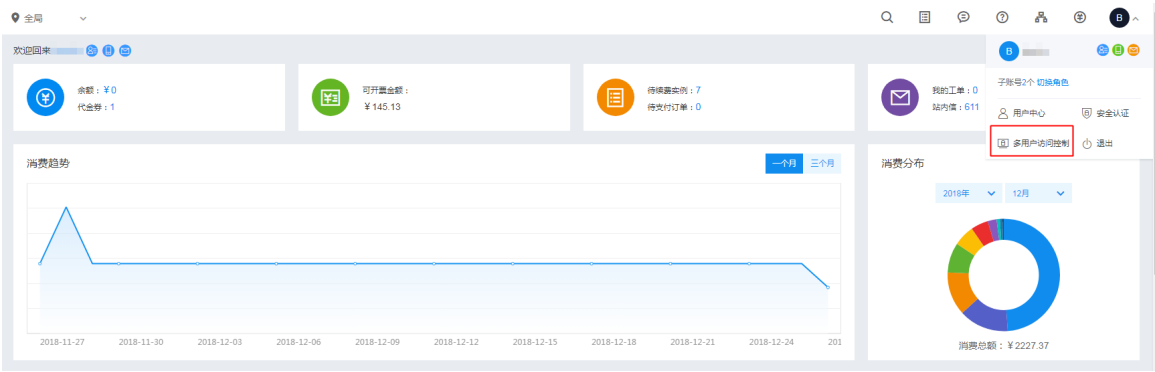
云数据库 Redis 多用户访问控制是指原用户作为子账号的主账号，可以给子账号分配相关产品或实例的操作权限，实现细粒度的权限管理。如给 DBA 等一线操作人员分配产品级或具体到实例级别的运维操作权限，给其他非操作人员分配产品级或实例级的查看权限。

使用场景

- 中大型企业客户：对公司内多个员工授权管理；
- 偏技术型 Vendor 或 SaaS 的平台商：对代理客户进行资源和权限管理；
- 中小开发者或小企业：添加项目成员或协作者，进行资源管理。

创建子用户

1. 主账号用户登录后在控制台选择“多用户访问控制”进入用户管理页面。



2. 在左侧导航栏点击“用户管理”，在“子用户管理列表”页，点击“新建用户”。
3. 在弹出的“新建用户”对话框中，完成填写“用户名”和确认，返回“子用户管理列表”区可以查看到刚刚创建的子用户。

配置策略

云数据库 Redis 支持“系统策略”和“自定义策略”两种权限策略，分别实现产品级权限和实例级权限控制。“系统策略”能够被直接使用并授权给子用户，“自定义策略”需要由主用户自行创建后才能够使用。

系统策略

系统策略为“产品级权限”，目前包括产品级的完全控制管理、运维操作、只读访问三种权限。权限范围详细解释如下：

权限	权限名称	实例list列表	权限范围
产品线 管理	SCSFullControlAccessPolicy	所有云数据库 Redis 实例资源展示及操作	所有云数据库 Redis 实例所有操作权限
产品线 运维	ScsOperateAccessPolicy	所有云数据库 Redis 实例资源展示及操作	所有云数据库 Redis 实例所有运维权限（不包括创建实例、配置变更、续费、Tag、计费变更、释放实例）
产品线 只读	ScsReadAccessPolicy	所有云数据库 Redis 实例资源（操作列无权限）	所有云数据库 Redis 实例详情查看、监控

自定义策略

自定义策略为“实例级权限”，与系统策略不同，只对选定的实例生效。

主用户可以通过点击“策略管理>创建策略”添加自定义策略来进行实例级权限控制，自定义策略的添加有“按策略生成器创建”、“按标签创建”和“按策略语法创建”三种方式，用户可以根据具体的权限设置修改策略内容。具体配置方法可以参见[创建自定义策略](#)。

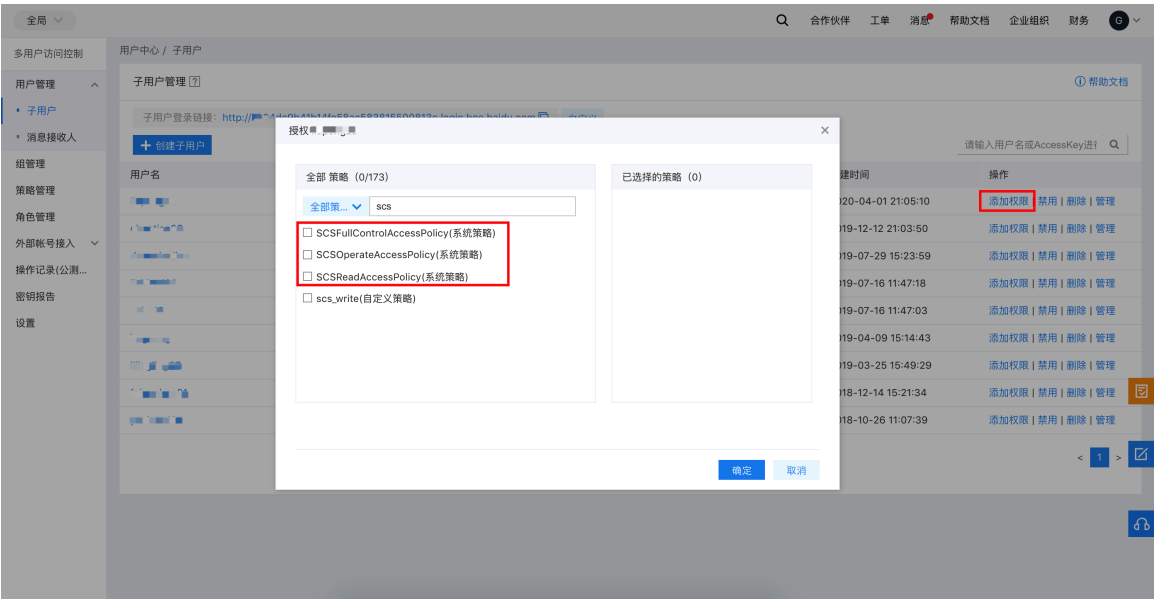
注意：目前每条权限只能选择单个区域进行实例配置

权限范围详细解释如下：

权限	实例list列表	权限范围
实例级 管理	所选云数据库 Redis 实例资源展示及操作	所选云数据库 Redis 实例所有操作权限
实例级 运维	所选云数据库 Redis 实例资源展示及操作	所选云数据库 Redis 实例所有运维权限（不包括创建实例、配置变更、续费、Tag、计费变更、释放实例）
实例级 只读	所选云数据库 Redis 实例资源（操作列无权限）	所选云数据库 Redis 实例详情查看、监控

子用户授权

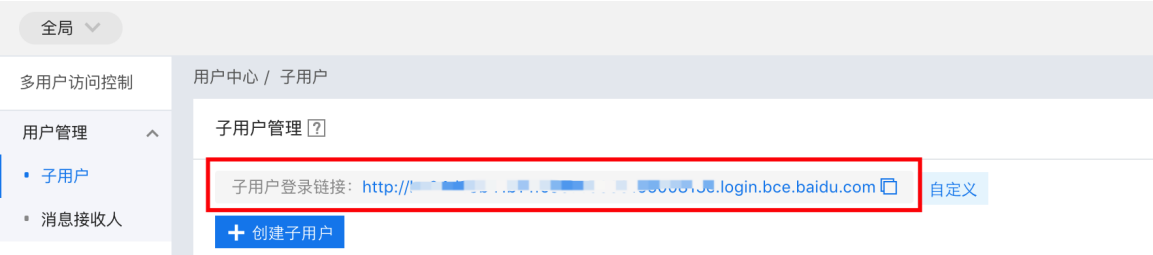
在“用户管理->子用户管理列表页”的对应子用户的“操作”列选择“添加权限”，并为用户选择系统权限或自定义策略进行授权。



说明：如果在不修改已有策略规则的情况下修改某子用户的权限，只能通过删除已有的策略并添加新的策略来实现，不能取消勾选已经添加过的策略权限。

子用户登录

主账号完成对子用户的授权后，可以将链接发送给子用户；子用户可以通过 IAM 用户登录链接登录主账号的管理控制台，根据被授权的策略对主账户资源进行操作和查看。



说明：主用户账号与子用户账号单独配置实例配额，如果您希望提高子用户的实例配额，请在工单中提供子用户账号等信息。

相关文档

其他详细操作参考：[多用户访问控制](#)。

标签管理

创建标签

概述

百度智能云提供标签管理功能，通过给每个云资源添加标签，从而快速分类和识别管理这些资源。

- 标签：每个标签由键和值两部分组成，标签（键+值）唯一。
- 单个和批量：支持为单个资源设置标签，也可以批量为云资源创建标签。

限制

每个用户最多可以创建200个标签。

创建标签

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页。
2. 勾选一个或多个实例，点击**编辑标签**按钮。



3. 在弹出的对话框中，输入自定义**标签键**和**标签值**，注意每个实例可以有多个标签键，但每个标签的键必须唯一，值可以为空，不填写。



- 4. 点击**确定**按钮，完成标签创建。
- 5. 创建完成后，可以在实例列表页目标实例的**标签**列查看已创建的标签。



注意事项

- 批量编辑标签只能添加标签不能管理已有标签。注意标签键设定时区分大小写。
- 批量创建标签时，无法查看已添加的单个实例标签。
- 如果修改某个标签，需要先确定解绑标签后再重新设置。
- 标签创建完成后可以在左侧导航的**标签管理**中查看已创建的所有标签，及各标签绑定的资源数。

删除标签

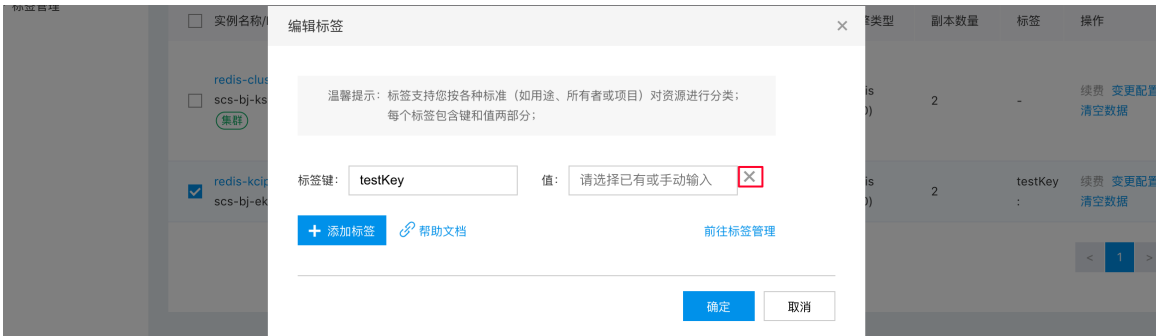
概述

若实例无须使用标签，您可以将实例的标签删除，以下将介绍如何删除标签。

操作指南

方法一：实例列表页面删除标签

1. 登录**云数据库 Redis** 管理控制台，进入实例列表页。
2. 勾选一个或多个实例，点击**编辑标签**按钮。
3. 在弹出对话框中，点击标签右侧的 **×**，删除后，点击**确定**即可删除该标签。



方法二：标签管理页面删除标签

1. 登录 [云数据库 Redis](#) 管理控制台，进入实例列表页。
2. 选择页面左侧导航栏中的**标签管理**，进入标签管理页面。



3. 在标签列表页面找到目标**标签键**，点击 [>](#)，可以看到标签键的详细信息，选择一个或多个要删除的标签值，点击**批量删除标签值**按钮。



4. 在弹出的确认窗中，点击**确定**按钮，进行删除。

🔗 资源账单

概述

资源账单是记录每个资源的生命周期与消费相关的数据的功能，方便用户以资源为维度查看资源及产品的消费数据。

查看账单

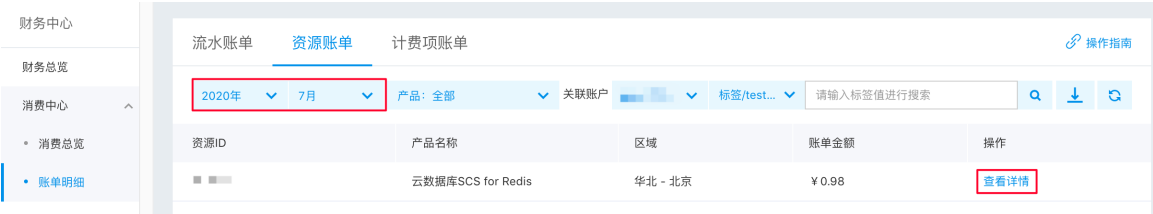
1. 登录 [云数据库 Redis](#) 管理控制台，进入实例列表页。
2. 选择页面左侧导航栏中的**标签管理**，进入标签管理页面。



3. 在标签列表页面找到目标**标签键**，点击 [>](#)，可以看到标签键的详细信息，勾选一个标签，在操作栏点击**查看账单**。



4. 进入资源账单页面，选择需要查看账单的月份，会显示当前标签资源的消费信息。还可以通过组合筛选条件执行标签搜索，按标签键和值进行选择其他标签的消费详情，并下载相关信息。



5. 点击查看详情，可以查看资源 ID、产品名称、账单金额以及账单的明细等。

根据标签筛选实例

概述

使用百度智能云标签管理功能给每个云资源添加标签后，您可以在实例列表根据标签筛选出绑定的资源进行管理。

操作步骤

按以下步骤在控制台上根据标签筛选实例：

- 1. 登录 [云数据库Redis管理控制台](#)，进入实例列表页。
- 2. 实例列表页标签字段可以查看每个实例绑定的标签。



- 3. 如果需要单独筛选出某个标签绑定的全部实例，请点击标签筛选框，在弹出的对话框中选择目标标签键和标签值，进行搜索。
- 注意：筛选结果和输入的筛选条件是完全匹配的，如图实例绑定redis:test2标签，筛选时需要完整输入redis:test2标签进行筛选。



- 4. 如需清除筛选条件，请点击筛选框  图标。



备份管理

备份管理

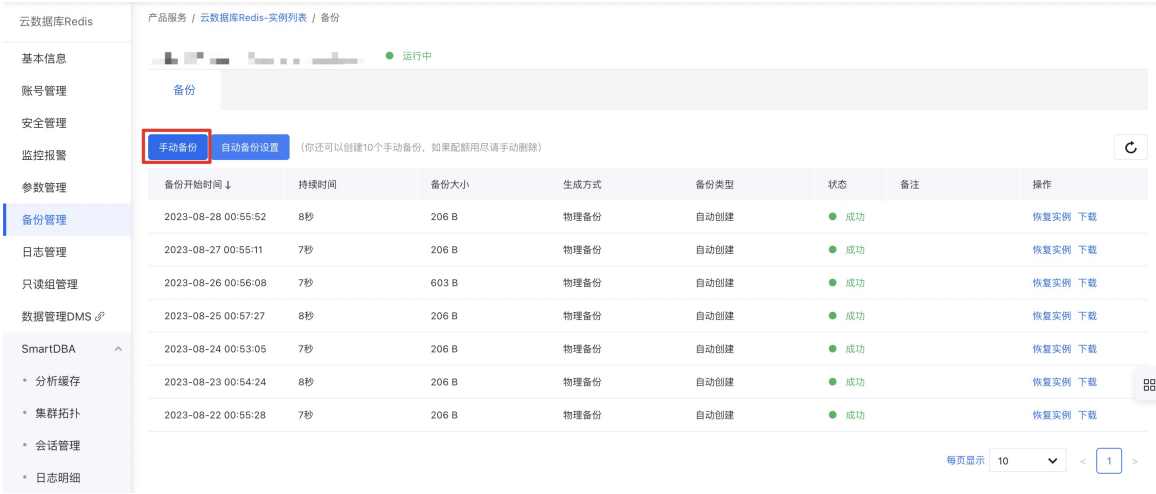
**概述** 云数据库 Redis 提供数据库备份，支持自动备份和手动备份两种方式。

限制条件

- 手动备份最多支持创建10个。
- 自动备份无数量限制，最多保留15天。
- 手动备份与自动备份均免费。

手动备份

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击备份管理。
4. 在备份管理页面，点击手动备份按钮。



5. 弹出备份对话框，输入备注信息，点击确定按钮。



手动备份



备份可能需要数分钟，期间无法进行实例管理操作

备注：

请输入备注信息，最多256个字符

0/256

确定

取消

6.开始备份的时候，实例状态显示为**备份中**，备份完成后实例状态显示为**运行中**。

产品服务 / 云数据库Redis-实例列表 / 备份

● 保存成功 (4s) X

● 备份中

备份

手动备份 自动备份设置 (你还可以创建9个手动备份，如果配额用尽请手动删除)

备份开始时间 ↓	持续时间	备份大小	生成方式	备份类型	状态	备注	操作
列表加载中，请稍候...							

每页显示 10 < 1 >

自动备份

- 1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 2. 点击目标**实例名称**，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 **备份管理**。
- 4. 在备份管理页面，点击**自动备份**按钮。

● 运行中

备份

手动备份 自动备份设置 (你还可以创建10个手动备份，如果配额用尽请手动删除)

备份开始时间 ↓	持续时间	备份大小	生成方式	备份类型	状态	备注	操作
2023-08-28 00:55:52	8秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-27 00:55:11	7秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-26 00:56:08	7秒	603 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-25 00:57:27	8秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-24 00:53:05	7秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-23 00:54:24	8秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载
2023-08-22 00:55:28	7秒	206 B	物理备份	自动创建	● 成功		恢复实例 下载

每页显示 10 < 1 >

5. 弹出备份对话框，选择自动时间（执行时间），重复日期、及备份保留时长，选择完成后，点击**确定**按钮。

自动备份设置

\* 自动时间:

00时

52分

10秒

重复日期:

☒ 周日

☒ 周一

☒ 周二

☒ 周三

☒ 周四

☒ 周五

☒ 周六

\* 保留时长:

保留

7

天后自动删除 (可选范围为1-15天)

确定

取消

6.开始备份的时候，实例状态显示为**备份中**，备份完成后实例状态显示为**运行中**

产品服务 / 云数据库Redis-实例列表 / 备份

● 备份中

保存成功 (4s) X

备份

手动备份

自动备份设置

(你还可以创建9个手动备份。如果配额用尽请手动删除)

刷新

备份开始时间 ↓	持续时间	备份大小	生成方式	备份类型	状态	备注	操作
列表加载中，请稍候...							

每页显示

10

<

1

>

相关API [修改备份策略](#)

🔗 备份恢复

概述

备份列表显示该实例已经备份的文件，您可以下载指定备份文件以及使用该备份文件进行数据恢复。

限制条件

- **备份恢复**支持原实例覆盖与克隆实例两种方式。
- **原实例覆盖** 4.0，5.0版本不支持。
- 加入热活实例组的实例不支持**原实例覆盖**。

操作步骤

按以下步骤进入备份列表页面：

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标**实例名称**，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击**备份管理**，进入到备份列表页面。

下载

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标**实例名称**，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击**备份管理**，进入到备份列表页面。

4.在备份列表页面，点击下载按钮。

备份

手动备份

自动备份设置

(你还可以创建9个手动备份, 如果配额用尽请手动删除)

备份开始时间 ↓	持续时间	备份大小	生成方式	备份类型	状态	备注	操作
2023-08-28 17:58:43	8秒	206 B	物理备份	手动创建	成功		恢复实例 删除 下载
2023-08-28 00:55:52	8秒	206 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-27 00:55:11	7秒	206 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-26 00:56:08	7秒	603 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-25 00:57:27	8秒	206 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-24 00:53:05	7秒	206 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-23 00:54:24	8秒	206 B	物理备份	自动创建	成功		恢复实例 下载
2023-08-22 00:55:28	7秒	206 B	物理备份	自动创建	成功		恢复实例 下载

每页显示

10

<

1

>

5.在弹出的下载对话框中，点击下载按钮或复制链接去浏览器下载。

下载

链接地址：

https://bj.bcebos.com/scs-backup-rdb-bucket-bj/expiration/41870\_239012\_f14a52c1-e4ef-4eb4-bb4d-3dae40a3397d?authorization=bce-auth-v1%2Ffc0d28fa00f342e49dc6a779778dba3b%2F2023-08-

复制链接

链接有效时间：

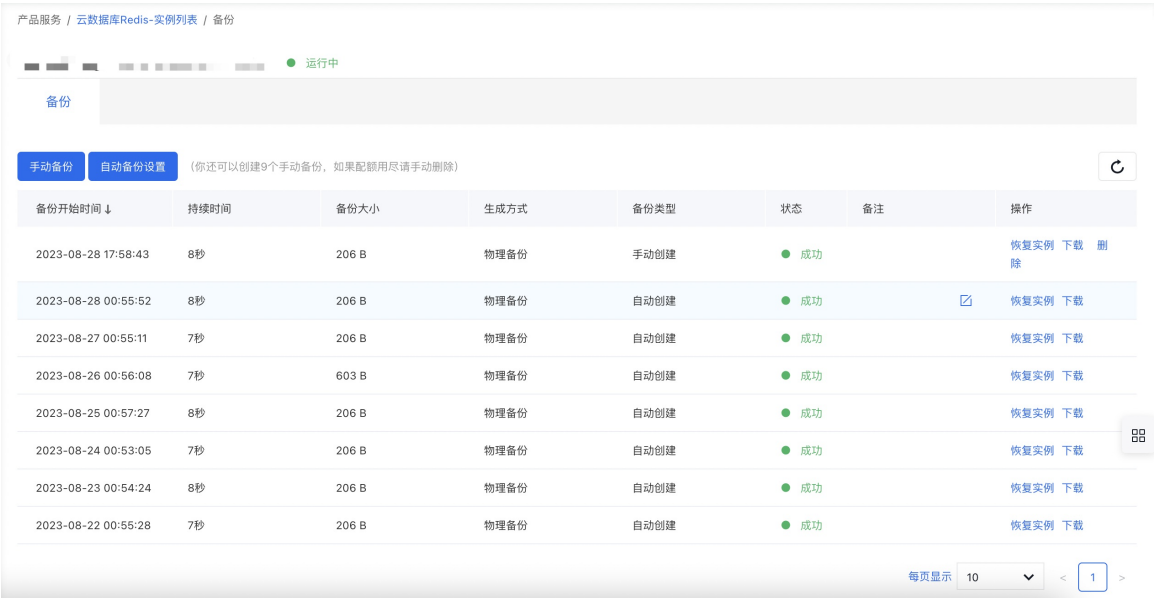
30分钟

直接下载

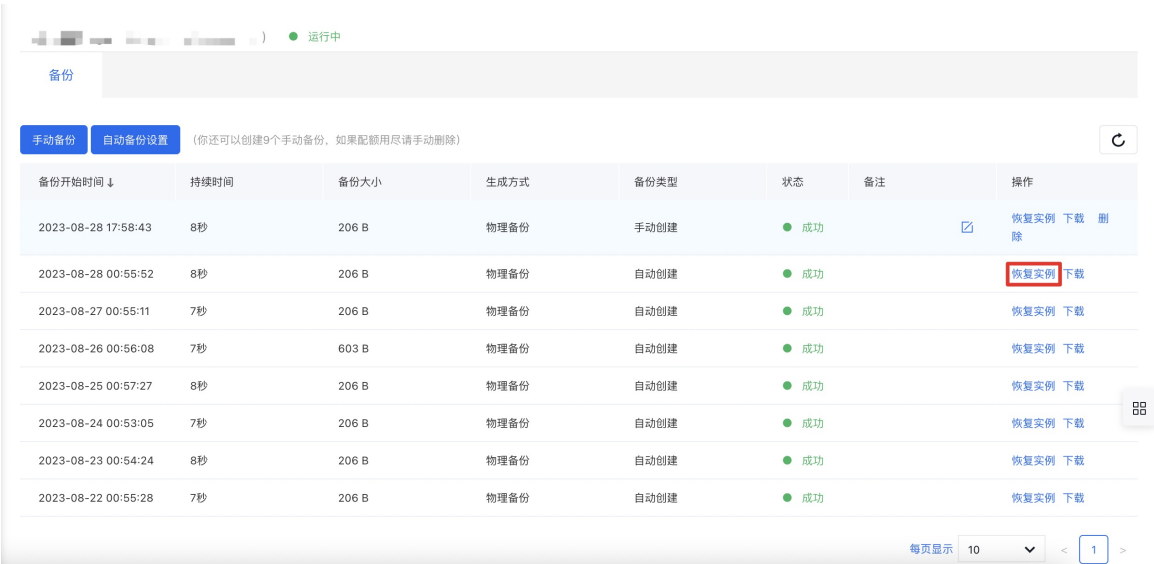
取消

恢复实例

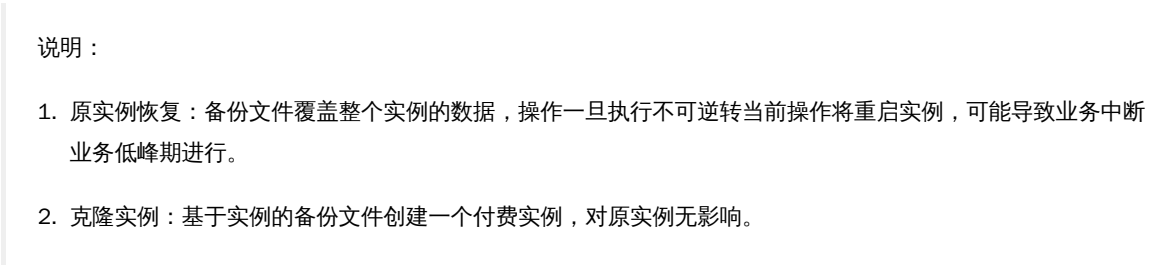
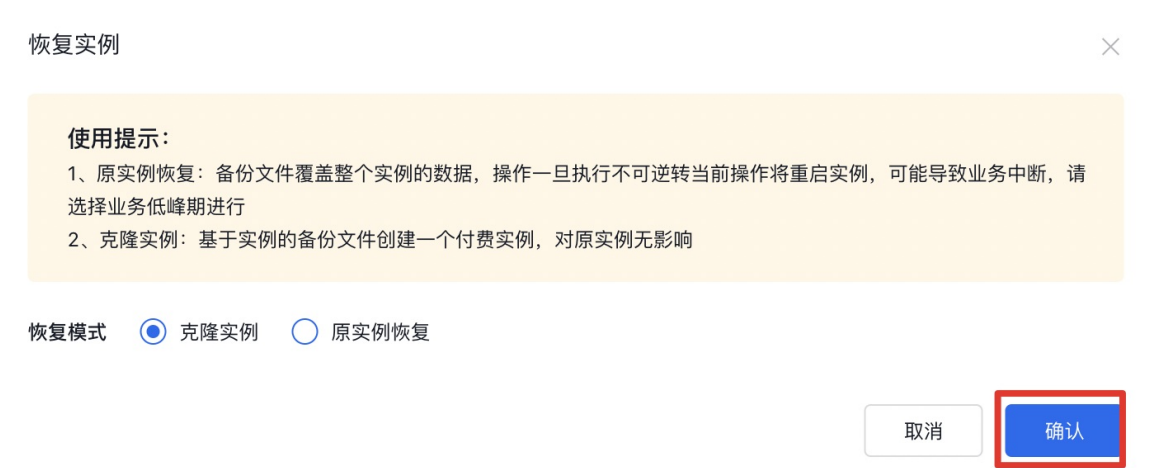
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击备份管理，进入到备份列表页面。



4.在备份列表页面，点击恢复实例按钮。



5. 在弹出恢复实例的对话框，选择恢复模式（克隆实例或原有实例），选择完成后，点击确定按钮。



3. 恢复模式：原实例恢复、克隆实例。 [相关API](#) [查看备份列表](#) [获取备份信息](#)

Redis持久化与备份恢复概述

概览

为避免Redis内存中的数据在服务器宕机或其它故障情况下丢失，云数据库Redis提供了不同的数据持久化策略以及备份恢复策略，通过合理选择和配置，实现数据的持久化和恢复，提高系统的可用性和可靠性。

持久化策略

RDB持久化

- 在指定的时间间隔内，生成RDB文件，将内存中的数据快照写入磁盘，数据恢复时将快照文件直接再读到内存。
- 您可以在实例详情-备份管理中根据业务需求进行自动备份策略的设置，也可以发起手动备份生成RDB备份文件。
- Redis的RDB持久化策略默认每天备份一次，您可以根据业务需求修改自动备份策略，也可以手动发起临时的备份。

AOF持久化

将Redis服务器所执行的所有写操作命令（例如 `SET KEY VALUE`）以追加的方式写入到AOF文件中，当服务器启动时会把AOF文件中的命令重新执行一遍来恢复数据。

创建的Redis实例默认开启AOF持久化，您可以在实例详情-参数管理中通过修改`appendonly`参数进行持久化策略设置，例如[关闭AOF落盘](#)。

**备份与恢复** 云数据库 Redis 实例提供基于RDB持久化、AOF文件的数据备份与恢复。

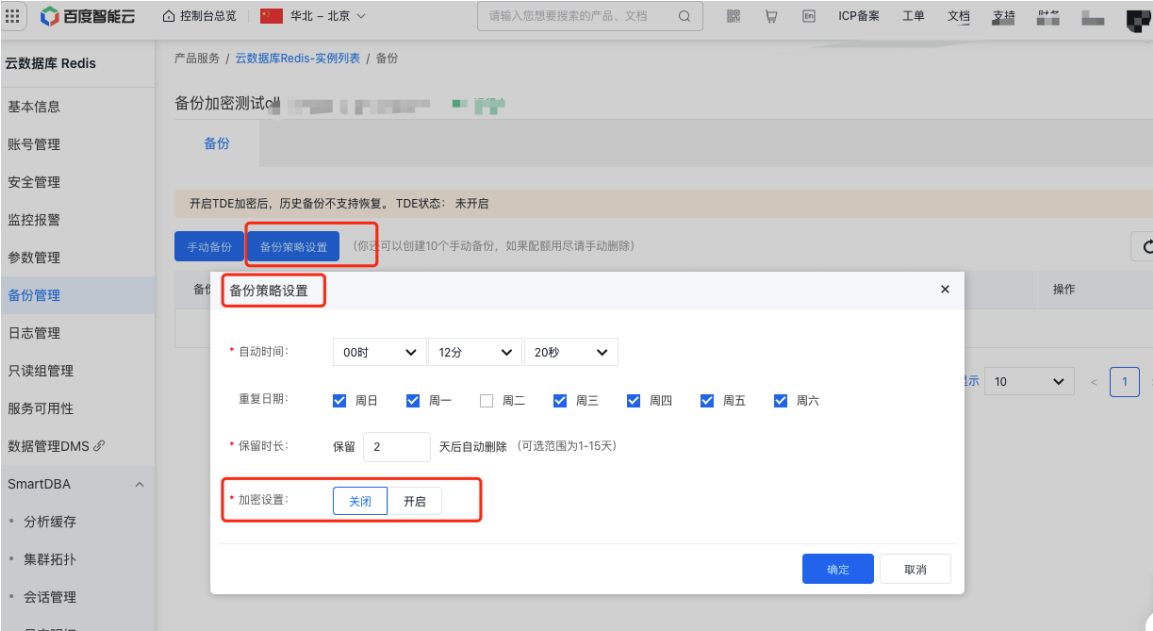
类别	方案	说明
数据备份	<a href="#">自动备份</a> <a href="#">或手动备份</a>	提供RDB数据备份，按照默认的策略支持自动备份和手动备份两种方式。
	<a href="#">下载备份文件</a>	Redis备份文件支持保留1~15天，如果有长时间备份存档的需求，您可以下载RDB备份文件到本地。
备份加密	<a href="#">备份加密</a>	在Redis上使用备份时（克隆恢复、原地覆盖恢复）不需要手动解密，后端会自动解密后执行相关操作（例如恢复数据）。但是如果下载加密的备份到本地，需要使用密钥手动解密才能使用备份。
数据恢复	<a href="#">恢复实例</a>	使用实例备份文件进行原实例模式、克隆实例模式恢复数据。
按时间点恢复	<a href="#">按时间点恢复</a>	使用实例AOF文件采取克隆实例的模式按时间点恢复数据。

备份加密

**\*\*概览\*\*** 为了更好地保证数据的安全性，Redis支持对备份数据进行加密编码。配置加密后，当文件上传时，会通过加密参数来对上传数据执行本地服务端或远程存储加密设置。 **注意事项**

- 所有上传的备份数据都会使用Redis托管的密钥统一进行加密保护。
- 在Redis上使用备份时（克隆恢复、原地覆盖恢复）不需要手动解密，后端会自动解密后执行相关操作（例如恢复数据）。
- 如果下载加密的备份到本地，需要使用密钥手动解密才能使用备份。
- 开通加密后，新产生的备份才会生效。存量的备份不会生效。
- 开通加密后，无法再修改密钥。

- 只有Redis内存型支持备份加密。 操作步骤
- 登录云数据库 Redis 管理控制台，进入实例列表页面。
  - 点击目标实例名称，进入实例详情页面。
  - 在实例详情页的左侧导航栏中，点击备份管理。
  - 在备份管理页面，点击备份策略设置按钮。
  - 弹出备份对话框，加密设置字段选择开启，点击确定按钮。



按时间点恢复数据

概述

开启数据闪回后，在备份文件的保存期内（最多7天），您可以将实例全量Key的数据恢复至某个秒级的时间点（PITR，point-in-time recovery），同时支持选择将数据恢复至新实例。数据闪回功能提供方便快捷、更精细化的数据恢复能力，能最大程度地降低误操作带来的数据损失，保护您的数据安全。

注意事项

- 数据闪回功能仅Redis内存型 6.0、7.0版本支持。
  - 部分存量的 7.0实例（在该功能发布之前创建的）需要先将小版本升级到最新后再使用该功能。否则可能导致备份 AOF功能受影响。
- 开启该功能后将默认打开该实例的AOF参数。
- 可恢复的时间范围为当前时间点至数据闪回功能开启的时间点（最长为7天）。
- 开启数据闪回功能后，如下操作会关闭数据闪回功能或影响可恢复的时间点。
  - 若变配实例的架构（例如从标准架构变配至集群架构等），数据闪回功能将关闭，如需继续使用请重新配置。
  - 若新增或删减了集群架构的分片数，数据闪回功能备份的新、老节点数据可能会不一致，如需继续使用请重新配置。
  - 加入热活实例组的实例不支持数据闪回功能，若实例已开启数据闪回功能，加入热活实例组后将禁用该功能。
  - 已经加入多活实例组的实例在退出实例组时，数据闪回功能将自动关闭。

操作步骤

- 登录云数据库 Redis 管理控制台，进入实例列表页面。

- 2. 点击目标**实例名称**，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击**备份管理**。
- 4. 在备份管理页面，点击**数据闪回**。
- 5. 进入数据闪回后，点击按钮开启**数据闪回功能**。数据闪回功能开启将默认打开AOF参数。



6.数据闪回功能开启（需要一定耗时）后点击**创建闪回**。



7.确认弹窗信息后，点击确认进入克隆实例页面，填写相应的实例信息后下单，新建实例便会根据相应的时间点恢复数据。



只读组管理

只读组管理



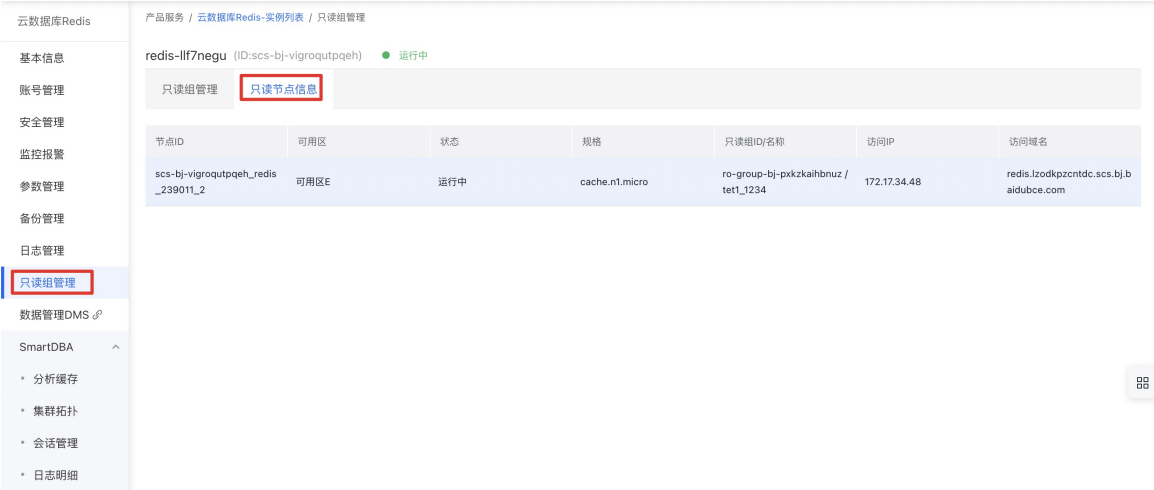
**概述** 云数据库 Redis 只读组包含：只读组管理和只读节点信息。

**注意**

- 只读节点会单独提供读入口，和从节点仅提供灾备功能不能读有差异。

**只读节点信息**

- 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 点击目标**实例名称**，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击**只读组管理**。
- 在备份管理页面，点击**只读节点信息**页签，可查看只读节点信息。



**只读组管理**

- 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 点击目标**实例名称**，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击**只读组管理**。
- 在备份管理页面，点击**只读组管理**页签，可查看只读节点信息。

**任务管理**

**任务列表**

**概述**

本文为您介绍通过任务列表，查看已经执行过的任务，或者正在执行的任务具体进程的方法。

**注意事项**

- 目前任务列表记录了用户触发的所有后台任务，对于每个任务会拆分出对应的关键步骤并返回状态信息供参考。

**操作步骤**

- 登录[云数据库 Redis 管理控制台](#)。
- 在控制台页面的“云数据库 Redis-任务列表”页面，查看**任务列表**。

部署集

实例

热活实例组

多活实例组

硬件待维护实例

回收站

任务列表

参数模板

标签管理

资源管理

配额管理

全部

今天

昨天

近7天

近30天

2023-08-06 00:00:00 - 2024-08-06 15:31:01

实例ID

任务ID	任务类型	实例名称	任务执行状态	任务进展	任务创建时间	任务开始时间	任务完成时间	操作
2531	创建任务	实例名称	● 执行成功	100%	2024-08-06 14:56:20	2024-08-06 14:56:20	2024-08-06 14:57:05	查看详情
2488	创建任务	实例名称	● 执行成功	100%	2024-08-06 11:17:50	2024-08-06 11:17:50	2024-08-06 11:18:38	查看详情
2487	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:51:45	2024-08-06 10:51:45	2024-08-06 10:52:31	查看详情
2485	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:50:30	2024-08-06 10:50:30	2024-08-06 10:52:30	查看详情
2287	重启	实例名称	● 执行成功	100%	2024-08-05 20:26:49	2024-08-05 20:26:49	2024-08-05 20:27:31	查看详情
2286	创建任务	实例名称	● 执行成功	100%	2024-08-05 20:24:54	2024-08-05 20:24:54	2024-08-05 20:25:38	查看详情
1193	配置变更	实例名称	● 执行中	71.43%	2024-08-02 13:20:00	2024-08-02 13:20:00	--	查看详情
1191	配置变更	实例名称	● 执行成功	100%	2024-08-02 13:17:14	2024-08-02 13:17:14	2024-08-02 13:18:03	查看详情
1186	重启	实例名称	● 执行成功	100%	2024-08-02 13:11:19	2024-08-02 13:11:19	2024-08-02 13:12:13	查看详情
1184	配置变更	实例名称	● 执行成功	100%	2024-08-02 13:08:38	2024-08-02 13:08:38	2024-08-02 13:09:19	查看详情

10条/页

< 1 2 3 4 ... 8 >

3.列表上方可进行时间筛选，支持快捷筛选一定时间周期的数据，可以自定义时间筛选。

全部今天昨天近7天近30天2023-08-06 00:00:00 - 2024-08-06 15:38:49

实例ID

任务ID	任务类型	实例名称	任务执行状态	任务进展	任务创建时间	任务开始时间	任务完成时间	操作
2531	创建任务	实例名称	● 执行成功	100%	2024-08-06 14:56:20	2024-08-06 14:56:20	2024-08-06 14:57:05	查看详情
2488	创建任务	实例名称	● 执行成功	100%	2024-08-06 11:17:50	2024-08-06 11:17:50	2024-08-06 11:18:38	查看详情
2487	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:51:45	2024-08-06 10:51:45	2024-08-06 10:52:31	查看详情
2485	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:50:30	2024-08-06 10:50:30	2024-08-06 10:52:30	查看详情
2287	重启	实例名称	● 执行成功	100%	2024-08-05 20:26:49	2024-08-05 20:26:49	2024-08-05 20:27:31	查看详情
2286	创建任务	实例名称	● 执行成功	100%	2024-08-05 20:24:54	2024-08-05 20:24:54	2024-08-05 20:25:38	查看详情
1193	配置变更	实例名称	● 执行中	71.43%	2024-08-02 13:20:00	2024-08-02 13:20:00	--	查看详情
1191	配置变更	实例名称	● 执行成功	100%	2024-08-02 13:17:14	2024-08-02 13:17:14	2024-08-02 13:18:03	查看详情
1186	重启	实例名称	● 执行成功	100%	2024-08-02 13:11:19	2024-08-02 13:11:19	2024-08-02 13:12:13	查看详情
1184	配置变更	实例名称	● 执行成功	100%	2024-08-02 13:08:38	2024-08-02 13:08:38	2024-08-02 13:09:19	查看详情

10条/页

< 1 2 3 4 ... 8 >

4.可输入实例ID与实例名称，查看部分实例的相关操作任务详情。

全部今天昨天近7天近30天2023-08-06 00:00:00 - 2024-08-06 15:41:49

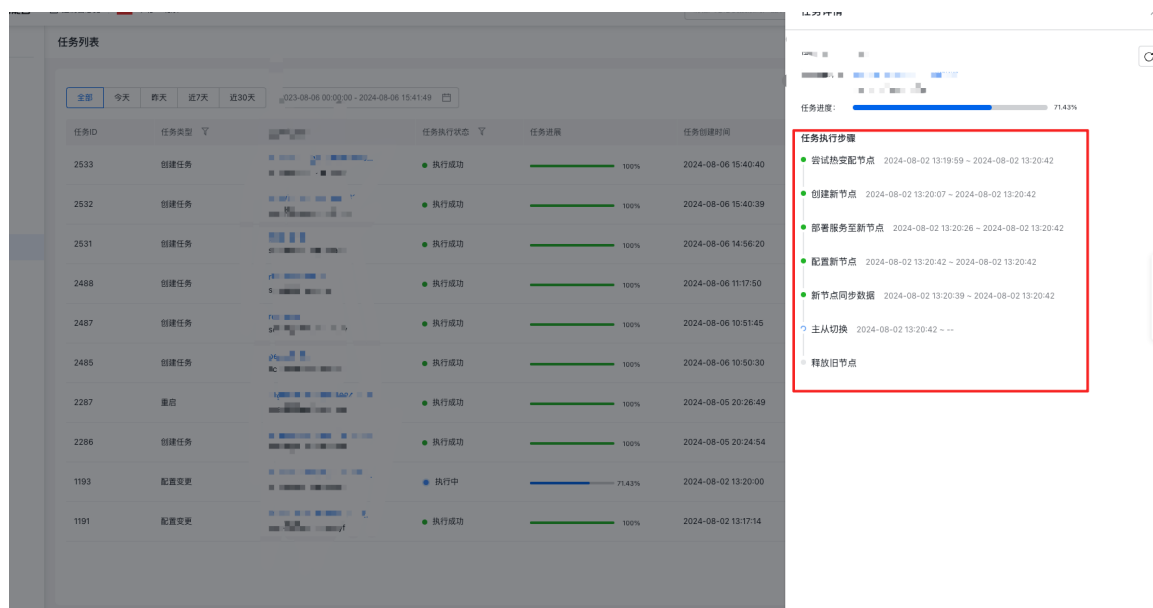
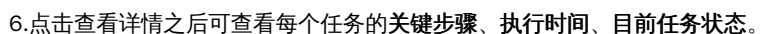
实例ID实例名称

任务ID	任务类型	实例名称	任务执行状态	任务进展	任务创建时间	任务开始时间	任务完成时间	操作
2533	创建任务	实例名称	● 执行中	33.33%	2024-08-06 15:40:40	2024-08-06 15:40:40	--	查看详情
2532	创建任务	实例名称	● 执行成功	100%	2024-08-06 15:40:39	2024-08-06 15:40:39	2024-08-06 15:41:24	查看详情
2531	创建任务	实例名称	● 执行成功	100%	2024-08-06 14:56:20	2024-08-06 14:56:20	2024-08-06 14:57:05	查看详情
2488	创建任务	实例名称	● 执行成功	100%	2024-08-06 11:17:50	2024-08-06 11:17:50	2024-08-06 11:18:38	查看详情
2487	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:51:45	2024-08-06 10:51:45	2024-08-06 10:52:31	查看详情
2485	创建任务	实例名称	● 执行成功	100%	2024-08-06 10:50:30	2024-08-06 10:50:30	2024-08-06 10:52:30	查看详情
2287	重启	实例名称	● 执行成功	100%	2024-08-05 20:26:49	2024-08-05 20:26:49	2024-08-05 20:27:31	查看详情
2286	创建任务	实例名称	● 执行成功	100%	2024-08-05 20:24:54	2024-08-05 20:24:54	2024-08-05 20:25:38	查看详情
1193	配置变更	实例名称	● 执行中	71.43%	2024-08-02 13:20:00	2024-08-02 13:20:00	--	查看详情
1191	配置变更	实例名称	● 执行成功	100%	2024-08-02 13:17:14	2024-08-02 13:17:14	2024-08-02 13:18:03	查看详情

10条/页

< 1 2 3 4 ... 8 >

5.根据任务类型、任务状态进行筛选。



任务列表步骤拆分与状态信息 备注：不同架构的产品具体任务步骤存在一定差异，以下表格仅供参考。

153

重启	- 主从切换   - 重启(旧的)主节点
创建任务	- 任务总进度：50%   - 创建新节点   - 部署服务至新节点   - 配置新节点
升级任务	- 任务总进度：33%   - 升级从节点   - 主从切换   - 升级(旧的)主节点   - 升级代理
副本变更任务	- 任务总进度：33%   - 创建新节点   - 部署服务至新节点   - 配置新节点   - 新节点同步数据
只读副本变更任务	- 任务总进度：33%   - 创建新节点   - 部署服务至新节点   - 配置新节点   - 新节点同步数据
数据恢复任务	- 任务总进度：33%   - 准备数据   - 恢复数据
配置变更	- 任务总进度：33%   - 创建新节点   - 部署服务至新节点   - 配置新节点   - 数据槽重分配

## 导出实例列表

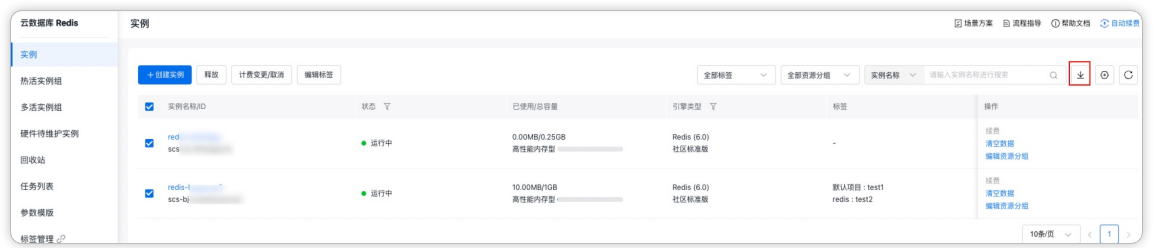
### 概述

本文为您介绍导出云数据库Redis实例列表方法。

### 操作步骤

按以下步骤在控制台上导出实例列表：

1. 登录 [云数据库Redis管理控制台](#)，左上角选择地域信息进行地域切换。
2. 进入指定地域的实例列表页，勾选需要导出的实例，点击右上角按钮。系统会自动将实例信息以CSV格式下载到本地。



3. 您可以在本地使用合适工具打开CSV文件进行查看管理。

## 创建热活实例组

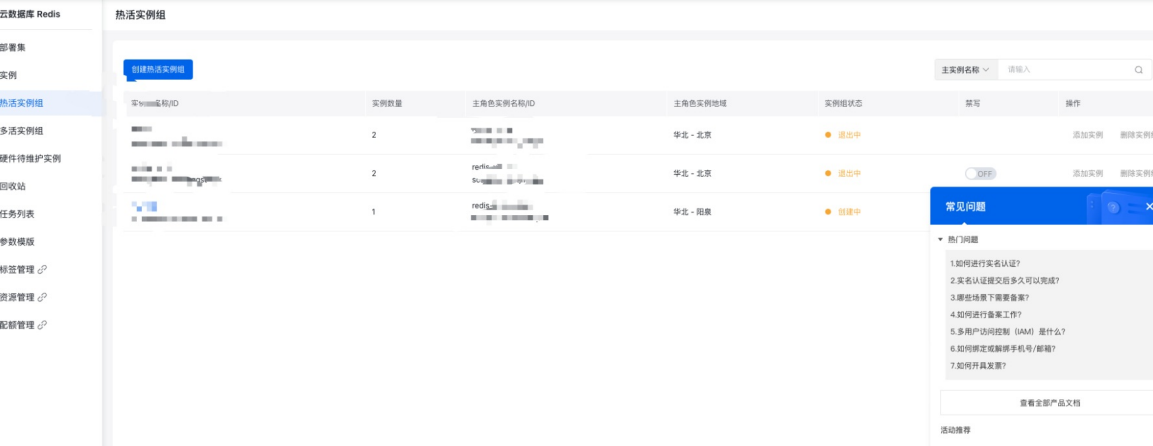
### 概述

百度云Redis提供创建热活实例架构管理的功能，用户可以管理热活实例组。

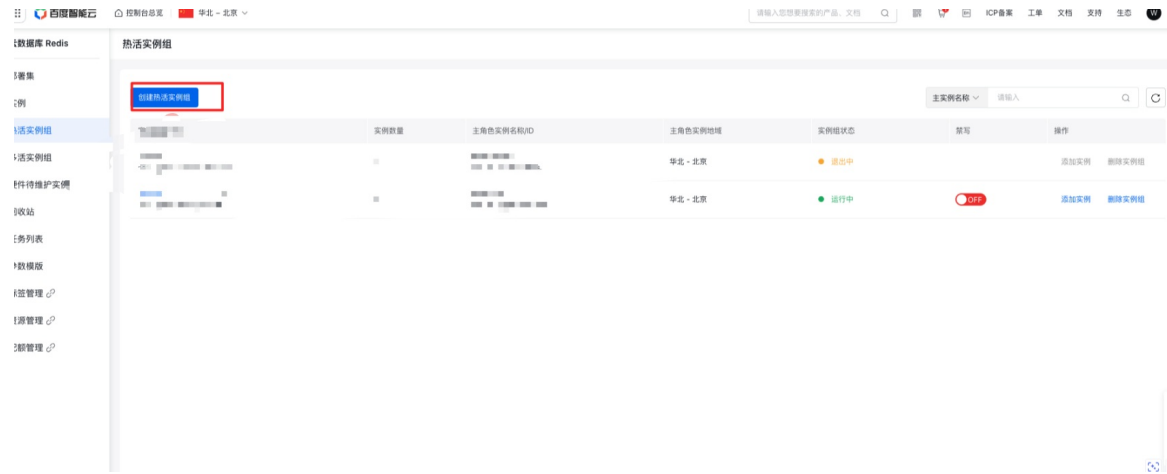
- 跨地域热活实例组是由多个Redis集群实例组成的高可用集群，并且允许以上实例分布在不同的地域和可用区，因此能够实现跨地域容灾。
- 为了便于管理，我们会以主实例为单位，进行热活实例组的相关操作。
- 热活组中的标准版实例不支持配置白名单。

### 操作步骤

1. 登录 [云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库Redis-热活实例组”页面。



- 3.创建热活实例组。



主角色信息

地域: 华北 - 北京

实例ID: [ID]

实例组名称:

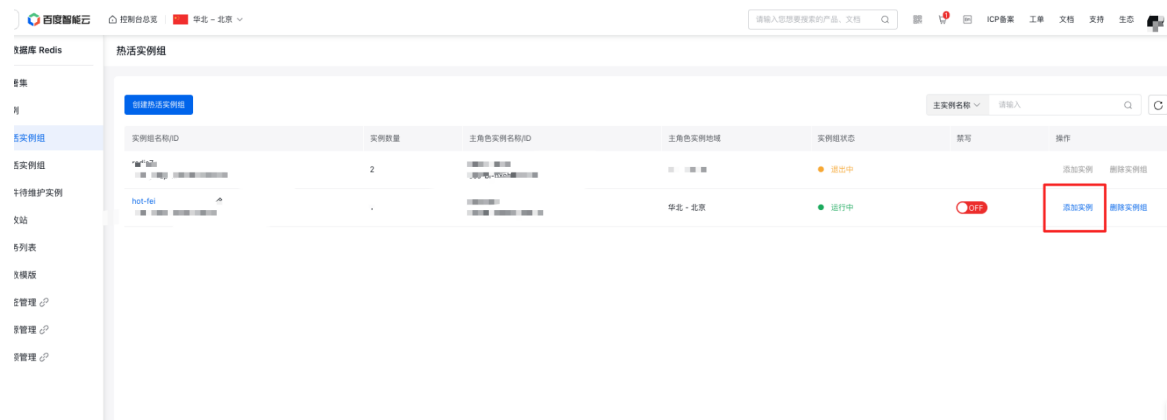
主角色信息: 实例名称: [名称] 企业版Redis (6.0) 副本数量: 2

主角色检查:

实例状态是否为运行中	通过
实例引擎类型是否为社区标准版或者企业集群版	通过
实例节点规格是否不大于16GB	通过
实例是否未加入实例组	通过
实例是否未关联VPC侧安全组	通过
实例是否未开启TDE加密功能	通过
是否具备相同的hashtag参数值	通过
是否设置了密码	通过

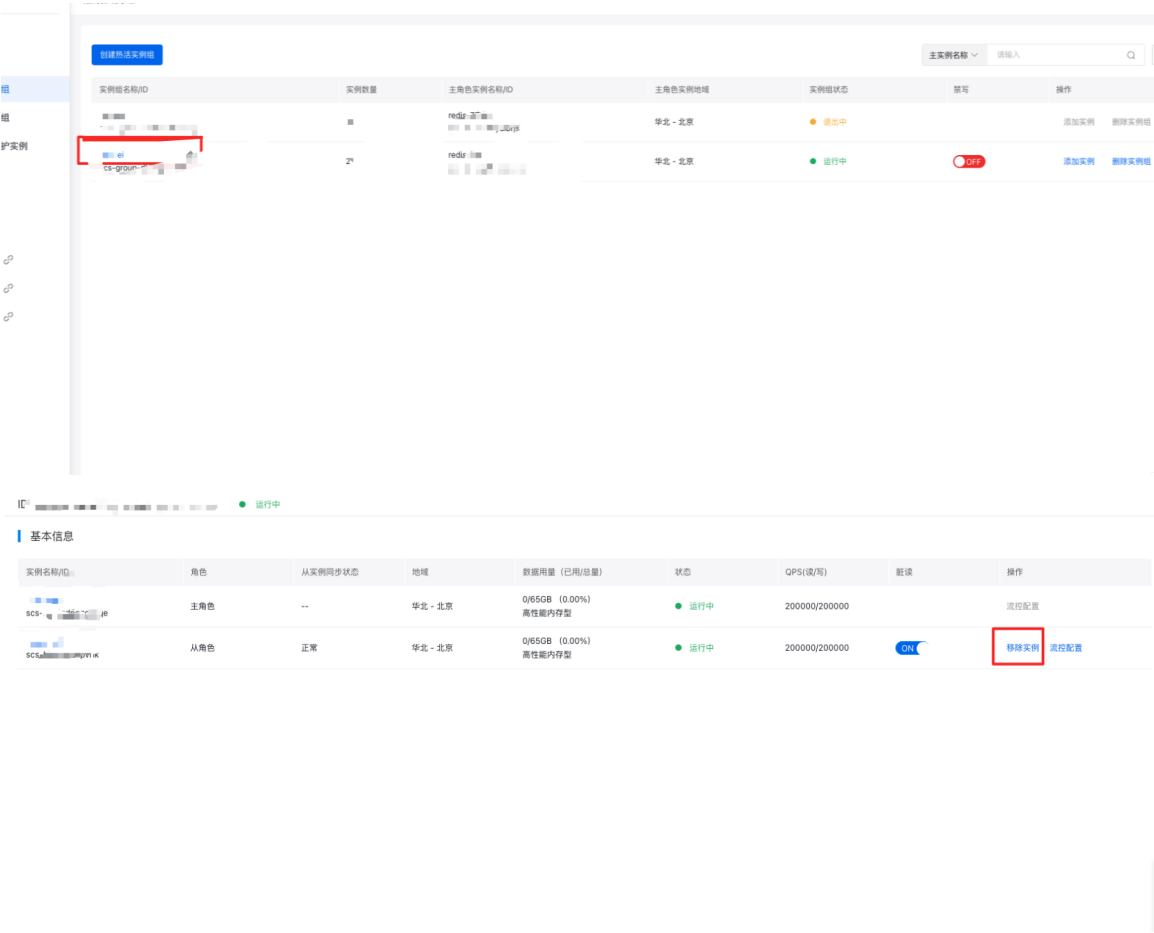
4.添加实例。热活实例组中实例需满足特定条件（以下均需为是）：

- 实例状态是否为运行中
- 实例引擎类型是否为社区标准版或者企业集群版
- 实例节点规格是否不大于16GB
- 实例是否未加入实例组
- 实例是否未关联VPC侧安全组
- 实例是否未开启TDE加密功能
- 是否具备相同的hashtag参数值
- 是否设置了密码



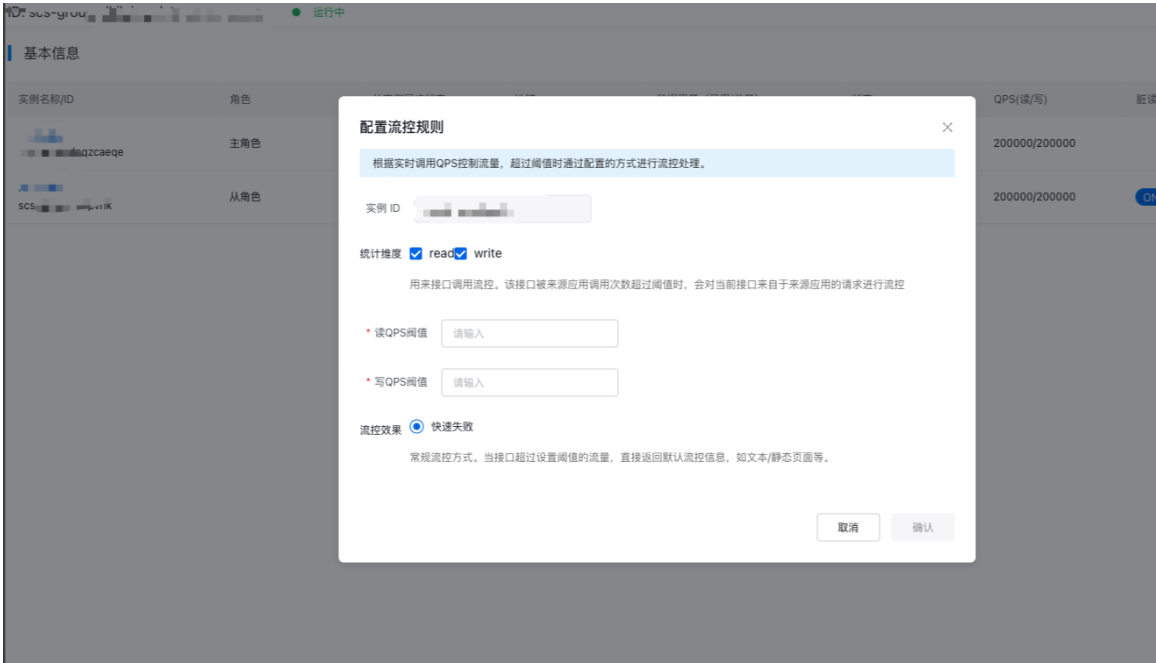


5.点击基本信息页面查看，可进行部分实例移除。



6.点击角色流量配置，可调用QPS控制流量。

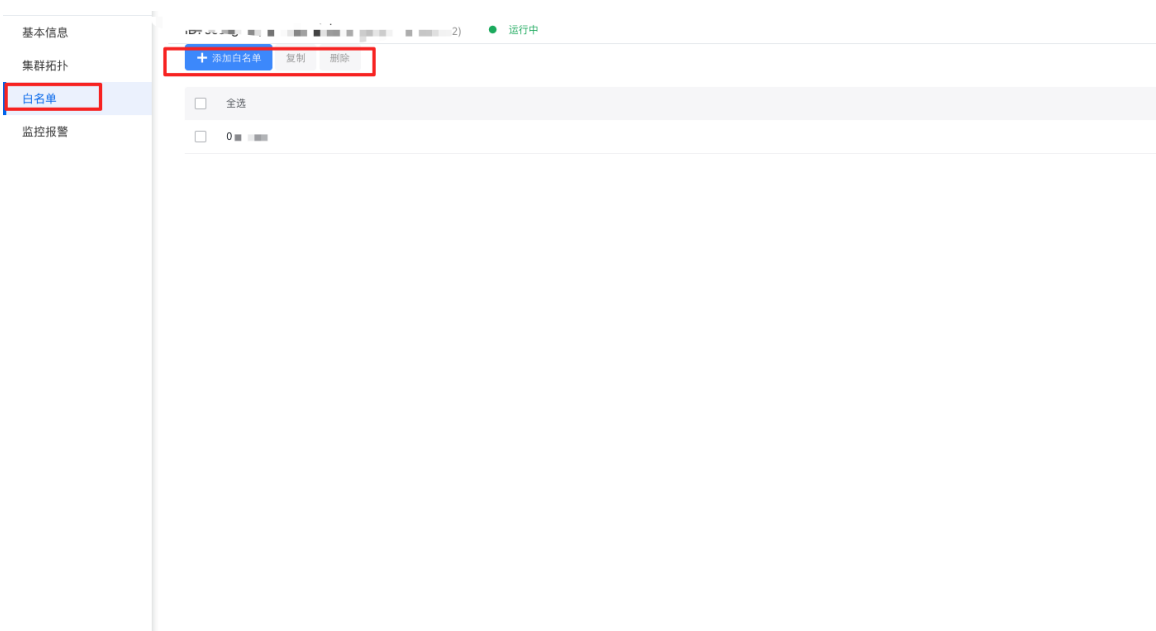




7.点击集群拓扑，查看信息。

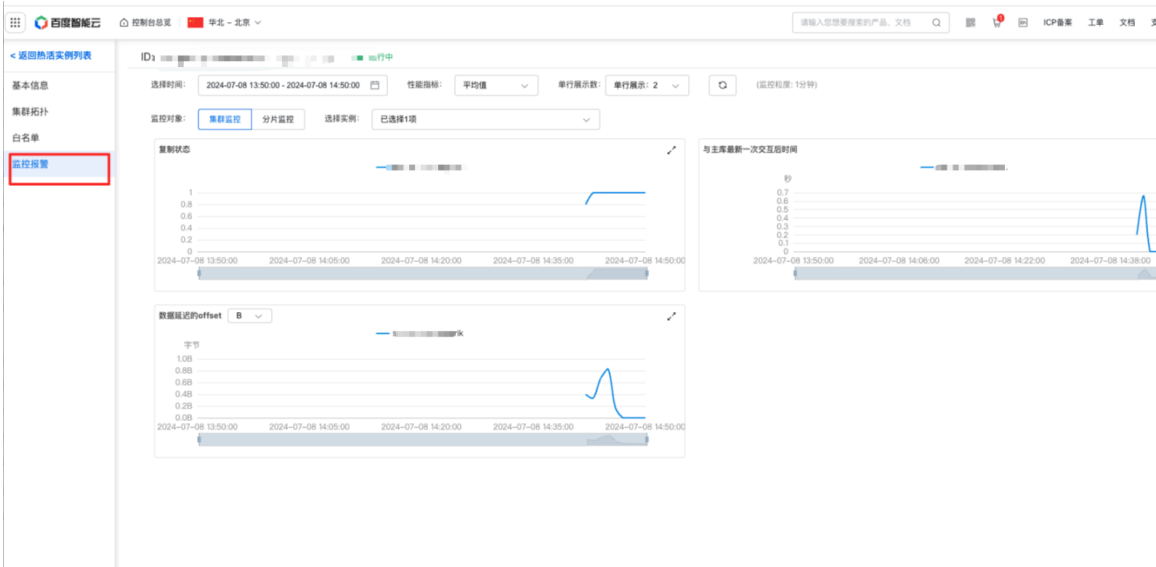
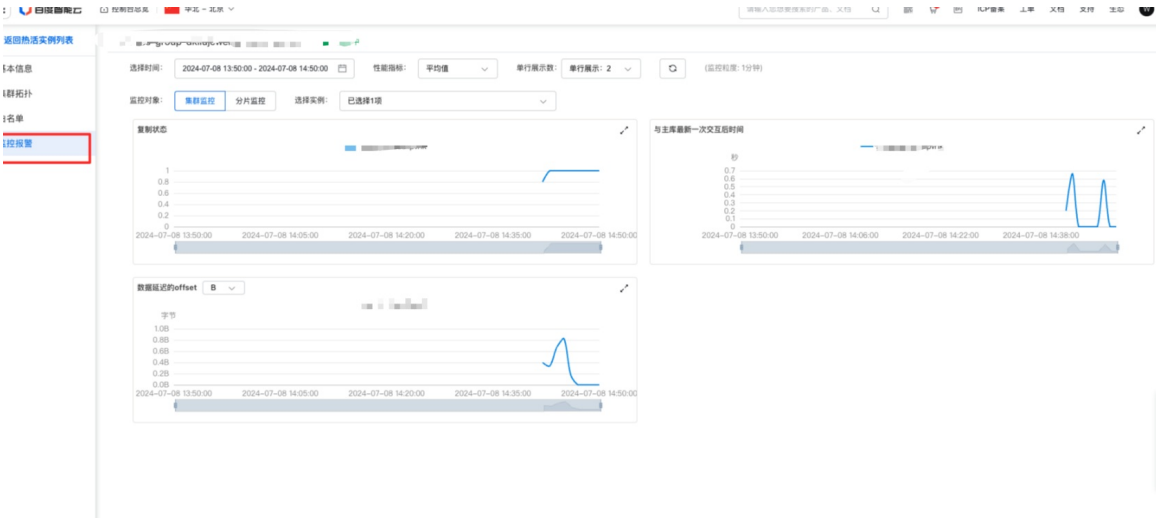


8.点击白名单配置，可添加白名单信息。



9.点击监控报警，查看基本信息，可筛选时间，具体指标，监控对象、实例等进行小范围监控。





10.删除热活实例组。

库 Redis	热活实例组						
	创建热活实例组   主实例名称 请输入						
	实例组名称/ID	实例数量	主角色实例名称/ID	主角色实例地域	实例组状态	禁写	操作
	实例组名称/ID	2	主角色实例名称/ID	华北 - 北京	退出中		添加实例 删除实例组
维护实例	实例组名称/ID	1	主角色实例名称/ID	华北 - 北京	运行中	禁写	添加实例 删除实例组

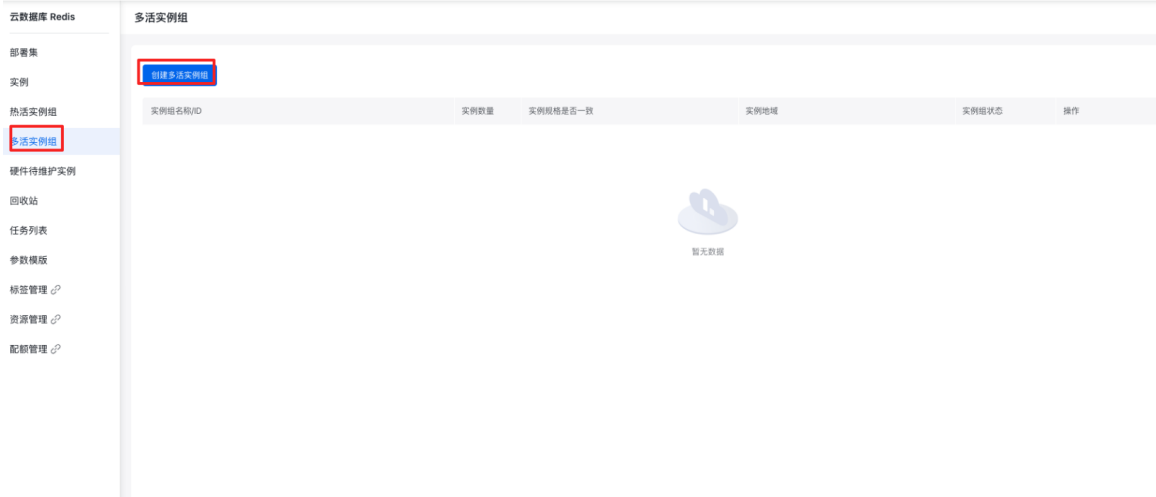
创建多活实例组

概述

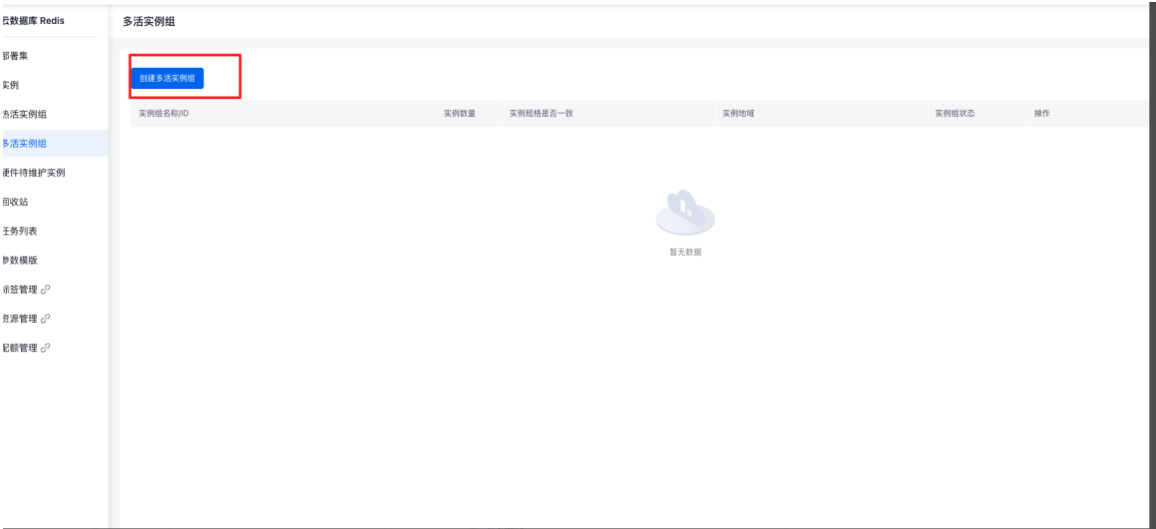
百度云Redis提供创建多活实例架构管理的功能，用户可以管理多活实例组。

操作步骤

1. 登录 [云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库Redis-多活实例组”页面。



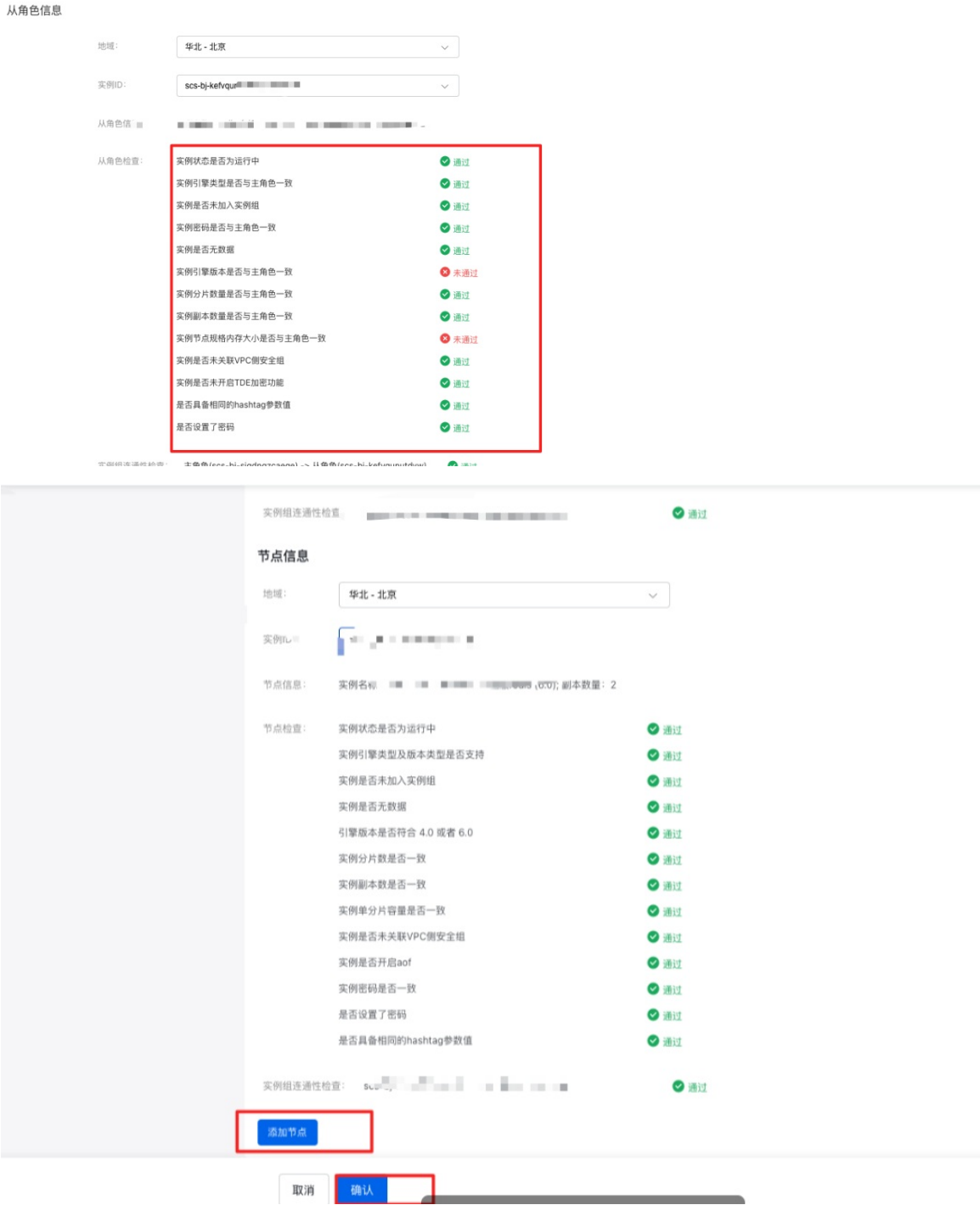
3. 创建多活实例组。



4.添加实例并建立。要求节点数>=2，故至少添加两个实例，且实例需要满足一定要求。

实例需要满足的要求（均需为是）：

- 实例状态是否在为运行中
- 实例引擎类型以及版本类型是否支持
- 实例是否加入实例组
- 实例是否无数据
- 引擎版本是否符合4.0或者6.0
- 实例分片数是否一致
- 实例副本数是否一致
- 实例单片容量是否一致
- 实例是否关联VPC侧安全组
- 实例是否开启aof
- 实例密码是否一致
- 是否设置了密码
- 是否具备相同的hashtag参数值



5.继续添加子实例，子实例不能与原有实例重复，且密码要相同。

新增的实例需要满足的要求（均需为是）：

- 实例状态是否在为运行中
- 实例引擎类型以及版本类型是否支持
- 实例是否加入实例组
- 实例是否无数据
- 引擎版本是否符合4.0或者6.0
- 实例分片数是否一致
- 实例副本数是否一致
- 实例单片容量是否一致

- 实例是否关联VPC侧安全组
- 实例是否开启aof
- 实例密码是否一致
- 是否设置了密码
- 是否具备相同的hashtag参数值

创建多活实例组

实例组名称/ID	实例数量	实例规格是否一致	实例地域	实例组状态	操作
	2	节点规格：一致 分片规格：一致	华北 - 北京：可用区D	运行中	<a href="#">删除实例组</a> <a href="#">添加子实例</a>

基本信息

实例组名称：

实例组ID：

节点信息

地域：

华北 - 北京

实例ID：

节点信息：

实例名称： 类型：企业版redis (6.0); 副本数量：2

节点检查：

实例状态是否为运行中

通过

实例引擎类型及版本类型是否支持

通过

实例是否未加入实例组

未通过

实例是否无数据

通过

引擎版本是否符合 4.0 或者 6.0

通过

实例分片数是否一致

通过

实例副本数是否一致

通过

实例单分片容量是否一致

通过

实例是否未关联VPC侧安全组

通过

实例是否开启aof

通过

实例密码是否一致

未通过

是否设置了密码

通过

是否具备相同的hashtag参数值

通过

6.点击基本信息页，查看实例信息，也可进行删除子实例。

返回多活实例列表

产品简介 / 云数据库Redis-多活实例组列表 / 基本信息

ID: 运行中

基本信息

实例名称

实例规格

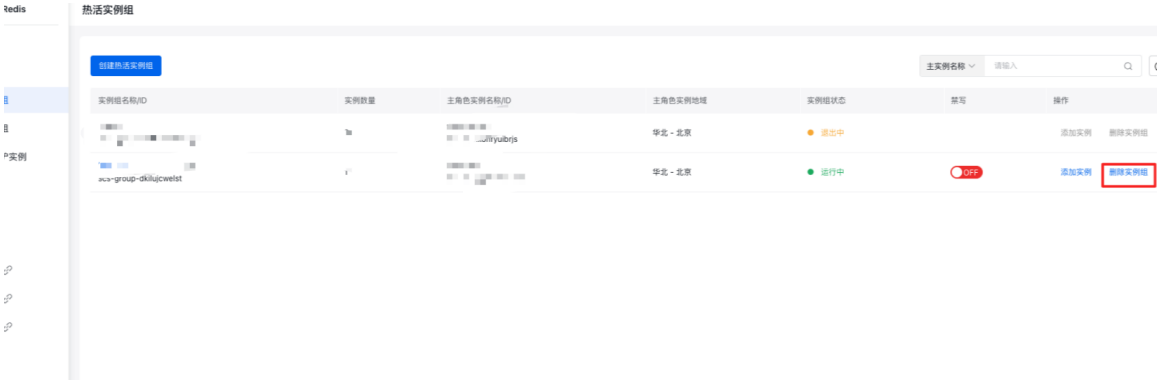
地域

数据用量 (已用/总量)

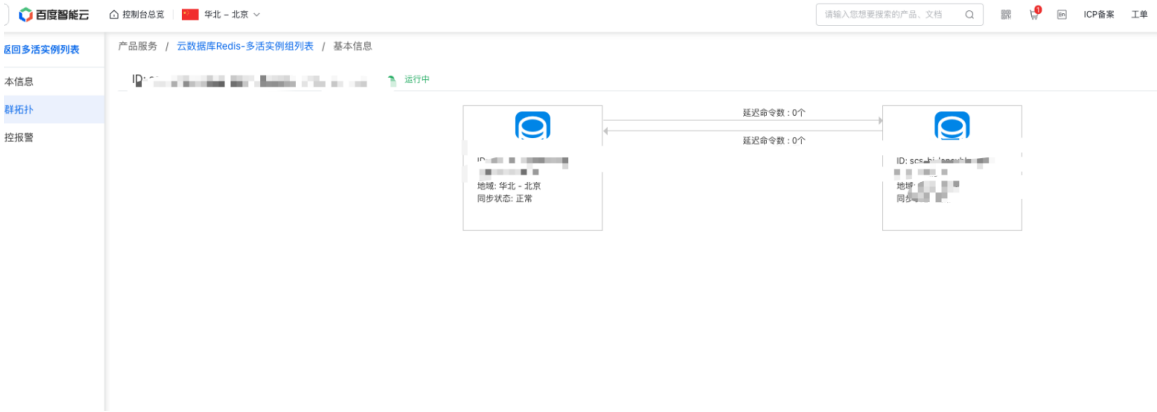
状态

操作

	正常	华北 - 北京	0.00MB/4GB (0.00%)	运行中	<a href="#">删除子实例</a>
	正常	华北 - 北京	0.00MB/4GB (0.00%)	运行中	移除子实例



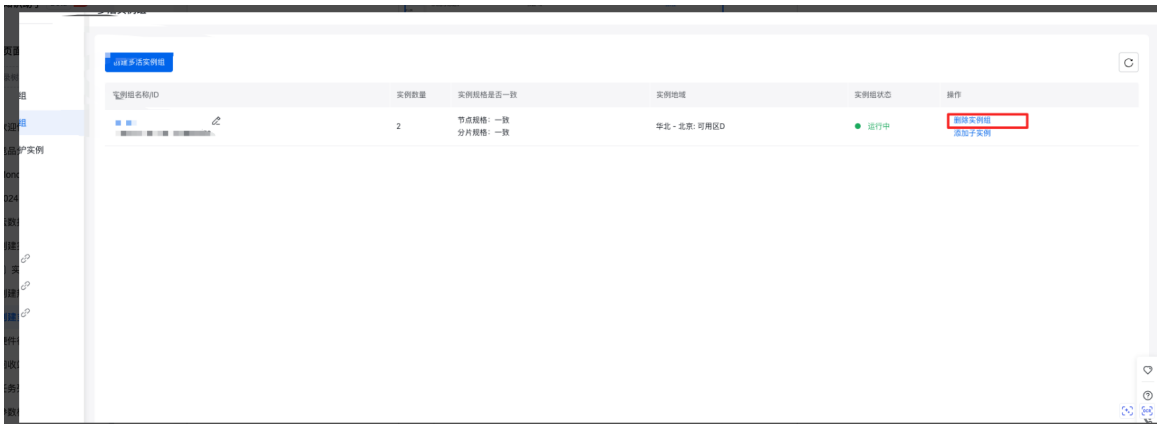
7.点击集群拓扑查看信息。



8.监控预警，可选择时间、监控指标等进行条件筛选后监控。



9.删除实例组。



实例状态与影响

在对Redis实例执行变更配置、重启等操作时，实例的状态将发生变化并产生某些影响（例如产生秒级的连接闪断），本文为您

列出相关信息，为您管理实例提供参考。

🔗 实例状态说明

实例状态可通过下述两种方式查询到：

- 实例状态可在[控制台](#)的实例列表页查看。
- API：调用 [API接口](#) 获取。

控制台	API	触发该状态的操作	说明或影响
创建中	<a href="#">创建</a>	<a href="#">创建实例</a>	完成创建实例的操作后，在进入运行中之前的状态，此时不可提供数据库服务。
运行中	无	无	实例处于该状态时，可正常提供数据库服务。
转换中	无	<a href="#">预付费转后付费</a> ； <a href="#">后付费转预付费</a>	<a href="#">计费方式变更</a>
变配中	<a href="#">变配</a>	<a href="#">变更配置</a>	变配过程可能引发延迟波动，请在业务低峰期执行该操作并确保应用具备重连机制。
重启中	<a href="#">重启</a>	<a href="#">重启实例</a>	实例会出现秒级的连接闪断，请在业务低峰期执行该操作并确保应用具备重连机制。
小版本升级	无	<a href="#">小版本升级</a>	升级过程中实例将重启，可能会有分片数*5秒内的闪断，请选择业务低峰期进行，并确保应用具有自动重连机制。
TDE修改中	<a href="#">TDE加密</a>	<a href="#">开启TDE加密</a>	无影响
可用区迁移中	无	<a href="#">迁移可用区</a>	实例会出现秒级的连接闪断，请在业务低峰期执行该操作并确保应用具备重连机制。
清空实例中	<a href="#">清除</a>	<a href="#">清空数据</a>	清空数据即刻生效，且会对线上业务造成影响，请谨慎评估并在业务低峰期执行。
切换中	无	<a href="#">主从切换</a>	主节点角色将移交到其他可用区的从节点。如果所有节点都位于同一个可用区，则不支持该操作。主从切换过程中可能会有1-2次闪断，请选择业务低峰期进行。
被禁用	无	无	该实例已到期，无法提供服务。如需继续使用，您需要执行手动续费操作，具体操作，请参见 <a href="#">手动续费</a> ， <a href="#">自动续费</a> 。
已释放	无	<a href="#">手动释放</a> 、 <a href="#">欠费或实例到期自动释放</a>	该实例已被释放，无法提供服务，且无法在实例列表中展示。如需继续使用，您可以在回收站中找到该实例并执行重建恢复操作。更多信息，请参见 <a href="#">实例回收站</a> 。

Redis 实例生命周期流程

本文介绍Redis实例的生命周期流程，即从实例创建（购买）到释放涉及的操作。

🔗 生命周期流程

图 1. 生命周期流程



**重要：**对实例执行相关操作，会引发实例状态的变化并产生某些影响（例如产生秒级的连接闪断）。更多详情，请参见[实例状

态与影响]。

操作	说明
创建实例	云数据库Redis分为社区版和企业版,您可以创建适应业务需求的Redis实例。
变更实例配置	通过变更实例的规格、架构和类型，适应不同场景对实例性能和兼容性的需求。
调整实例的分片数量	分片数量越多，实例的整体性能越强，您可以根据业务对性能的需求，调整实例的分片数量。
重启实例	当实例出现连接数满或性能问题时，您可以重启Redis实例以释放所有连接。
升级小版本	云数据库Redis会不断地对内核进行深度优化，修复安全漏洞，提升服务稳定性，推荐定期检查并升级小版本。
释放按量付费实例	对于闲置的Redis实例资源，您可以释放按量付费实例或退订包年包月实例。
管理回收站中的实例	回收站用于保存到期、欠费、已释放的Redis实例，您可以在回收站对这些实例进行续费开机、彻底销毁等操作。

迁移可用区

概述

当可用区资源不足无法升级实例规格、需要提高容灾能力或因其它原因需要将现有实例切换到其它可用区时，您可以通过控制台或API操作更换云数据库 Redis 版实例所属的可用区。更换后，实例的数据、账号、连接地址等信息均不会改变。

使用限制

- 迁移可用区功能不支持热活实例组、多活实例组、不支持只读节点。
- Redis社区版、集群版、容量版均支持可用区迁移功能。
- 主从节点同一个可用区下的子网需要保持一致。
- Redis社区版只读节点不支持迁移，但Redis社区版实例其它主从节点迁移时子网的选择会受到限制，即如果迁移到只读节点所在可用区时，子网信息默认需要与只读节点一致。
- 变更选取的可用区、子网不可与之前完全一致，即选取节点完全没变化时不支持变更。
- 开通公网的实例不支持可用区迁移。
- 实例有待执行的维护窗口任务，不支持可用区迁移。

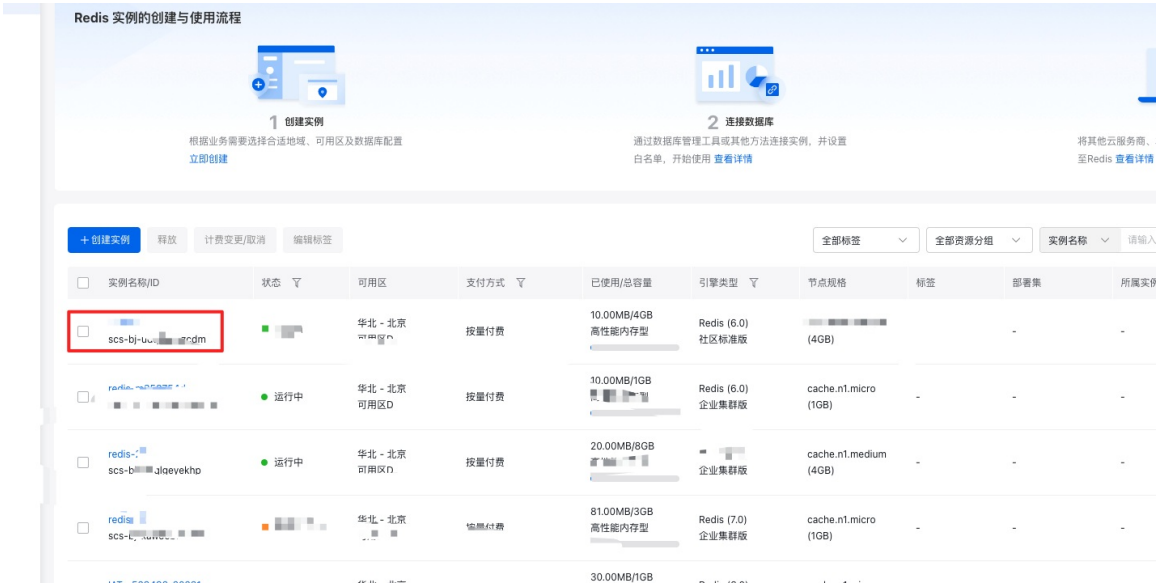
注意：不满足以上条件时，迁移按键将置灰。

注意事项

- 迁移可用区会造成连接闪断，请确保业务具有重连机制并在业务低峰期执行。
- 迁移过程中及完成后，可正常使用迁移前的内网访问IP。
- 迁移完成后建议在网络连接信息中进行内网IP变更，切换至节点最新IP，来避免跨区访问高延时问题。

操作步骤

1. 登录云数据库 Redis 管理控制台。
2. 选择要迁移的实例，点击实例名称，进入实例基本信息页面。



3. 在基本信息区域的可用区右侧，单击迁移。



4. 阅读注意事项。





5. 在弹出的面板中，完成下述配置。

配置	说明
主节点子网	选择目标可用区，以及子网信息。
从节点子网	选择目标可用区，以及子网信息。
执行时间	● <b>立即执行</b>：单击确定后，系统立即执行迁移可用区任务，迁移过程中不支持进行变更配置相关操作，实例状态恢复为运行中时，迁移可用区成功。    ● <b>可维护时间内执行</b>：单击确定后，该过程实例状态仍显示为运行中，可正常提供服务，但不支持进行变更配置相关操作，到达可维护时间段才正式执行迁移操作。    ● <b>设置维护时间具体操作</b>：<a href="#">设置维护窗口</a>

5. 明确注意事项后，单击**确定**。
6. 迁移确认后，您可在**任务列表**查看迁移具体过程，迁移进度。

IP变更

概述

迁移可用区完成后您可以在网络连接信息中进行内网IP变更，切换至节点最新IP，来避免跨区访问高延时间问题。

使用限制

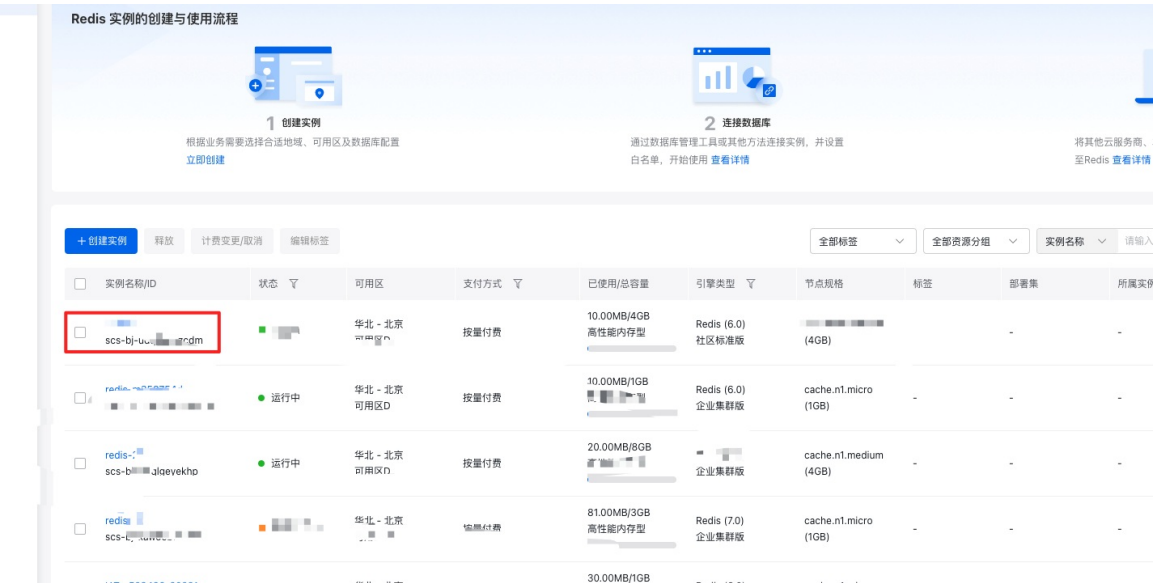
- VPC默认不变，可用区、子网默认选择与主节点一致。
- 修改网络地址生效后，会断开旧地址上的所有网络连接（旧地址不可用）。请确保业务使用连接地址（**域名**）连接实例。并确保迁移后DNS缓存及时刷新，且应用程序具备断线重连机制。
- Redis**社区版**、**集群版**、**容量版**均支持**IP变更**功能。

- 变更IP不支持热活实例组、多活实例组，只读节点。
- 变更IP不支持开通外网ip。
- BLB专属集群的实例不支持变更内网ip。



🔗 操作步骤

1. 登录云数据库 Redis 管理控制台。
2. 选择要迁移的实例，点击实例名称，进入实例基本信息页面。



3. 在网络信息区域的内网IP右侧，单击变更。

4. 阅读注意事项，明确后点击确认。

配置	说明
VPC、可用区、子网	默认选择与主节点一致。
执行时间	● <b>立即执行</b>：单击确定后，系统立即执行<b>IP变更</b>任务，变更过程中不支持进行<b>变更配置</b>相关操作，实例状态恢复为<b>运行中</b>时，变更成功。     ● <b>可维护时间内执行</b>：单击确定后，该过程实例状态仍显示为<b>运行中</b>，可正常提供服务，但不支持进行变更配置相关操作，到达可维护时间段才正式执行<b>IP变更</b>操作。     ● <b>设置维护时间具体操作</b>：<a href="#">设置维护窗口</a>

## 实例生命周期

- ## 9. 释放实例

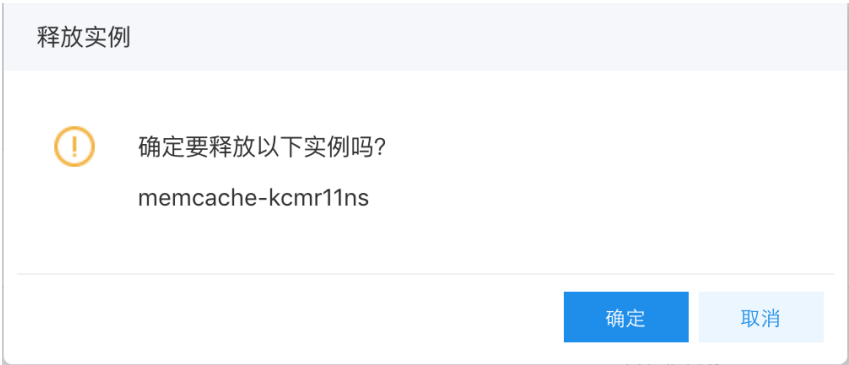
本文为您介绍通过管理控制台进行释放实例的通用方法。

释放实例

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面的“云数据库 Redis-实例列表”页面，选择需要释放实例名称/ID。



3. 点击释放按钮，弹出释放确认界面。



说明：预付费的SCS实例无法提前释放。

4. 点击确定按钮，弹出“安全验证页面”，进行手机验证码二次验证。



5. 输入短信验证码，点击确认按钮，完成释放操作。

相关 API

- API 接口：[释放实例](#)

续费实例

自动续费

概述

**自动续费：**您可在控制台上设置预付费实例进行自动续费。若预付费实例到期未续费，会导致业务中断甚至数据丢失。

- 使用场景：适用于长期使用云数据库 Redis 实例的场景。

- 优点：操作简便，避免因遗忘续费导致数据删除业务中断的情况，安全性更高。

注意事项

- 开通自动续费后，系统会在到期日前7天自动为其续费。
- 请保证百度智能云账号的余额充足，避免续费失败。续费成功或失败都给予短信和邮件提醒。
- 自动续费将按照您最近一次设置的续费周期执行。
- 在一个周期内，资源的到期时间由于其他操作而延长，自动续费将在下个到期日再执行。
- 自动续费执行时，会针对某一个实例资源生成一个续费订单。

创建实例时开通

您可以在购买预付费实例时开通**自动续费**，自定义续费周期，选择按月（1~9个月）或按年（1~3年）进行自动扣费。

购买信息

购买时长:

1个月

2

3

4

5

6

7

8

9

Sale

1年

Sale

2年

Sale

3年

实例个数:

-

1

+

最多可购买10个SCS实例，还可购买9个

自动续费:

ON

什么是自动续费?

选择续费周期:

按年

1年

系统将于到期前7天进行扣费，扣费时长为1年

创建实例后开通

1. 登录[云数据库 Redis 管理控制台](#)，进入“云数据库 Redis-实例列表”页面。
2. 在控制台右上方，选择**财务**菜单下的子菜单**续费管理**。
3. 在续费管理页面选择**到期时间**，点击[云数据库Redis](#) 页签，选择实例所在地域，找到目标实例。

续费管理

操作指南

到期时间:

全部时间

已到期

7天内到期

15天内到期

30天内到期

3个月内到期

产品筛选:

云服务器 BCC(19)

云数据库DocDB for MongoDB(1)

VPN网关 VPN(2)

云数据库SCS for Redis(1)

弹性公网IP EIP(10)

云磁盘 CDS(4)

云数据库RDS(1)

应用防火墙 WAF(1)

地域选择:

华北 - 北京 (1)

续费

设置自动续费

关闭自动续费

实例名称 请输入实例名称进行搜索

刷新

<input checked="" type="checkbox"/>	实例名称	实例ID	到期时间	自动续费/续费周期	下次自动续费日期
<input checked="" type="checkbox"/>	redis-k104quhg		2019-10-26 11:26:19, 已过期261天	未开通	-

每页展示

15

<

1

>

4. 勾选目标实例，点击实例名称上方的**设置自动续费**按钮。
5. 在弹出的设置自动续费窗口中选择**续费周期**（按年或按月），点击**确定**按钮。



6. 自动续费开通成功，系统会发送短信和邮件通知。

手动续费

概述

**手动续费：**默认续费方式，适用于初期使用且有运维人员按时进行续费操作的场景。

- 缺点：续费周期即将结束前需要手动续费，操作较为繁琐，可能遗忘续费而导致实例停止服务或数据丢失。
- 解决方案：设置自动续费续费周期，设置一次永久生效。

说明：

- 预付费实例：有到期时间。若到期未续费，会导致业务中断甚至数据丢失。您可以根据业务需要，对“预付费”实例进行续费。
- 后付费实例：无到期时间，无须进行续费相关操作。

Redis 控制台续费实例

1. 登录[云数据库 Redis 管理控制台](#)，进入“云数据库 Redis-实例列表”页面。
2. 选择需要续费的实例，点击此实例操作栏中的续费选项，进入“财务中心/续费”页面。
3. 在批量选择续费时长选项中，先选择按月或按年进行续费，再确定具体续费时间长度。



**注意：** 选择按年续费，最多续费3年；按月续费，最多续费9个月。

4. 确定好续费时间和续费金额后，点击右下角下一步按钮，进行订单付费，订单支付后，正式完成续费操作。
5. 完成续费后，返回“云数据库 Redis-实例列表”页面，可在支付方式下看到SCS实例的到期时间已更新。

续费管理续费实例

1. 登录[云数据库 Redis 管理控制台](#)。

2. 在控制台右上方，选择 **财务** 菜单的子菜单项**续费管理**。



3. 进入续费管理页面，通过“到期时间、产品筛选、地域选择”筛选栏，找到目标续费实例。

4. 勾选一个或多个实例，点击**续费**按钮，进入“财务中心／续费”页面，即可进行单个或批量实例续费。



5. 在**批量选择续费时长**选项中，先选择**按月**或**按年**进行续费，再确定具体续费时间长度。



注意：选择按年续费，最多续费3年；按月续费，最多续费9个月。

6. 确定好续费时间和续费金额后，点击右下角**下一步**按钮，进行订单付费，订单支付后，正式完成续费操作。

7. 完成续费后，返回“云数据库 Redis-实例列表”页面，可在**支付方式**下看到 Redis 实例的到期时间已更新。

计费变更

☞ 预付费转后付费

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍预付费转后付费的操作。

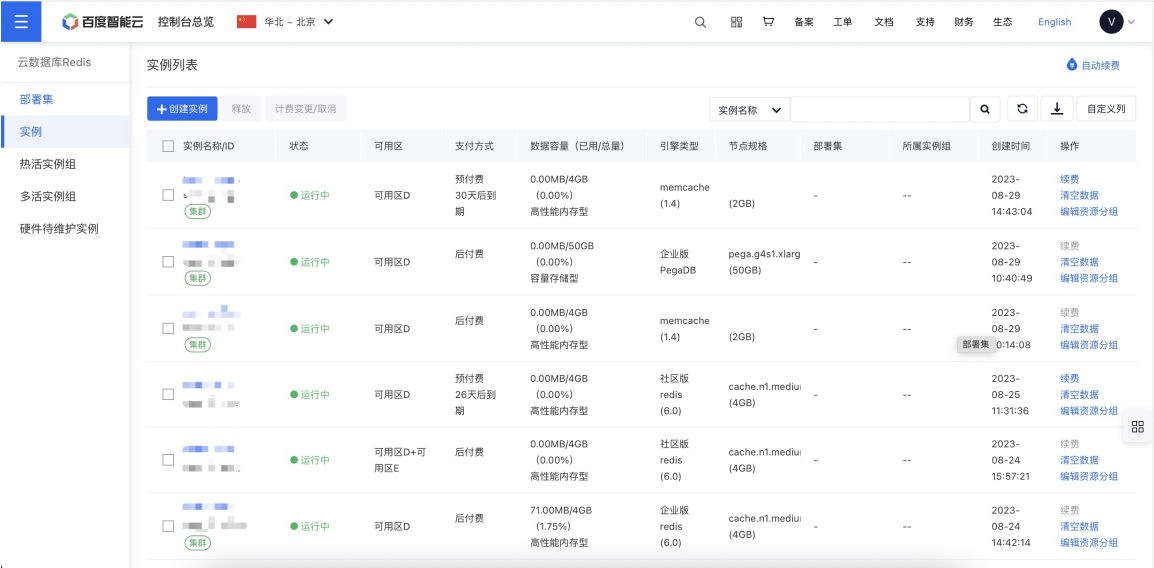
注意事项

- 已选择了自动续费的预付费实例不能进行计费变更。
- 运行状态为已到期、重启中、迁移中的实例不支持预付费转后付费。
- 用户将预付费计费方式变更为后付费时需要确保余额充足（余额>100），避免设置生效时余额检查失败。

\*\*预付费转后付费\*\*

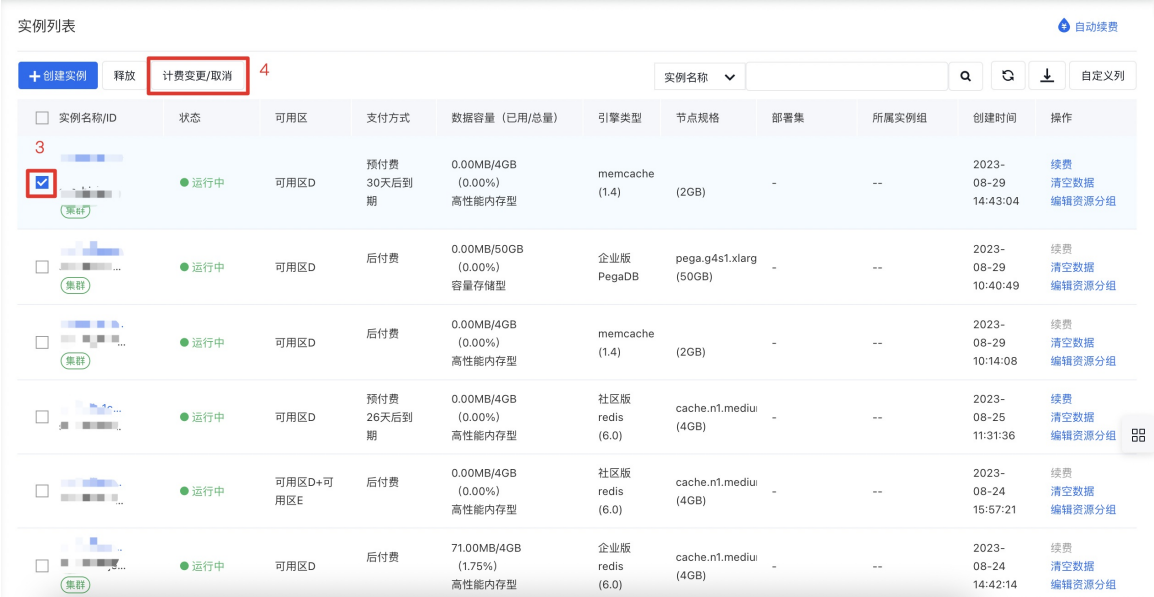
1. 登录**云数据库 Redis 控制台**。

2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。



3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。

4. 点击**计费变更/取消**按钮，进入计费变更页面，对实例进行计费模式的变更。



5. 确认订单，点击**下一步**按钮，按照提示支付订单。



6. 预付费转变为后付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

7. 在实例列表的右上角，单击**刷新**列表，可查看到实例状态显示为**运行中**。“支付方式”显示为**预付费(待变更)**。实例的后付费计费方式需要等**预付费到期**后才会生效，未到期前可以**取消计费变更**。到期后支付方式会显示为**后付费**。



产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>   集群	运行中	可用区D	预付费 (待变更)  1  30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	后付费	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	续费 清空数据 编辑资源分组

取消计费变更

用户在进行预付费转后付费后，预付费未到期前可以取消计费变更，继续使用预付费的计费方式。

1. 登录云数据库 Redis 管理控制台。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>   集群	运行中	可用区D	预付费 (待变更)  1  30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	后付费	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	续费 清空数据 编辑资源分组
<input type="checkbox"/>   集群	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	续费 清空数据 编辑资源分组

3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量取消计费变更。
4. 点击计费变更/取消按钮，

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

4

实例名称

Q

🔄

📄

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
3   <input checked="" type="checkbox"/>   集群	● 运行中	可用区D	预付费 (待变更)  1   30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	后付费	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

5. 在弹出的取消计费变更对话框中点击确定按钮，可以看到计费变更已取消的消息。

取消计费变更

!

确认取消计费变更？

您已开通计费变更-预付费转后付费功能。

实例将会在到期后自动转换为后付费的计费方式。

确定

取消

6. 取消计费变更成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

7. 在实例列表的右上角，单击  刷新列表，可查看到实例状态显示为运行中。“支付方式”显示为预付费。

百度智能云 控制台总览 华北 - 北京

云数据库Redis

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

Q

🔄

📄

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	后付费	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 0:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	后付费	71.00MB/4GB (1.75%) 高性能内存型	企业版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 14:42:14	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍后付费转预付费的操作。

注意事项

- 运行状态为已到期、重启中、迁移中的实例不支持后付费转预付费。

\*\*后付费转预付费\*\*

- 登录[云数据库 Redis 管理控制台](#)。
- 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

Q

↺

⬇

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量（已用/总量）	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>  集群	运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	后付费	71.00MB/4GB (1.75%) 高性能内存型	企业版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 14:42:14	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

- 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。
- 点击计费变更/取消按钮，进入计费变更页面，对实例进行计费模式的变更。

产品服务 / 云数据库Redis-实例列表

自动续费

+ 创建实例

释放

计费变更/取消

4

实例名称

Q

↺

⬇

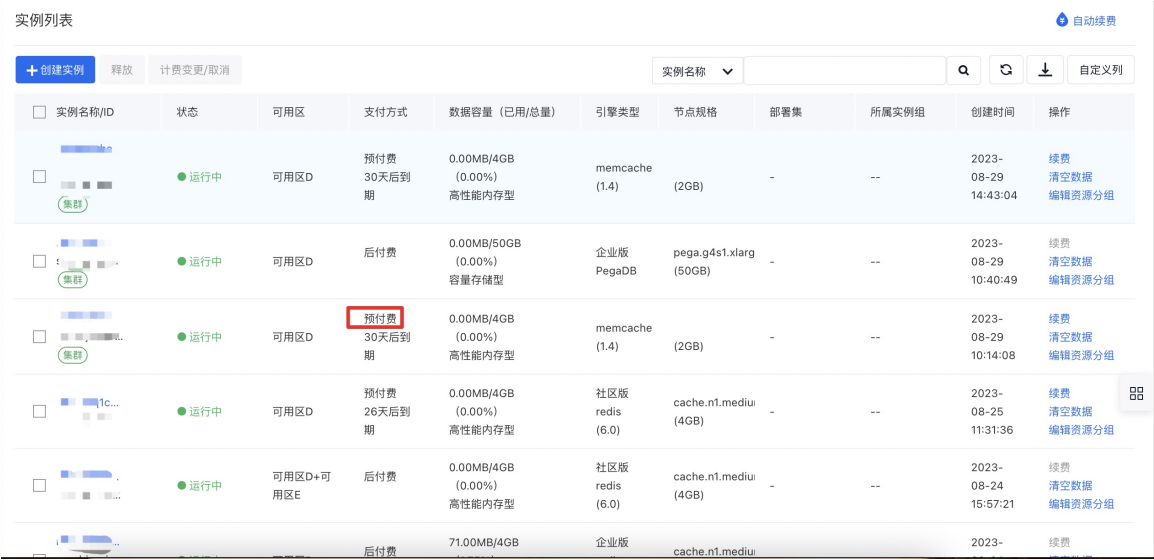
自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量（已用/总量）	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>  集群	运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.xlarg (50GB)	-	--	2023-08-29 10:40:49	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input checked="" type="checkbox"/>  集群	运行中	可用区D	后付费	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D	预付费 26天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>  集群	运行中	可用区D+可用区E	后付费	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-24 15:57:21	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

- 选择预付费购买时长，可以按月购买或按年购买，最小预付费时长为一个月。
- 确认订单，点击下一步按钮，按照提示支付订单。



7. 后付费转变为预付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。
8. 在实例列表的右上角，单击  刷新列表，可查看到后付费转预付费完成后，实例状态显示为运行中。“支付方式”显示为预付费。



## 变更实例

### 变更配置

#### 概述

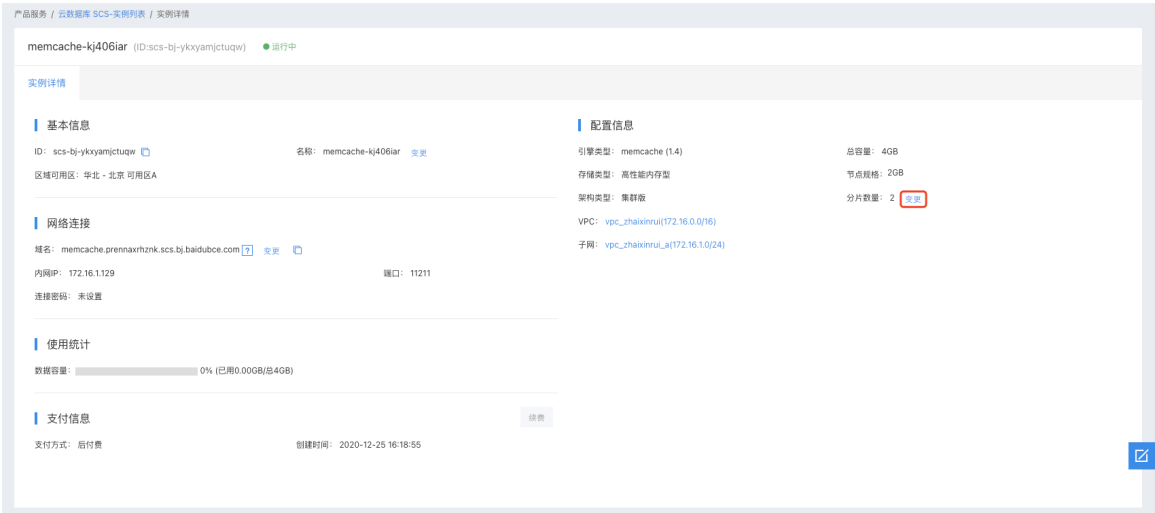
本文为您介绍通过云数据库 Redis 管理控制台变更配置的基本方法。

#### 注意事项

- 如果变更配置时所需IP数量超过子网剩余IP数量，请提交[工单](#)。

#### 释放实例

- 登录[云数据库 Redis 管理控制台](#)。
- 在SCS控制台页面的“云数据库 Redis-实例列表”页面，选择需要变更配置的实例。
- 点击实例名称进入实例详情页面，点击右侧配置信息中分片数量一栏的变更，进入“变更配置”页面。



4. 在“变更配置”中，选择您需要增加的分片容量，例如您需要由1GB增加到16GB，只需为15GB的缓存容量付费。



5. 然后在右侧订单栏点击下一步按钮，进行订单付费，完成订单付费后，正式完成变更配置的操作。

常见问题

- 资源配置发生更改后，资费是如何变更的？  
答：资源配置发生更改后，需要根据用户需要重新购买服务。

相关API

- API 接口：[变更实例](#)

🔗 修改实例名称

概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例名称的基本方法。

修改实例名称

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在SCS产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处运行中。
3. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



4. 点击基本信息模块**变更**按钮，输入变更后名称。



5. 点击**确认**按钮，用户可以完成修改实例名称的操作。

相关 API

- API 接口：[修改实例名称](#)

🔗 修改实例域名

概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例域名的基本方法。

修改实例域名

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在SCS产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处**运行中**。
3. 点击“实例名称/ID”下面的目标**实例名称**，进入“实例详情”页面。



4. 点击**变更**，输入变更后域名。



5. 点击**确认**按钮，用户可以完成修改实例域名的操作。

相关 API

- API 接口：[变更实例](#)

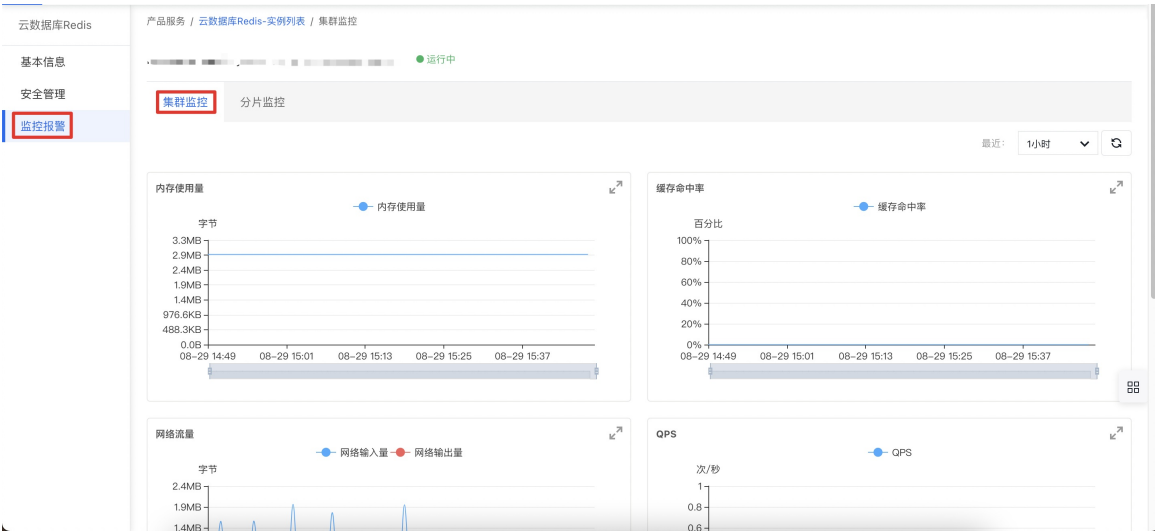
监控报警

查看监控数据

查看集群监控数据

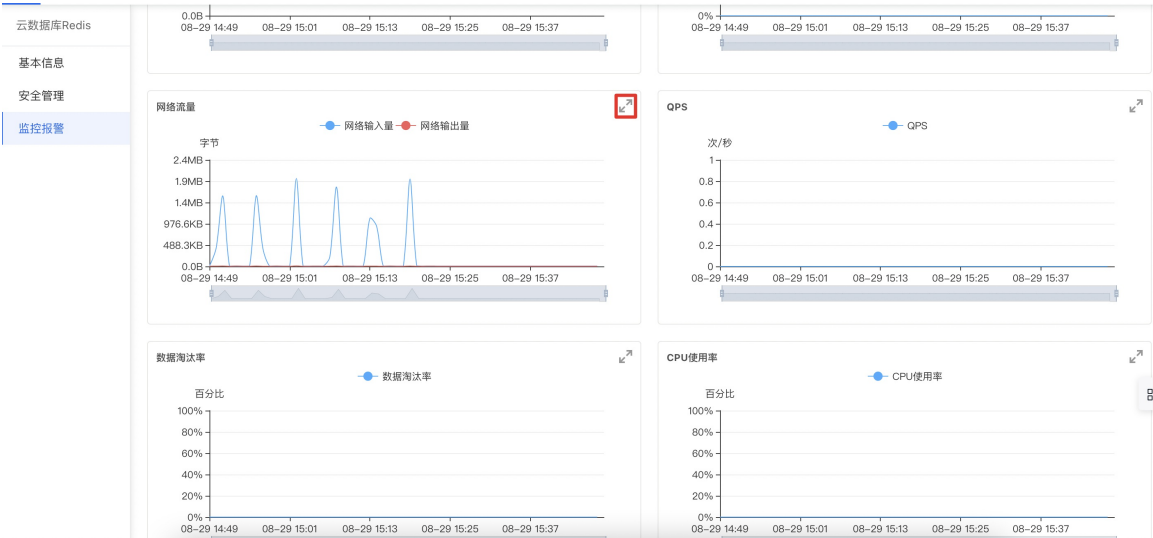
1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。
3. 找到目标 Memcache 实例，点击**实例名称**，进入实例详情页面。
4. 在左侧导航栏中单击**监控管理**，进入监控管理页面。
5. 在监控管理分为**集群监控**和**分片监控**页签。
6. 在实例的**集群监控**页面，可以看到实例监控信息趋势图。您可查看最近1小时-40天内的监控数据。

监控项包括：内存使用量、缓存命中率、网络流量、QPS、数据淘汰率、CPU使用率。

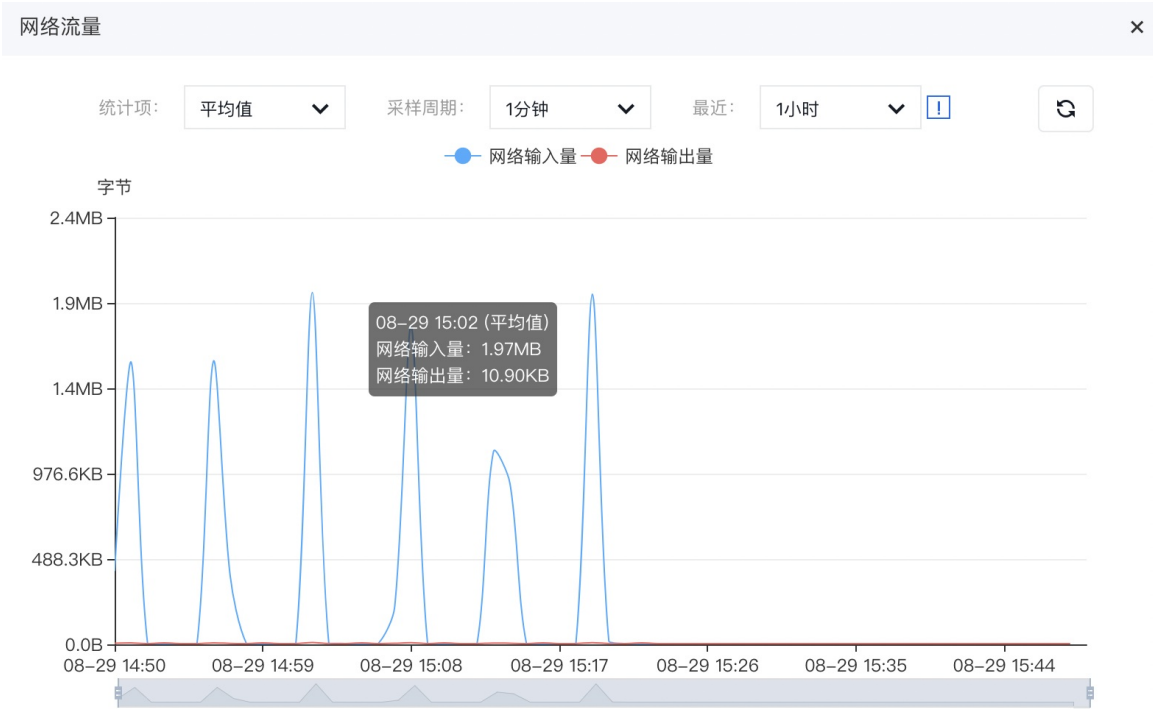


以“网络流量”监控项为例：

- 点击监控曲线图的**缩放**按钮，可查看详细信息。



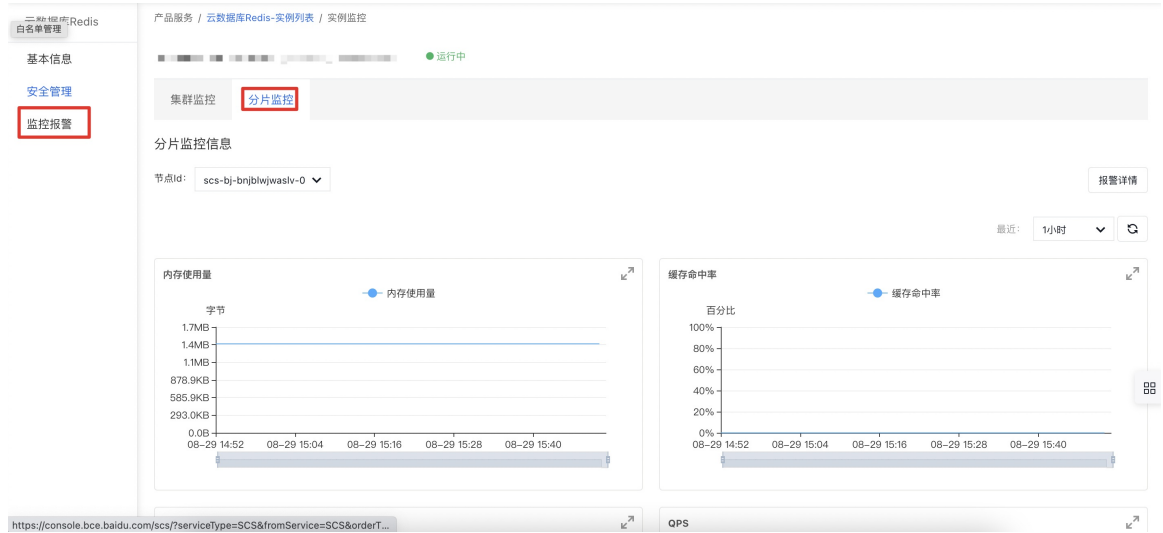
- 选择统计项、采样周期、统计时间，即可查看最新数据。



查看指定节点的监控数据

1. 选择监控管理页面的分片监控页签，进入分片监控页面。
2. 在“分片监控”页面，选择想要查看的节点 ID 和查询时间，您可查看最近1小时-40天内的监控数据。





3. 查看监控数据：

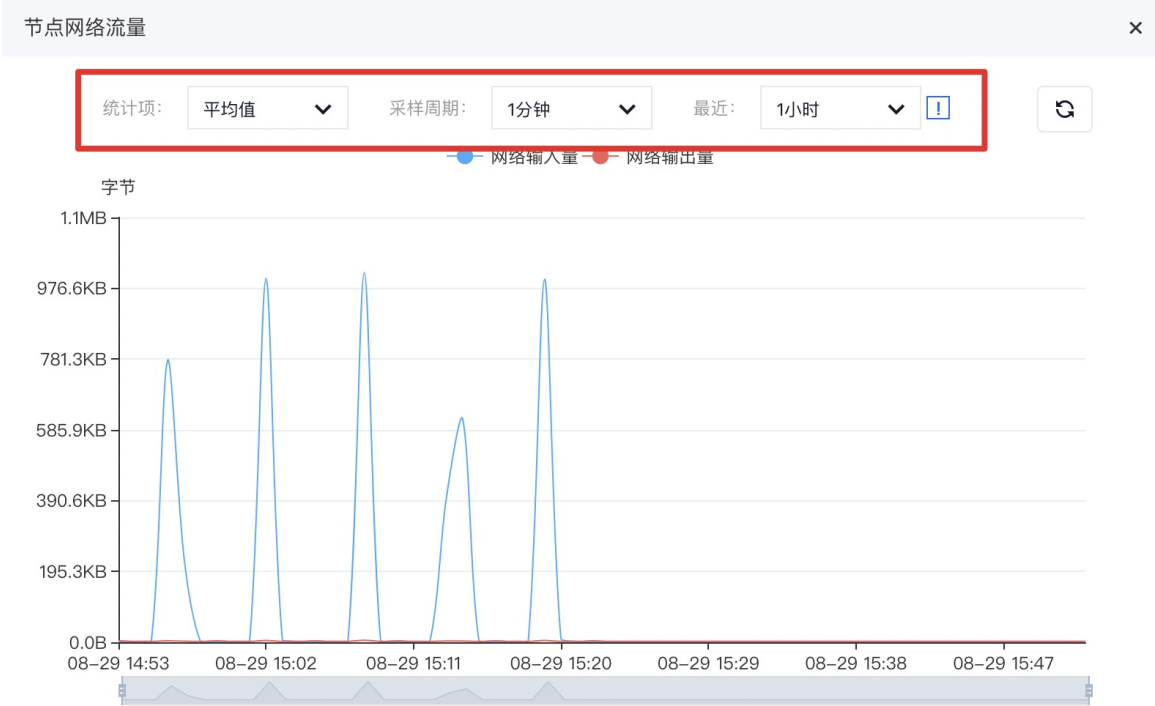
监控项包括：内存使用量、缓存命中率、节点网络流量、QPS、数据淘汰率、CPU使用率。

以"节点网络流量"监控项为例：

- 点击监控曲线图的缩放按钮，可查看详细信息。

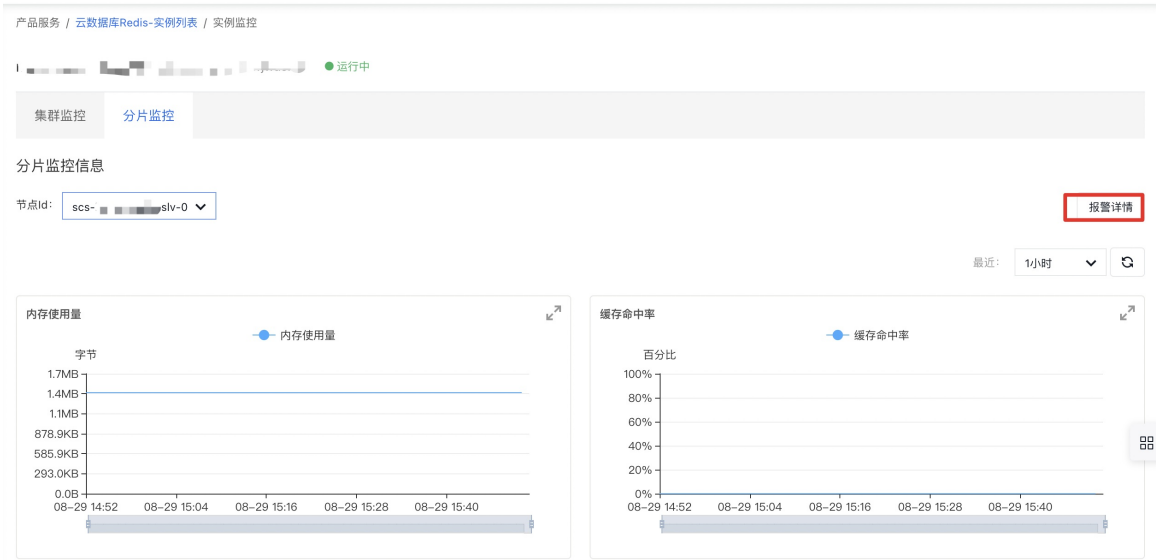


- 选择统计项、采样周期、统计时间，即可查看最新数据。



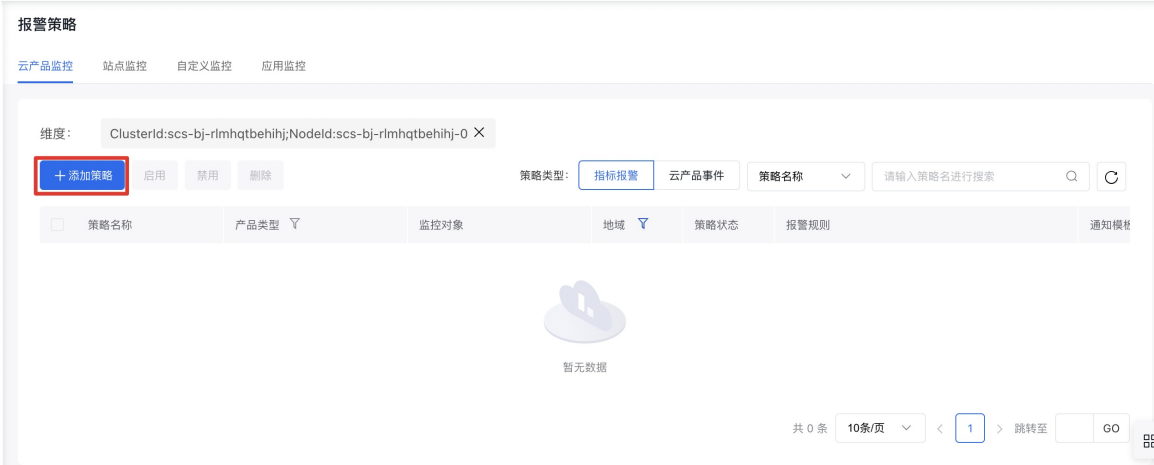
设置报警规则

- 1. 登录[云数据库 Redis 管理控制台](#)。
- 2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。
- 3. 找到目标 Memcache 实例，点击实例名称，进入实例详情页面。
- 4. 在左侧导航栏中单击监控管理。
- 5. 进入监控页面，在页面上方选择分片监控页签，进入分片监控页面。
- 6. 点击页面右上角的报警详情，即可进入实例的设置报警规则页面。



添加报警策略

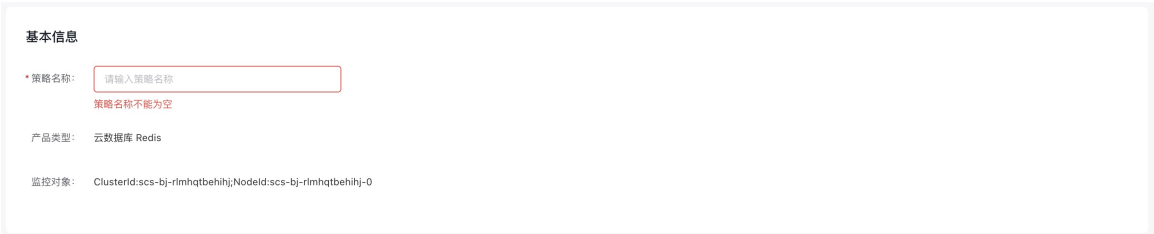
- 1. 点击添加策略可以添加报警策略，进入定义策略具体流程，添加策略信息。



2. 在此页面中可以创建报警策略，具体分为：基本信息、策略规则、报警通知三部分，具体如下。

基本信息

基本信息包含：策略名称：产品的类型：监控对象。其中策略名称用户自定义输入。产品类型默认为 云数据库Redis，监控对象为实例ID。



策略规则

策略规则包括策略类型、报警等级、报警规则，具体如下。策略类型：指标报警。

报警等级：通知、警告、重要、严重。

报警规则：选择的报警模版发生的次数，设置出发规则。

报警重复提醒：报警发生后如未恢复正常，触发/不触发，重复报警。

报警恢复处理：报警发生后如数据恢复正常，触发/不触发，恢复报警。

数据不足报警：监控数据不足时，触发/不触发，报警动作。



报警通知

添加报警模版

点击创建模版，进入创建报警模版的页面，填写以下参数。

参数	说明
动作名称	创建报警动作名称
可用地域	不同地域的报警动作仅适用于各自地域的报警策略
通知方式	百度智能云目前支持“邮件通知”，“短信通知”和“邮件+短信通知”的形式
通知对象	支持“用户组”和“用户”两种形式 若您当前没有已创建的通知对象，您可以点击“添加用户组”来进行创建
定时关闭	可选择是否定时关闭该报警动作

查看报警模版

查看已经新建的报警记录列表。

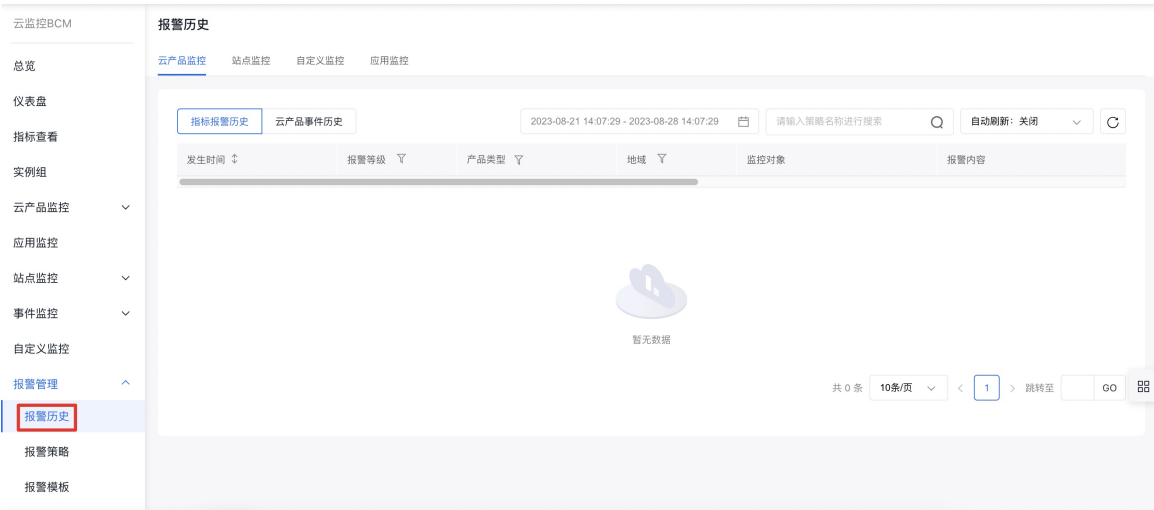


3. 点击保存按钮后，页面提示创建成功，报警动作列表显示新创建的报警模版。

说明：关于报警管理的具体信息，您也可以查看《BCM 操作指南 管理报警》。

查看报警历史

选择左侧导航栏的报警历史页签，进入报警历史页面，您可查看此实例产生的报警事件。



安全管理

设置IP白名单

概述

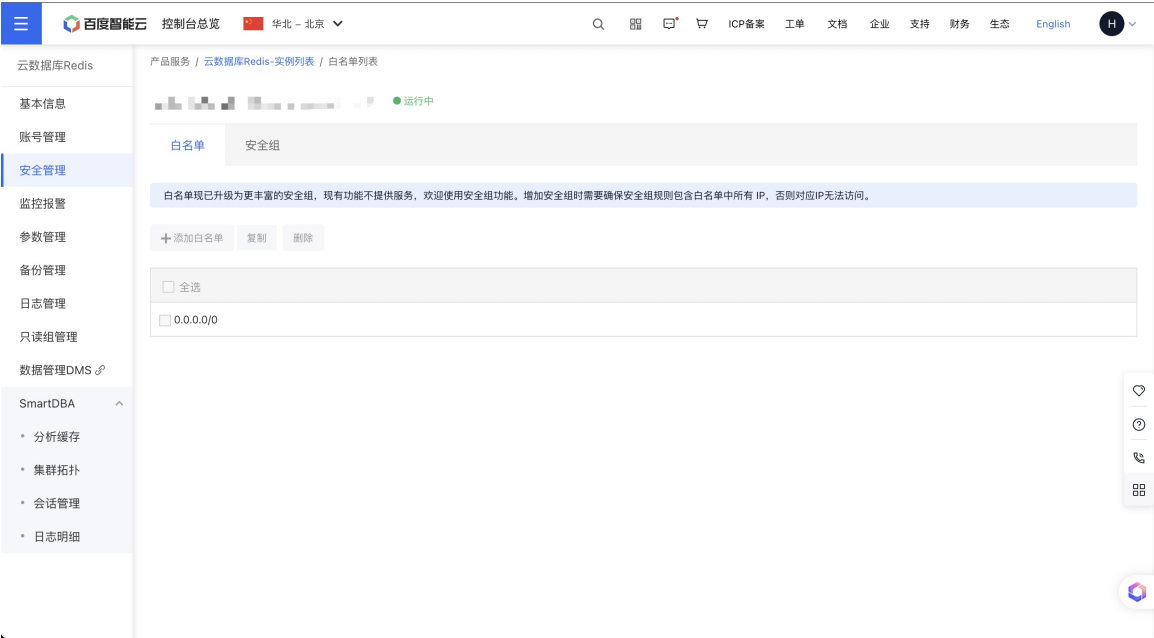
云数据库 Redis for Memcached 提供自定义 IP 白名单的功能，实现只允许白名单内的 IP 地址访问目标实例。本文为您介绍通过管理控制台对白名单进行设置。

说明：

- 支持常规地址: 如 192.168.0.1。
- 支持 CIDR 地址: 如 192.168.1.0/24。
- 多个地址以英文逗号或空格分隔，如 192.168.0.1,192.168.1.0/24。
- 0.0.0.0/0 代表允许所有地址。
- \* 代表允许所有地址。

注意事项

- IP 白名单功能将会逐步升级到安全组功能，提供更丰富的白名单管理功能，因此目前 IP 白名单编辑功能已经禁用。
- 对于已经设置了 IP 白名单的用户，可以把 IP 迁移到[安全组](#)。
- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。



安全组

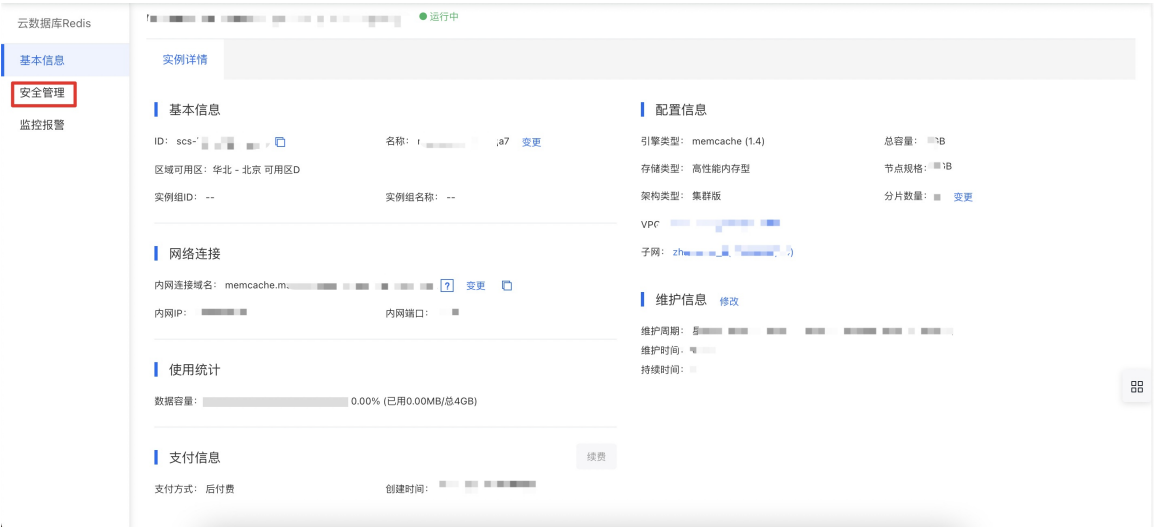
**概述** 查看绑定当前实例相关通信协议的出入站规则列表、关联该实例的安全组列表和绑定关联组列表。

注意事项

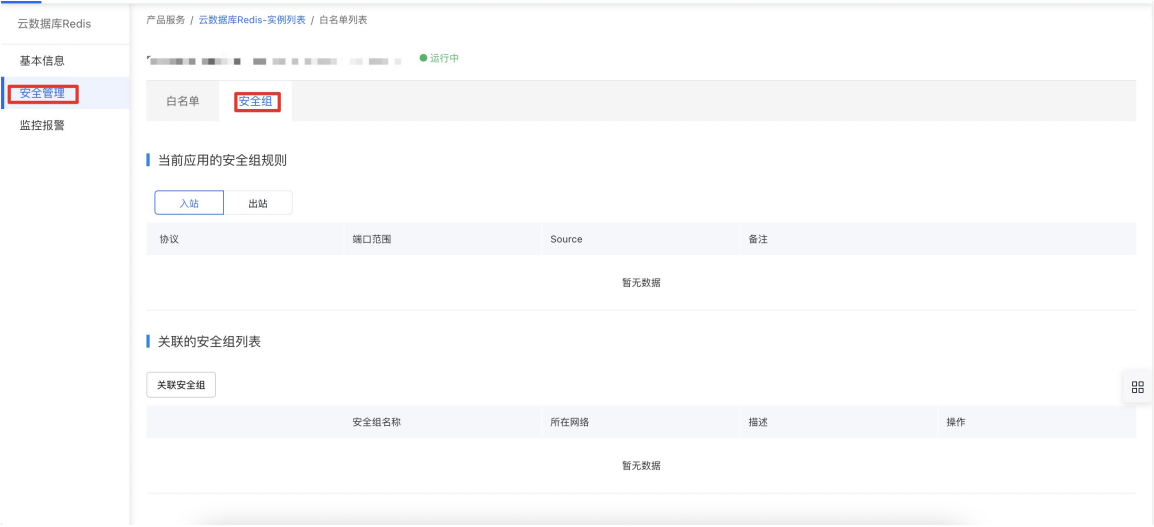
- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。

**\*\*操作指南\*\***

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **安全管理**，打开安全管理页面。



4. 默认选中白名单页签，点击安全组页签，切换至安全组页面。



5. 在安全组页面，查看当前应用的安全组规则列表、关联安全组列表两部分。

安全组规则列表

1) 在安全组页面的安全组规则列表区域，点击入站按钮，可以查看当前安全组入站的列表，具体字段包括协议、端口范围、Source、备注，详情如下图所示。



2) 在安全组页面的安全组规则列表区域，点击出站按钮，可以查看当前安全组入站的列表，具体字段包括协议、端口范围、Target、备注，详情如下图所示。



关联的安全组列表

1) 鼠标滑动至关联安全组列表区域，即可查看实例已经关联安全组列表，具体列表字段包括安全组名称、所在网络、描述、操作，详情如下图所示。



绑定的安全组列表

1) 在关联的安全组列表区域下方点击**关联安全组**按钮，弹出**关联安全组**对话框。



2) 在列表中、用户可以勾选多个**安全组**，点击**确定**按钮，即可完成绑定，并刷新**关联的安全组列表**列表。



Redis 容量型(原 PegaDB)操作指南

监控报警

自定义监控项（PegaDB）

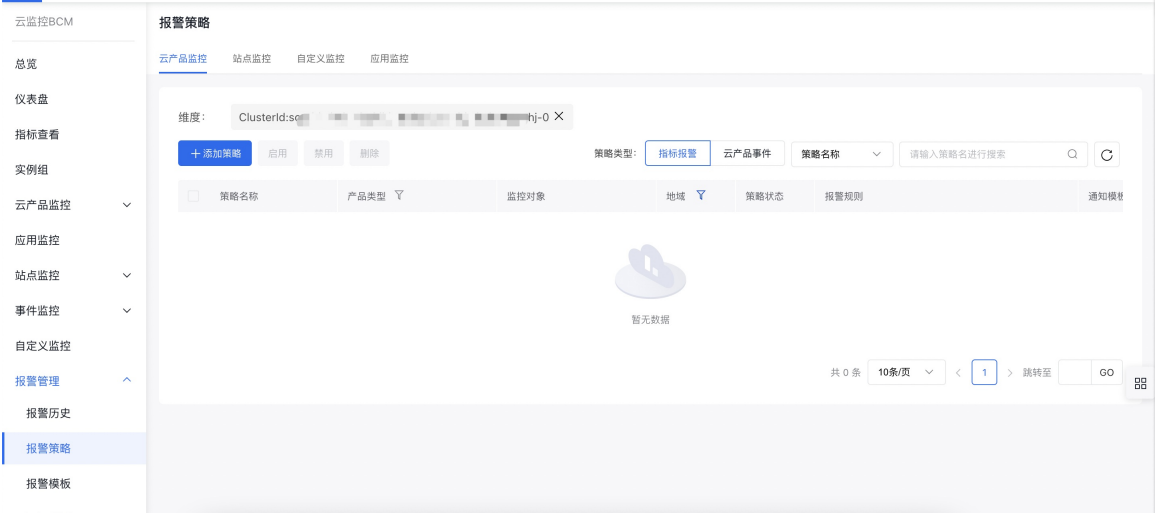
概述

云数据库Redis提供查看实例监控数据的功能，本文为您介绍 Redis容量型（PegaDB）实例支持的监控项及说明。

**\*\*集群监控\*\*** ||Redis控制台|单位| |-|-| |CPU|CPU使用率|百分比| |内存|内存使用量|字节| |内存使用率|百分比| |网络|网络输入量|字节| |网络输出量|字节| |缓存|缓存key数量|个| ||BlockCache命中率|百分比| ||命中率|百分比| |磁盘|磁盘读取IO|字节| |磁盘写入IO|字节| |数据空间使用率|百分比| |数据文件占用空间|兆字节| |访问|采样周期内处理命令数|次| |每秒处理命令数|次| ||QPS|次/秒| ||ReadQPS|次/秒| ||WriteQPS|次/秒| ||SetCmdQps|次/秒| ||GetCmdQps|次/秒| ||IncrCmdQps|次/秒| ||HgetCmdQps|次/秒| ||HsetCmdQps|次/秒| ||LpushCmdQps|次/秒| ||RpushCmdQps|次/秒| ||LpopCmdQps|次/秒| ||RpopCmdQps|次/秒| ||SaddCmdQps|次/秒| ||ScardCmdQps|次/秒| ||SismemberCmdQps|次/秒| ||ZaddCmdQps|次/秒| ||ZcardCmdQps|次/秒| ||ZcountCmdQps|次/秒| ||采样周期内慢查询次数|次/s| ||平均执行时延|us| ||P999执行时延|us| ||采样周期内读请求命中|次| ||采样周期内读请求Miss|次| |连接|实例连接数|个| ||连接使用率|百分比| ||最大连接数|个| ||采样周期拒绝连接数|个| |状态|实例状态|百分比| ||复制状态|个| **\*\*分片监控\*\*** ||Redis控制台|单位| |-|-| |CPU|CPU使用率|百分比| |内存|内存使用量|字节| |内存使用率|百分比| |网络|节点网络输入量|字节| |节点网络输出量|字节| |网络输入带宽使用率|百分比| |网络输出带宽使用率|百分比| |缓存|缓存key数量|个| ||命中率|百分比| ||BlockCache命中率|百分比| |磁盘|磁盘读取IO|字节| |磁盘写入IO|字节| |数据空间使用率|百分比| |数据文件占用空间|兆字节| |访问|采样周期内处理命令数|次| |每秒处理命令数|次| ||QPS|次/秒| |读请求命中|次| ||采样周期内读请求命中|次| ||采样周期内读请求Miss|次| ||采样周期内慢查询次数|次| |每秒慢查询次数|次| |连接|节点实例连接数|个| ||连接使用率|百分比| ||最大连接数|个| ||采样周期拒绝连接数|个| |状态|实例状态|百分比| ||复制状态|个| **代理监控** ||Redis控制台|单位| |-|-| |CPU|CPU使用率|百分比| |网络|网络输入量|字节| |网络输出量|字节| |访问|ProxyQPS|次/秒| ||Mget请求数|次/秒| |读请求|次/秒| |写请求|次/秒| |其他请求|次/秒| ||ReadQPS|次/秒| ||WriteQPS|次/秒| ||SetCmdQps|次/秒| ||GetCmdQps|次/秒| ||IncrCmdQps|次/秒| ||HgetCmdQps|次/秒| ||HsetCmdQps|次/秒| ||LpushCmdQps|次/秒| ||RpushCmdQps|次/秒| ||LpopCmdQps|次/秒| ||RpopCmdQps|次/秒| ||SaddCmdQps|次/秒| ||ScardCmdQps|次/秒| ||SismemberCmdQps|次/秒| ||ZaddCmdQps|次/秒| ||ZcardCmdQps|次/秒| ||ZcountCmdQps|次/秒| ||采样周期内慢查询次数|次/秒| ||平均执行时延|us| ||P99执行时延|us| ||P999执行时延|us| |连接|实例连接数|个| ||连接使用率|百分比| ||最大连接数|个| |状态|实例状态|百分比| **节点监控** ||Redis控制台|单位| |-|-| |CPU|CPU使用率|百分比| |内存|节点内存使用量|字节| |内存使用率|百分比| |网络|节点网络输入量|字节| |节点网络输出量|字节| |网络输入带宽使用率|百分比| |网络输出带宽使用率|百分比| |缓存|缓存key数量|个| ||命中率|百分比| ||BlockCache命中率|百分比| |磁盘|节点磁盘读取IO|字节| |节点磁盘写入IO|字节| |数据空间使用率|百分比| |数据文件占用空间|兆字节| |访问|采样周期内处理命令数|次| |每秒处理命令数|次| ||QPS|次/秒| ||采样周期内读请求命中|次| ||采样周期内读请求Miss|次| ||采样周期内慢查询次数|次| |每秒慢查询次数|次| |连接|节点实例连接数|个| ||连接使用率|百分比| ||最大连接数|个| ||采样周期拒绝连接数|个| |状态|实例状态|百分比| ||复制状态|百分比|

🔗 设置报警规则

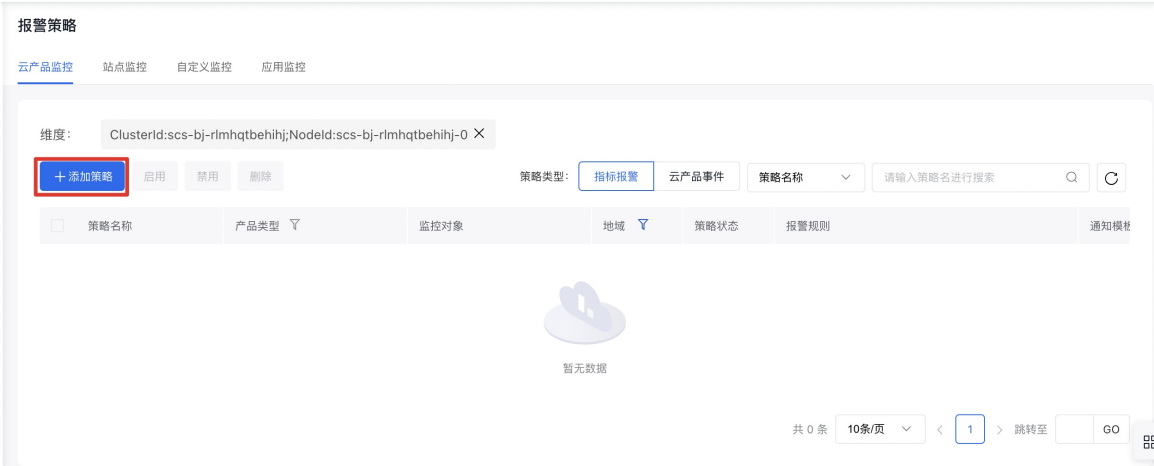
1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。
3. **集群版：**
  - (1) 找到目标实例集群版实例，点击**实例名称**，进入实例详情页面。
  - (2) 在左侧导航栏中点击**监控管理**，进入监控页面。
  - (3) 在页面上方选择**分片监控**、**代理监控**、**代理节点**，即可进入分片监控、代理监控、代理节点其中之一的页面。
  - (4) 点击页面右上角的**报警详情**按钮，即可进入实例的“**报警管理**”页面设置**报警规则**。



添加报警策略



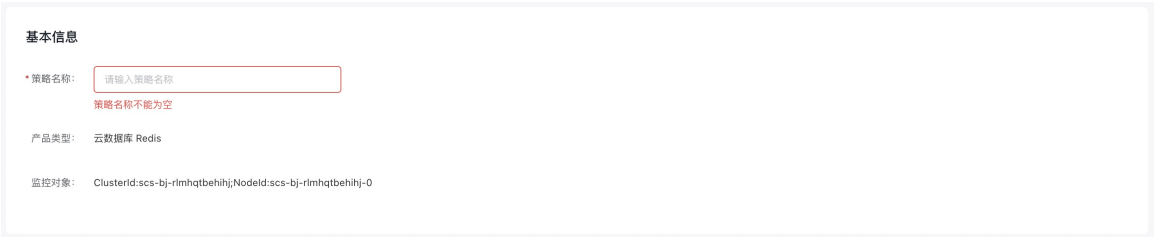
1. 选择报警策略页签，进入报警策略页面。



2. 在此页面中可以创建报警策略，具体分为：基本信息、策略规则、报警通知三部分，具体如下。

基本信息

基本信息包含：策略名称：产品的类型：监控对象。其中策略名称用户自定义输入。产品类型默认为云数据库Redis，监控对象为实例ID。



策略规则

策略规则包括策略类型、报警等级、报警规则，具体如下。策略类型：指标报警。

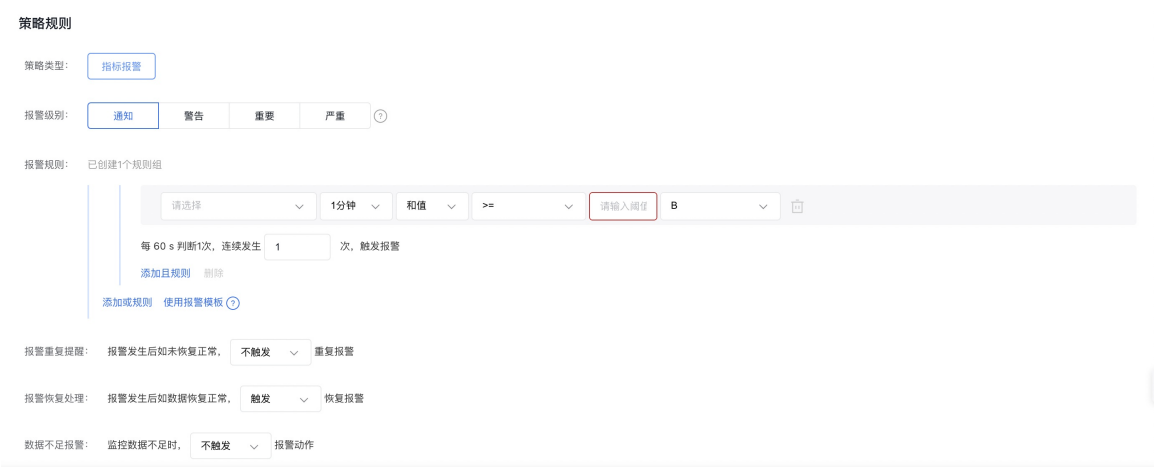
报警等级：通知、警告、重要、严重。

报警规则：选择的报警模版发生的次数，设置出发规则。

报警重复提醒：报警发生后如未恢复正常，触发/不触发，重复报警。

报警恢复处理：报警发生后如数据恢复正常，触发/不触发，恢复报警。

数据不足报警：监控数据不足时，触发/不触发，报警动作。



报警通知

添加报警模版

点击创建模版，进入创建报警模版的页面，填写以下参数。

参数	说明
动作名称	创建报警动作名称
可用地域	不同地域的报警动作仅适用于各自地域的报警策略
通知方式	百度智能云目前支持“邮件通知”，“短信通知”和“邮件+短信通知”的形式
通知对象	支持“用户组”和“用户”两种形式 若您当前没有已创建的通知对象，您可以点击“添加用户组”来进行创建
回掉地址	触发该模版之后的会调路径

查看报警模版

查看已经新建的报警记录列表。

报警通知

新建模版

共1条，已选 0 条，还可以选 5 条

请输入通知模版名称

Q

C

<input type="checkbox"/>	通知模版名称	用户/用户组	通知渠道	报警回调	定时关闭	更新时间	操作
暂无数据							

10条/页

<

1

>

跳转至

GO

3. 点击保存后，页面提示创建成功，报警动作列表显示新创建的报警模版。

说明：关于报警管理的具体信息，您也可以查看《BCM 操作指南 管理报警》。

查看报警历史

选择左侧导航栏的报警历史页签，进入报警历史页面，您可查看此实例产生的报警事件。

云监控BCM

总览

仪表盘

指标查看

实例组

云产品监控

应用监控

站点监控

事件监控

自定义监控

报警管理

报警策略

报警模版

报警历史

云产品监控

站点监控

自定义监控

应用监控

指标报警历史

云产品事件历史

2023-08-21 14:07:29 - 2023-08-28 14:07:29

请输入策略名称进行搜索

Q

自动刷新: 关闭

C

发生时间	报警等级	产品类型	地域	监控对象	报警内容
暂无数据					

共 0 条

10条/页

<

1

>

跳转至

GO

器

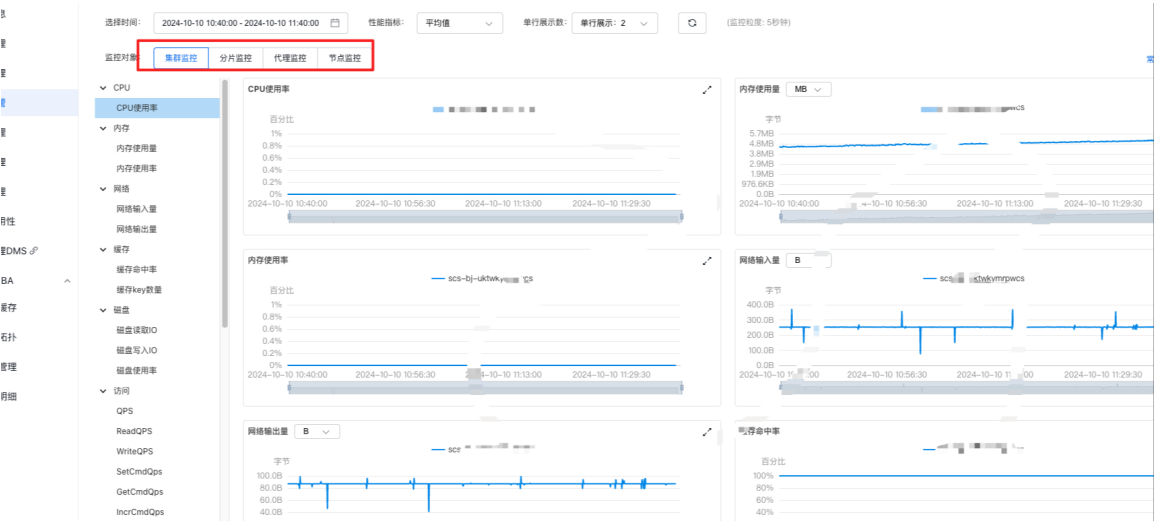
查看监控数据(PegaDB)

云数据库Redis版性能监控提供了CPU使用率、内存使用率、平均时延、QPS等性能监控指标。您可以查询过去指定时间段的监控数据，排查性能问题。

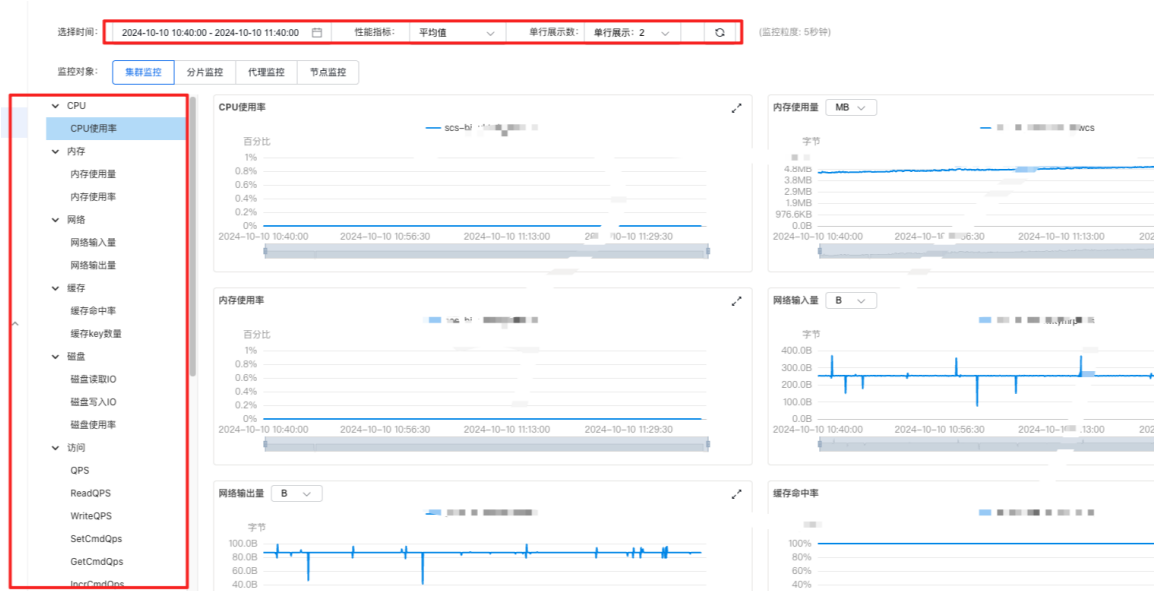
操作步骤

1. 登录云数据库 Redis 管理控制台。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

- 3. 找到目标实例，点击**实例名称**，进入实例详情页面。
- 4. 在左侧导航栏中单击**监控管理**。
- 5. 实例监控分为**集群监控**和**分片监控**、**代理监控**、**节点监控**页面。



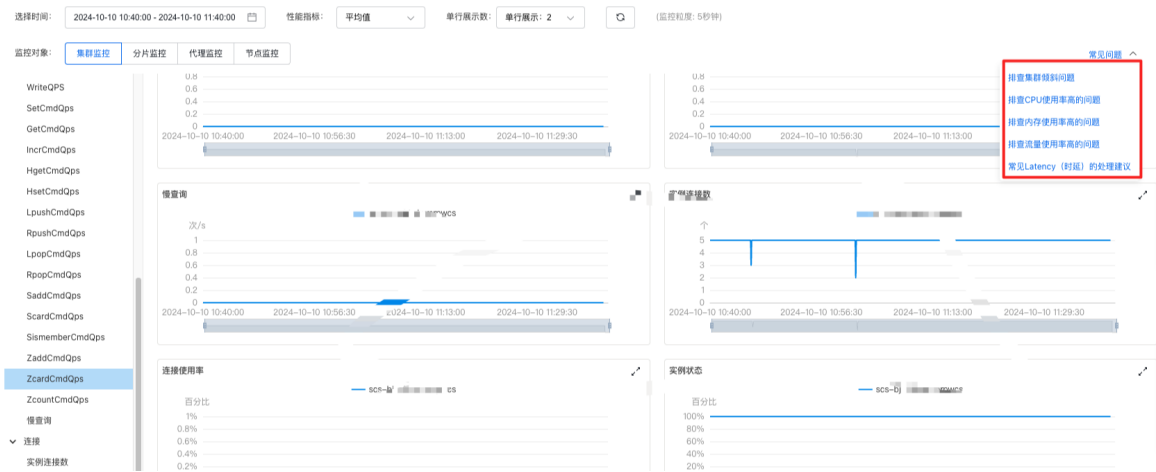
- 6.可筛选特定时间周期、性能指标计算方式后查看指标监控数据。



- 目前PegaDB支持的**监控指标项**。

问题排查

- 针对CPU打满、内存打满、流量打满等常见性能问题，提供排查方案，可点击[相关文档链接](#)进行查看。



账号管理

创建账号

概述

创建实例时，除系统默认创建的 default 账号外，您可根据业务的需要，通过控制台创建其他的业务账号。

注意事项

- 每个用户最多可创建50个账号，请进根据需要进行账号管理及分配。
- default账号是系统默认账号，不允许删除；允许清空密码，但有一定的风险，请谨慎操作。
- 用户账号是您创建的账号，允许删除，不允许清空密码。

\*\*创建账号规则\*\*

参数	说明
账号	填写账号名称。要求如下：     • 长度为2~16个字符。    • 以字母开头，以字母或数字结尾。    • 由小写字母、数字或下划线组成。
权限设置	该账号的数据库权限。     • 只读权限。    • 读写权限。
密码	• 长度为8~16个字符。    • 至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^*( )_+ =。
确认密码	再次输入密码。
备注说明	非必填，备注该账号的相关信息，便于后续账号管理。最多支持256个字符。

账号类型

云数据库 Redis 实例支持两种数据库账号：**默认账号**和**用户账号**。

您可以在控制台管理所有账号和数据库，账号拥有的具体权限请参见“账号权限列表”。

账号类型	说明
普通账号	- 可以通过控制台、API或者SQL语句创建和管理。   - 一个实例可以创建多个普通账号，具体的数量与实例内核有关。   - 需要手动给普通账号授予特定数据库的权限。   - 普通账号不能创建和管理其他账号，也不能断开其他账号的连接。

创建用户账号

1.

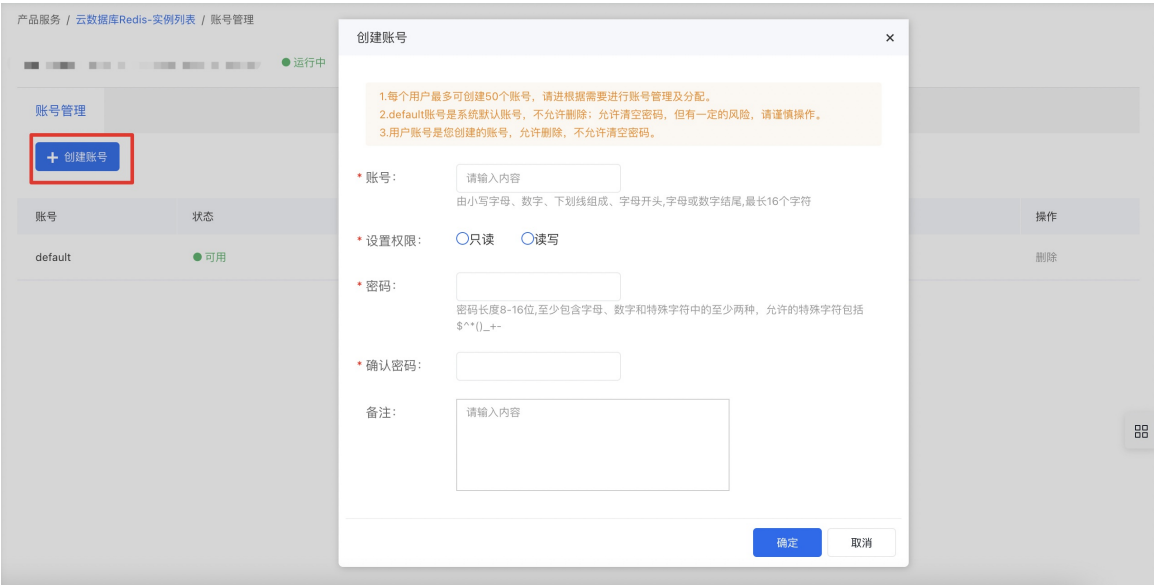
登录云数据库 Redis [管理控制台](#)。
2.

在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis 实例列表”页面。
3.

找到目标实例，点击**实例名称**，进入实例详情页面。
4.

在左侧导航栏中单击**账号管理**。
5.

在“账号管理”子页面，点击创建账号按钮，进入创建账号页面，设置参数。



6.

点击**确定**按钮，创建账号。

常见问题

1.

账号可以无限创建吗，账号数量上限是多少？  
答：账号不能无限创建，最多创建50个账号。

概述

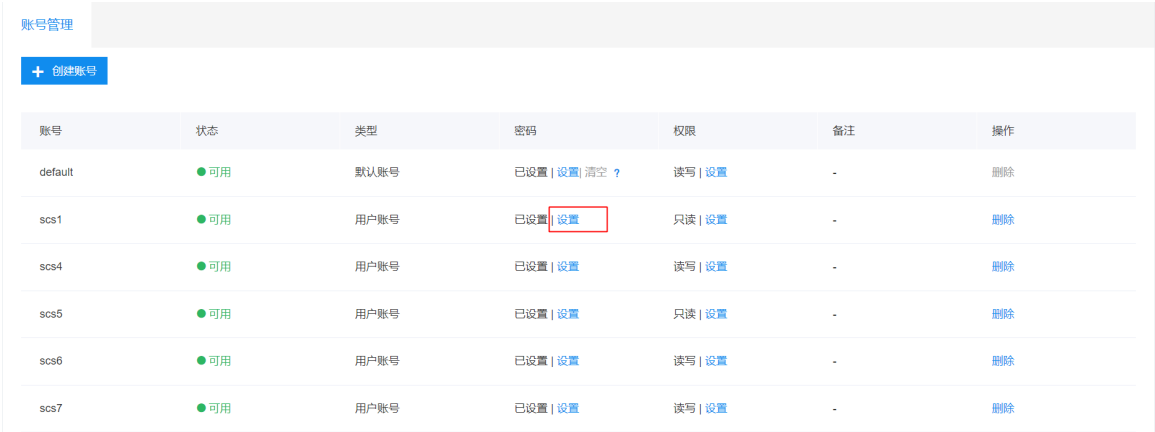
- 在使用实例的过程中，如果忘记数据库账号密码，可以通过控制台重新设置密码或清空密码。
- 为了数据安全，建议您定期更换密码。

修改密码规则

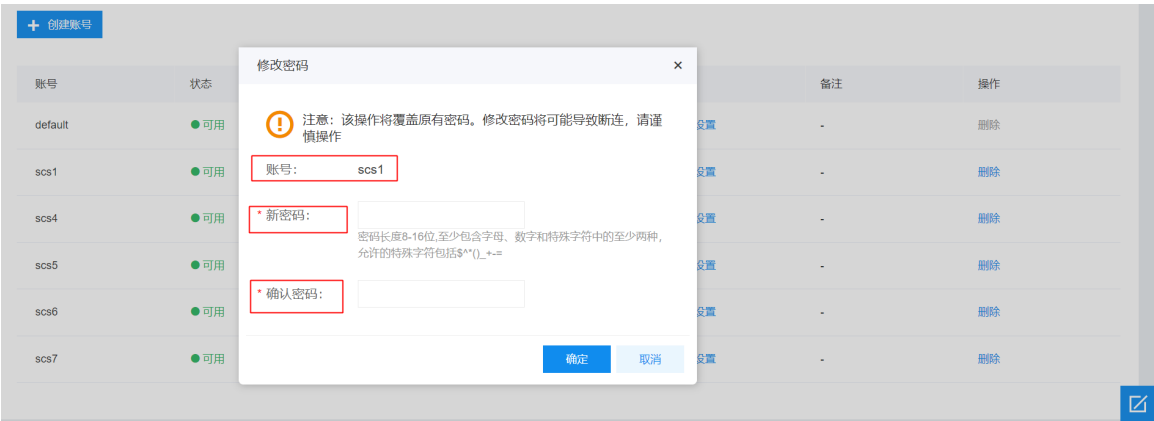
参数	规则	样例
密码	• 密码长度范围为8到16位字符。    • 至少包含字母、数字和特殊字符中的至少两种，允许的特殊字符包括\$^*()_+=。	Inst12#@

修改账号密码

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击实例名称，进入实例详情页面。
4. 在左侧导航栏中单击账号管理。
5. 找到要重置密码的账号，找到密码，然后点击设置。



6. 在弹出的对话框中，输入新密码并确认后，单击确定，提交修改。



说明：新的输入的密码参考修改密码规则。

修改账号权限

概述

您可以根据需要在控制台的账号管理页面修改用户账号的权限。

修改账号权限

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击实例名称，进入实例详情页面。
4. 在左侧导航栏中单击**账号管理**，选择目标账号，单击权限列的**设置**。

账号管理

+ 创建账号

账号	状态	类型	密码	权限	备注	操作
default	● 可用	默认账号	已设置   <a href="#">设置</a>   <a href="#">清空 ?</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs1	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs4	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs5	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs6	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs7	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>

5. 在设置权限页面中，调整账号权限，可将权限设置为只读或读写。
6. 单击**确定**按钮，提交修改。

设置权限

账号: scs1

权限: ☒ 只读  
☐ 读写

备注: 暂无

确定

取消

🔗 修改账号备注

概述

您在使用数据库账号时，可以通过控制台为账号添加/修改**账号备注**，便于后续账号管理。

修改账号备注

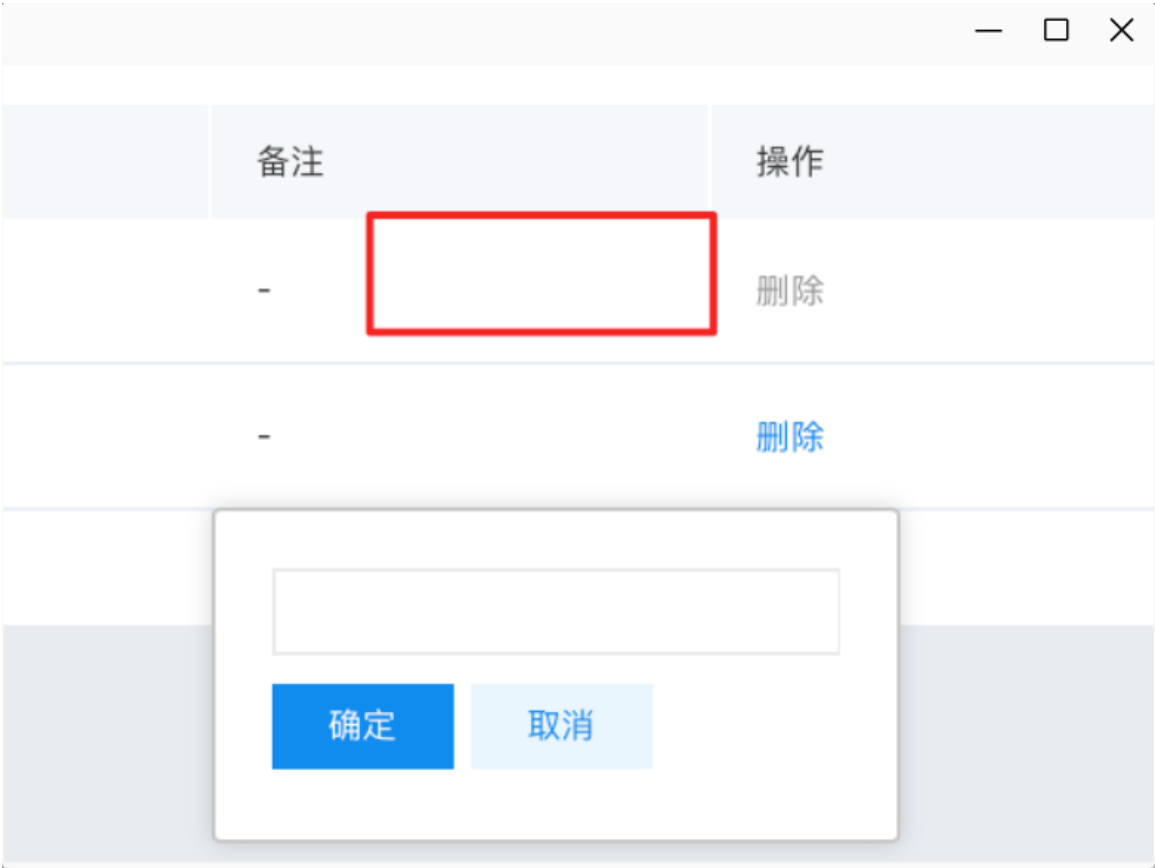
1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击实例名称，进入实例详情页面。
4. 在左侧导航栏中单击**账号管理**。
5. 在“账号管理”子页面，鼠标移动到账号行，“备注”列会出现手势标记。

账号管理

+ 创建账号

账号	状态	类型	密码	权限	备注	操作
default	● 可用	默认账号	已设置   <a href="#">设置</a>   <a href="#">清空 ?</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs1	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs4	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs5	● 可用	用户账号	已设置   <a href="#">设置</a>	只读   <a href="#">设置</a>	-	<a href="#">删除</a>
scs6	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>
scs7	● 可用	用户账号	已设置   <a href="#">设置</a>	读写   <a href="#">设置</a>	-	<a href="#">删除</a>

6. 出现手势标记后，点击即可出现编辑框，输入新的备注信息，点击**确认**按钮，即可保存。



🔗 删除账号

概述

若您需要对创建的数据库账号进行停用，可以通过控制台删除数据库账号。

注意事项

- 数据库账号删除后不可恢复，请谨慎操作。
- 避免因账号误删除而影响业务的正常使用，请确认即将删除的数据库账号不再对任何应用程序提供使用。

**\*\*删除账号\*\***

1. 登录云数据库 Redis [管理控制台](#)。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis 实例列表”页面。
3. 找到目标实例，点击**实例名称**，进入实例详情页面。



4. 在左侧导航栏中单击**账号管理**。
5. 到需要删除的账号，单击其右侧操作列中的**删除**。



6. 在弹出的删除账号对话框中，勾选风险提示，单击**确定**按钮，即可完成账号删除。



安全管理

开启免密访问

概述

云数据库 Redis 提供开启免密访问的功能（即清空密码），实现无须密码，即可访问目标实例。本文为您介绍通过管理控制台如何设置免费访问。

说明：

- 若进行清空密码，该操作将清空原有连接密码。允许无密码连接实例将可能存在安全风险，请谨慎操作。

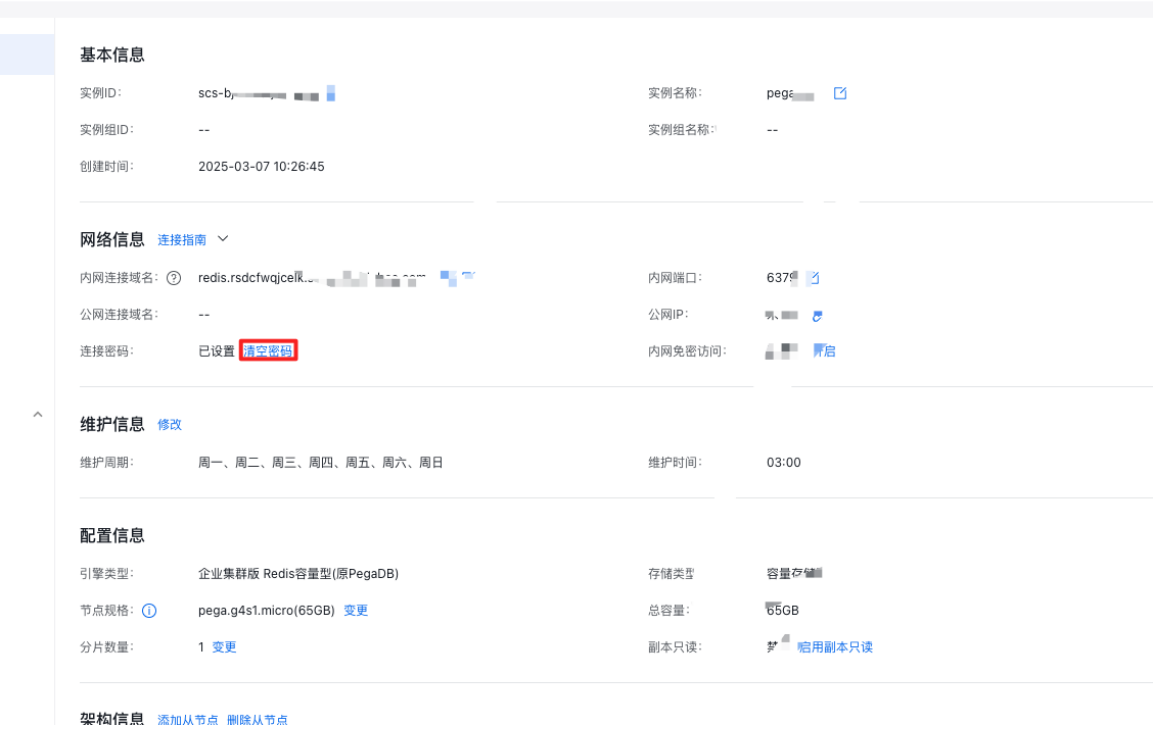
前提条件

- 实例的网络类型为私有网络（VPC）。
- 实例已设置连接密码，方可进行清空操作。
- 标准版和集群版实例均支持此功能

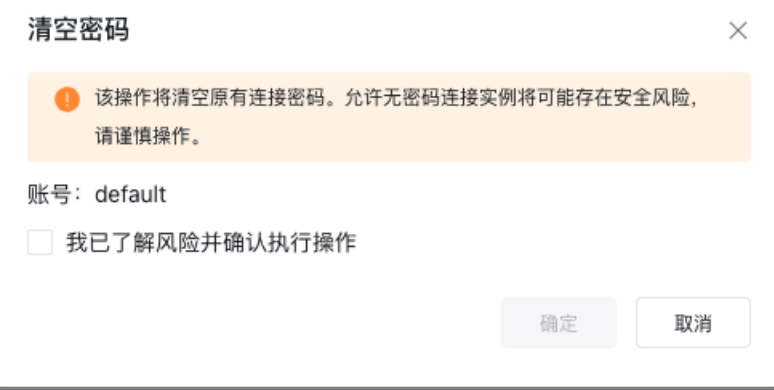
开启免密访问（清空密码）

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。

2. 点击目标实例名称，进入实例详情页面。在实例详情的网络连接模块，可以看到清空密码操作。



3. 点击 清空密码，在清空密码之前，您需要了解相关操作的风险并且同意之后，方可进行【确定】操作。



4.点击“确定”操作，实例的密码被清空。在实例详情可以看到，实例当前未设置连接密码，无法进行清空。

关闭免密访问（设置密码）

1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。在实例详情页面，可以看到【设置密码】按钮。

基本信息

实例ID: ...

实例名称: DC-...

实例组ID: --

实例组名称: ...

创建时间: 2025-07 10:26:45

网络信息

内网连接域名: rec-...

内网端口: 19

公网连接域名: --

公网IP: ...

连接密码: 未设置 设置密码

内网免密访问: 已关

维护信息

维护周期: 周一、周二、周三、周四、周五、周六、周日

维护时间: 3

配置信息

引擎类型: 企业集群版 Redis容量型(原PegaDB)

存储类型: ...

节点规格: ① pega.g4s1.micro(65GB) 变更

总容量: 65GB

分片数量: 1 变更

副本只读: 禁用 启用副本只读

附加信息

...

3. 点击 **设置密码**，根据要求设置连接密码，点击**确认**按钮后，连接密码即可成功设置。

设置密码

注意: 该操作将覆盖原有密码。修改密码将可能导致断连，请谨慎操作

\* 新密码:

密码长度8~16位，至少包含字母、数字和特殊字符中两种。允许的特殊字符包括\$^\*()\_+-

\* 确认密码:

取消

确定

TLS加密访问

概述

- 您可以通过控制台打开或关闭TLS 安全访问，还可以更新CA证书和下载CA证书。开启TLS后可以增加服务器安全访问机制。

注意事项

- TLS 仅支持企业版本。
- CA证书更新、CA证书下载必须是在TLS打开状态，方可操作。
- TLS 打开和关闭实例状态都将显示**配置变更中**。

开启 TLS 安全访问

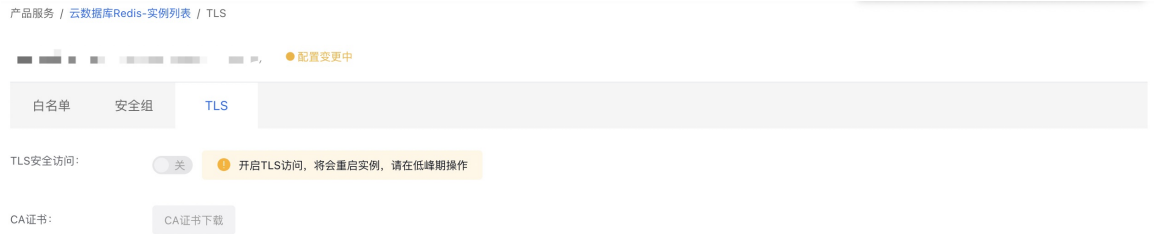
- 登录**云数据库 Redis** 管理控制台，进入实例列表页面。
- 点击目标**实例名称**，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击 **安全管理**，打开安全管理页面。

201

4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



5. TLS安全访问默认是关闭状态，并且CA证书无法下载，点击右侧开/关按钮，实例状态显示为配置变更中。



6.开通成功后，TSL 安全访问状态显示为开，并且会新增两行保护地址及证书有效期。

更新 CA 证书

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。
- 4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



5. 点击右侧更新证书按钮，弹出更新TLS证书有效期，点击确定按钮。



下载 CA 证书

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。

4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



5. 点击右侧下载CA证书按钮，将CA证书下载到本地。

关闭 TLS 安全访问

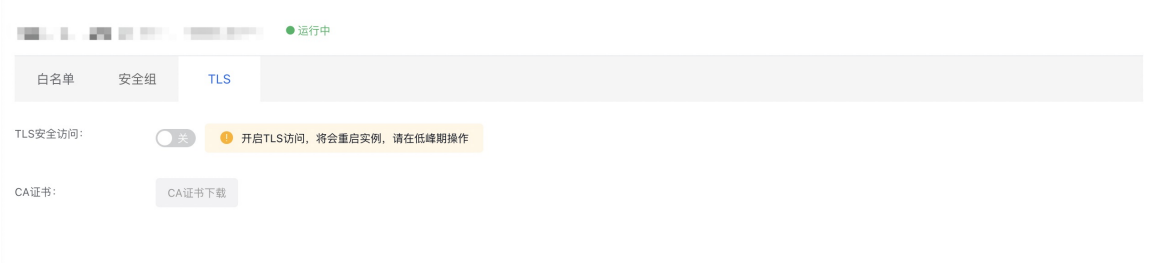
1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。
4. 默认选中白名单页签，点击TLS页签，切换至TLS页面。



5. 点击TLS安全访问右侧开/关按钮，实例状态显示为配置变更中。



- 6.关闭TLS安全访问后，实例状态显示为运行中。



安全组

概述

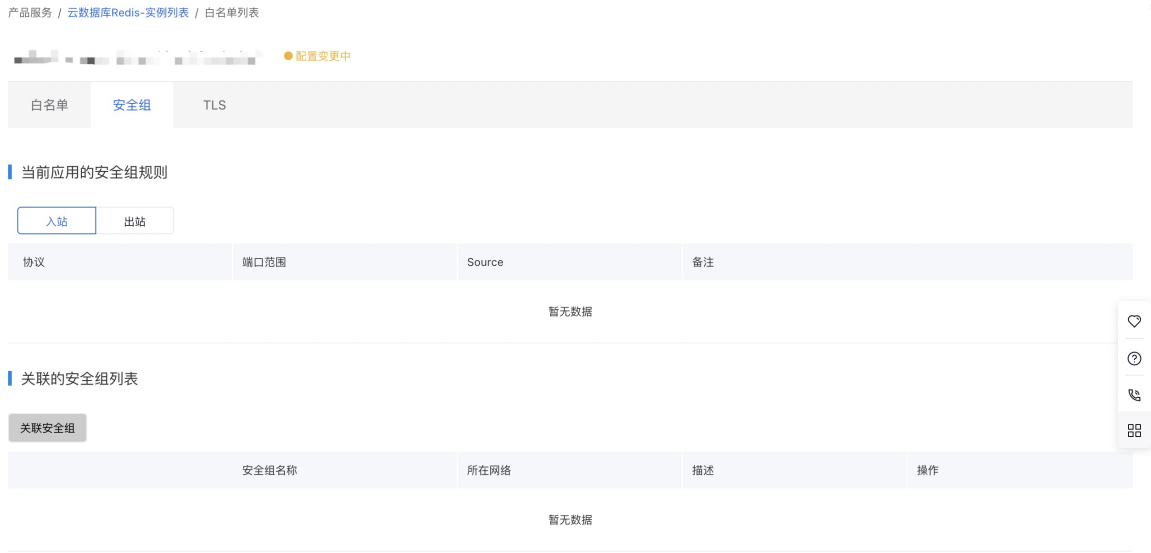
查看绑定当前实例相关通信协议的出入站规则列表、关联该实例的安全组列表和绑定关联组列表。

注意事项

- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。

\*\*操作指南\*\*

- 1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 安全管理，打开安全管理页面。
- 4. 默认选中白名单页签，点击安全组页签，切换至安全组页面。



- 5. 在安全组页面，查看当前应用的安全组规则列表、关联安全组列表两部分。

安全组规则列表

- 1) 在安全组页面的安全组规则列表区域，点击入站按钮，可以查看当前安全组入站的列表，具体字段包括协议、端口范围、Source、备注，详情如下图所示。



- 2) 在安全组页面的安全组规则列表区域，点击出站按钮，可以查看当前安全组入站的列表，具体字段包括协议、端口范围、Target、备注，详情如下图所示。



关联的安全组列表

- 1) 鼠标滑动至关联安全组列表区域，即可查看实例已经关联安全组列表，具体列表字段包括安全组名称、所在网络、描述、操作，详情如下图所示。



绑定的安全组列表

1) 在关联的安全组列表区域下方点击**关联安全组**按钮，弹出**关联安全组**对话框。



2) 在列表中中、用户可以勾选多个**安全组**，点击**确定**按钮，即可完成绑定，并刷新**关联的安全组列表**列表。



🔗 设置IP白名单

概述

云数据库 Redis 提供自定义 IP 白名单的功能，实现只允许白名单内的 IP 地址访问目标实例。本文为您介绍通过管理控制台对白名单进行设置。

说明：

- 支持常规地址: 如 192.168.0.1。

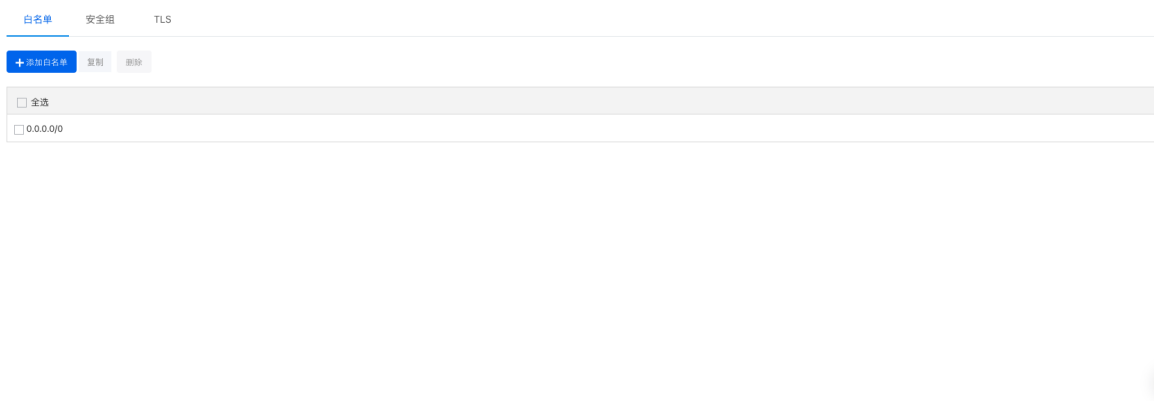
- 支持 CIDR 地址: 如 192.168.1.0/24。
- 多个地址以英文逗号或空格分隔，如 192.168.0.1,192.168.1.0/24。
- 0.0.0.0/0 代表允许所有地址。
- \* 代表允许所有地址。
- Redis社区版本不支持该功能。

注意事项

- IP 白名单功能将会逐步升级到安全组功能，提供更丰富的白名单管理功能，因此目前 IP 白名单编辑功能已经禁用。
- 对于已经设置了 IP 白名单的用户，可以把 IP 迁移到[安全组](#)。
- 增加安全组时需要确保安全组规则包含白名单中所有 IP，否则对应 IP 无法访问。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。
3. 在左侧导航栏，单击安全管理，找到IP白名单。



- 4.点击添加白名单，输入对应的IP地址，确认后即可添加成功。也可以选中对应白名单进行复制、删除等操作。



添加IP地址

IP地址：

支持常规地址: 如192.168.0.1

支持CIDR地址: 如192.168.1.0/24

多个地址以英文逗号或空格分隔，如192.168.0.1,192.168.1.0/24

0.0.0.0/0代表允许所有地址

确定

取消

节点变更

变更节点数量

概述

本文为您介绍通过管理控制台**变更节点数量**的通用方法，主从节点间具有复制关系。您可以根据业务量变动，灵活变更主备实例的从节点数量。本文介绍如何增加或减少单分片节点数。

前提条件

- 实例状态需为**运行中**。
- 集群版节点数量最少为2个。

增加从节点

- 登录[云数据库 Redis 管理控制台](#)。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
- 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块**添加从节点**按钮。

基本信息

账号管理

安全管理

监控报警

参数管理

备份管理

日志管理

服务可用性

数据开发(原DMS)

smartDBA

分析缓存

集群拓扑

会话管理

连接密码: 未设置 设置密码

维护信息 修改

维护周期: 周 周 周三、周四、 周六、周日

维护时间: 03:00

持续时间: 1小时

配置信息

引擎类型: 高可用版

版本: 6.2.3

存储类型: 高性能内存型

架构类: 集群

节点规格: (cache.m1.medium 4GB) 变更

副本数量: 1

副本只读: 禁用

数据容量: 1.00%(已用10.24MB/总1GB)

架构信息 添加从节点 删除从节点

分片ID	分片规格	副本数	分片操作
SCS-1	1GB	2	查看监控

4. 点击**添加从节点**进入“变更配置”页面，在“变更配置”模块，添加从节点信息，完成后续订单支付。

删除从节点

1. 登录[云数据库 Redis 管理控制台](#)。
2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
3. 在实例详情页的标签页导航栏中“实例详情”页，点击“架构信息”模块**删除从节点**按钮。



4. 点击**删除从节点**进入“变更配置”页面，在“变更配置”模块，**删除从节点**信息，完成后续订单支付。

实例生命周期

创建实例

创建 Redis 容量型(原 PegaDB)实例

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面左上角，选择实例所在**地域**，进入“云数据库 Redis-实例列表”页面。
3. 点击**创建实例**，在实例创建页面，选择计费方式：
  - **包年包月**：包年包月计费，采用预付费方式，按照开通实例规格和开通时长进行计费，价格较按需计费更低。
  - **按量付费**：按需计费，采用后付费方式，按照开通实例规格和使用分钟时长进行实时计费与扣费，计费方式较包年包月更灵活。
4. 设置以下参数。

参数	说明
地域	实例所在的地域，即实例所在的地理位置。如需修改为其他地域，请通过主导航进行切换。      ● 请选择靠近您客户的区域，可以降低网络时延、提高访问速度。创建成功后不支持更换地域    ● 仅允许相同私有网络内的设备访问本实例，请确保与需要连接的云服务器 BCC 实例创建于同一个私有网络。

	只支持选择 通用型 子网。创建后暂不支持更改。
数据库类型	当前支持Redis、Redis 容量型(原 PegaDB)、Memcache，这里选择Redis 容量型(原 PegaDB)。
存储类型	当前支持容量存储型、本地SSD型两种。容量存储型的存储资源为云磁盘、本地SSD型的存储资源为本地盘。
实例规格	用于保存数据的缓存节点的规格，不同规格的缓存节点提供不同的容量和性能。详细信息请参见 <a href="#">Redis 容量型(原 PegaDB)数据库</a> 。
存储磁盘	磁盘可选范围根据不同规格，会有差异，详情查看： <a href="#">节点规格</a>
分片数量	仅适用于<b>集群版实例</b>。     - 分片是一组缓存节点的集合，当实例包含多个分片时，数据集被切分到多个分片中，每个分片拥有完整数据集的子集。   - Redis容量型实例的总容量等于单个分片存储磁盘的容量与分片数量的乘积。
网络	网络即私有网络VPC。云数据库 Redis 实例仅允许相同 VPC 内的设备访问，请确保与需要连接的 BCC 实例创建于同一个 VPC。仅支持选择“通用型”子网。创建后暂不支持更改。
可用区	可用区地域内的独立物理区域。部分地域支持多可用区：     - 单可用区：可降低访问时延，但不再具有跨机房灾备优势。   - 多可用区：提高数据库可用性，但可能增加访问时延。
端口	端口可选1025-7041和7043-65534，默认6379，创建后支持变更。
实例名称	由大小写字母、数字以及-_/特殊字符组成，必须以字母或者中文开头，长度1-64。
购买信息	- 购买时长(包年包月)：选择所需的时长，系统会自动计算对应的配置费用，购买1年及以上有额外优惠。   - 购买数量：用于批量购买相同规格的实例。   - 自动续费（预付费）：   - 默认不勾选，不进行自动续费。

勾选后实例自动续费，自动续费周期与原订单周期一致。

5. 确认配置信息后，点击下一步按钮进入付款页面，可以在这里看到选择的实例的套餐和价格，进行确认。
6. 确认后点击去支付按钮，跳转进入“支付”页面，确认付款后订单完成。
7. 购买成功后，返回“云数据库 Redis-实例列表”页面，可看到已创建的实例，以及实例状态、位置、数据库类型等信息。其中，包年包月实例还可以看到 Redis 实例的到期时间。

常见问题

- 创建实例需要多长时间？  
答：创建实例消耗的时间与实例的分片数成正相关。实例的分片数越多，需要调配的资源越多，创建需要的时间则越长。

相关 API

- API 接口：[创建实例](#)

释放实例

概述

本文为您介绍通过管理控制台进行释放实例的通用方法。

释放实例

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台页面的“云数据库 Redis-实例列表”页面，选择需要释放实例名称/ID。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

搜索

刷新

下载

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量（已用/总量）	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input checked="" type="checkbox"/> f-... 集群	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> e-... 集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> s-... 集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> a-... 集群	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.mediu (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/> s-... 集群	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

3. 点击释放按钮，弹出释放确认界面。

释放实例



确定要释放以下实例吗？

redis-llxeeg4s

确定

取消

说明：预付费的SCS实例无法提前释放。

4. 点击**确定**按钮，弹出“安全验证页面”，进行手机验证码二次验证。

安全验证

为了保护您的账户安全，请通过安全验证：

您绑定的手机： \*\*\*\*\* 更换手机

短信验证码：

发送验证码

您当前的操作需要进行二次验证

确定

取消

5. 输入**短信验证码**，点击**确认**按钮，完成释放操作。

相关 API

- API 接口：[释放实例](#)

☞ 续费实例

☞ 自动续费

概述

**自动续费：**您可在控制台上设置预付费实例进行自动续费。若预付费实例到期未续费，会导致业务中断甚至数据丢失。

- 使用场景：适用于长期使用云数据库 Redis 实例的场景。
- 优点：操作简便，避免因遗忘续费导致数据删除业务中断的情况，安全性更高。

注意事项

- 开通自动续费后，系统会在到期日前7天自动为其续费。
- 请保证百度智能云账号的余额充足，避免续费失败。续费成功或失败都给予短信和邮件提醒。
- 自动续费将按照您最近一次设置的续费周期执行。
- 在一个周期内，资源的到期时间由于其他操作而延长，自动续费将在下个到期日再执行。
- 自动续费执行时，会针对某一个实例资源生成一个续费订单。

创建实例时开通

您可以在购买预付费实例时开通**自动续费**，自定义续费周期，选择按月（1~9个月）或按年（1~3年）进行自动扣费。

购买信息

购买时长:

1个月

2

3

4

5

6

7

8

9

Sale1年

Sale2年

Sale3年

实例个数:

-

1

+

最多可购买10个SCS实例，还可购买9个

自动续费:

ON

什么是自动续费?

选择续费周期:

按年

1年

按月

按年

系统将于到期前7天进行扣费，扣费时长为1年

创建实例后开通

1. 登录[云数据库 Redis 管理控制台](#)，进入“云数据库 Redis-实例列表”页面。

2. 在控制台右上方，选择财务菜单项的子菜单项续费管理。

3. 在续费管理页面选择到期时间，点击云数据库Redis 页签，选择实例所在地域，找到目标实例。

续费管理

操作指南

到期时间:

全部时间

已到期

7天内到期

15天内到期

30天内到期

3个月内到期

产品筛选:

云服务器 BCC(19)

云数据库DocDB for MongoDB(1)

VPN网关 VPN(2)

云数据库SCS for Redis(1)

弹性公网IP EIP(10)

云磁盘 CDS(4)

云数据库RDS(1)

应用防火墙 WAF(1)

地域选择:

华北 - 北京 (1)

续费

设置自动续费

关闭自动续费

实例名称

请输入实例名称进行搜索

实例名称

<input checked="" type="checkbox"/>	实例名称	实例ID	到期时间	自动续费/续费周期	下次自动续费日期
<input checked="" type="checkbox"/>	redis-k104quhg		2019-10-26 11:26:19, 已过期261天	未开通	-

每页展示

15

<

1

>

4. 勾选目标实例，点击实例名称上方的设置自动续费按钮。

5. 在弹出的设置自动续费窗口中选择续费周期（按年或按月），点击确定按钮。

产品筛选:

地域选择:

续费

实例

redis

设置自动续费

自动续费说明:

1. 开通自动续费后，系统会在到期日前7天自动为其续费。

2. 自动续费将按照您最近一次设置的续费周期执行。

3. 在一个周期内，资源的到期时间由于其他操作而延长，自动续费将在下个到期日再执行。

4. 自动续费执行时，会针对某一个实例资源生成一个续费订单。

已选实例:

您已选1个实例 展开详情

续费周期:

按月

1个月

按月

按年

确定

取消

6. 自动续费开通成功，系统会发送短信和邮件通知。

手动续费

概述

手动续费：默认续费方式，适用于初期使用且有运维人员按时进行续费操作的场景。

• 缺点：续费周期即将结束前需要手动续费，操作较为繁琐，可能遗忘续费而导致实例停止服务或数据丢失。

212

- 解决方案：设置自动续费续费周期，设置一次永久生效。

说明：

- 预付费实例：有到期时间。若到期未续费，会导致业务中断甚至数据丢失。您可以根据业务需要，对“预付费”实例进行续费。
- 后付费实例：无到期时间，无须进行续费相关操作。

Redis 控制台续费实例

1. 登录[云数据库 Redis 管理控制台](#)，进入“云数据库 Redis-实例列表”页面。
2. 选择需要续费的实例，点击此实例操作栏中的续费选项，进入“财务中心／续费”页面。
3. 在批量选择续费时长选项中，先选择按月或按年进行续费，再确定具体续费时间长度。

续费

1 续费时长 > 2 确认订单 > 3 订单支付 > 4 续费成功

批量选择续费时长: 

按月 1个月

ID	产品名称	到期时间	续费时长	续费后到期时间	金额
	云数据库SCS for Memcached	2019-10-26 11:26:19	按月 1个月	2020-08-16 15:12:02	¥ 50.00

请确认续费时长

取消

下一步

注意：选择按年续费，最多续费3年；按月续费，最多续费9个月。

4. 确定好续费时间和续费金额后，点击右下角下一步按钮，进行订单付费，订单支付后，正式完成续费操作。
5. 完成续费后，返回“云数据库 Redis-实例列表”页面，可在支付方式下看到SCS实例的到期时间已更新。

续费管理续费实例

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在控制台右上方，选择 财务菜单项的子菜单项续费管理。

华北 - 北京

Q 合作伙伴 工单 消息 帮助文档 企业资源 财务 B

云数据库 SCS

产品服务 / 云数据库 SCS-实例列表

实例列表

+ 创建实例 释放 计费变更/取消 编辑标签 实例名称

实例名称/ID	状态	可用	IP地址	支付方	数据容量 (已用/总量)	引擎类型	副本数量	标签	操作
---------	----	----	------	-----	--------------	------	------	----	----

余额: 0.1 充值 发票

消费总览 收支明细 订单管理 续费管理

3. 进入续费管理页面，通过“到期时间、产品筛选、地域选择”筛选栏，找到目标续费实例。
4. 勾选一个或多个实例，点击续费按钮，进入“财务中心／续费”页面，即可进行单个或批量实例续费。

续费管理 操作指南

到期时间: 

全部时间 已到期 7天内到期 15天内到期 30天内到期 3个月内到期

产品筛选: 

云服务器 BCC(19) 云数据库DocDB for MongoDB(1) VPN网关(2) **云数据库SCS for Redis(1)** 弹性公网IP EIP(10) 云磁盘 CDS(4)

云数据库RDS(1) 应用防火墙 WAF(1)

地域选择: 

华北 - 北京 (1)

**续费** 设置自动续费 关闭自动续费

实例名称 请输入实例名称进行搜索 🔍 🔄

<input checked="" type="checkbox"/>	实例名称	实例ID	到期时间	自动续费/续费周期	下次自动续费日期
<input checked="" type="checkbox"/>			2020-07-15 11:26:19, 已过期2天	未开通	-

5. 在批量选择续费时长选项中，先选择按月或按年进行续费，再确定具体续费时间长度。

续费 1 续费时长 > 2 确认订单 > 3 订单支付 > 4 续费成功

批量选择续费时长: 

按月 1个月

ID	产品名称	到期时间	续费时长	续费后到期时间	金额
	云数据库SCS for Memcached	2020-07-15 11:26:19	按月   1个月	2020-08-16 15:19:31	¥ 50.00

请确认续费时长 取消 下一步

注意：选择按年续费，最多续费3年；按月续费，最多续费9个月。

6. 确定好续费时间和续费金额后，点击右下角下一步按钮，进行订单付费，订单支付后，正式完成续费操作。
7. 完成续费后，返回“云数据库 Redis-实例列表”页面，可在支付方式下看到 Redis 实例的到期时间已更新。

计费变更

☞ 预付费转后付费

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍预付费转后付费的操作。

注意事项

- 已选择了自动续费的预付费实例不能进行计费变更。
- 运行状态为已到期、重启中、迁移中的实例不支持预付费转后付费。
- 用户将预付费计费方式变更为后付费时需要确保余额充足（余额>100），避免设置生效时余额检查失败。

\*\*预付费转后付费\*\*

- 登录云数据库 Redis 管理控制台。
- 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。



实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

Q

刷新

下载

自定义列

<input type="checkbox"/>	实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (60GB)	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4) (2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4) (2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis	cache.n1.medi (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。
4. 点击计费变更/取消按钮，进入计费变更页面，对实例进行计费模式的变更。

+ 创建实例

释放

计费变更/取消

实例名称

Q

刷新

下载

自定义列

<input type="checkbox"/>	实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input checked="" type="checkbox"/>	redis-llxho... scs-bj-zgln... 集群	运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (60GB)	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4) (2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4) (2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis	cache.n1.medi (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxho...  scs-bj-zgln...  集群	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a>

5. 确认订单后，点击下一步按钮，按照提示支付订单。

财务中心 / 计费变更

1 计费变更

2 确认订单

3 订单支付

4 变更成功

计费变更-预付费转后付费

ID	产品名称	变更前单价	变更前计费方式	变更后单价	变更后计费方式	变更后生效时间	金额
redis-llxho...  scs-bj-zgln...  集群	云数据库Redis	¥ 1597.00/月/台	包年包月	¥ 0.06791925/分钟/台	按时长	2023-09-30 16:48:04	¥ 0.06791925/分钟/台

取消

下一步

6. 预付费转变为后付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。
7. 在实例列表的右上角，单击刷新列表，可看到实例状态显示为运行中。“支付方式”显示为预付费(待变更)。实例的后付费计费方式需要等预付费到期后才会生效，未到前可以取消计费变更。到期后支付方式会显示为后付费。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称 

▼

🔍

🔄

📄

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 (待变更)  📅  30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-30 16:45:54	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (50GB)	-	--	2023-08-30 15:11:42	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	后付费	0.00MB/60GB	企业版	pega.g4s1.micro	-	--	2023-	续费

取消计费变更

用户在进行预付费转后付费后，预付费未到期前可以取消计费变更，继续使用预付费的计费方式。

1. 登录云数据库 Redis 管理控制台。
2. 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称 

▼

🔍

🔄

📄

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 (待变更)  📅  30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-30 16:45:54	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (50GB)	-	--	2023-08-30 15:11:42	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	续费 清空数据 编辑资源分组
<input type="checkbox"/>  实例名称	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	续费 清空数据 编辑资源分组

3. 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量取消计费变更。
4. 点击计费变更/取消按钮。

产品服务 / 云数据库Redis-实例列表

实例列表

自动续费

创建实例

释放

计费变更/取消

实例名称

搜索

刷新

下载

自定义列

实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
实例名称/ID      集群     运行中	可用区D	预付费 (待变更) 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>	
实例名称/ID      集群     运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>	
实例名称/ID      集群     运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>	
实例名称/ID      集群     运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>	
实例名称/ID      集群     运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>	

5. 在弹出的取消计费变更对话框中点击确定按钮。

取消计费变更



确认取消计费变更?

您已开通计费变更-预付费转后付费功能。

实例将会在到期后自动转换为后付费的计费方式。

确定

取消

6. 可以看到计费变更已取消的消息。

7. 取消计费变更成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

产品服务 / 云数据库Redis-实例列表

计费变更已取消 (4s) X

自动续费

实例列表

+ 创建实例

释放

计费变更/取消

实例名称

Q

刷新

下载

自定义列

实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
集群	运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micrr (60GB)	-	--	2023-08-30 16:45:54	续费   清空数据   编辑资源分组
集群	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micrr (50GB)	-	--	2023-08-30 15:11:42	续费   清空数据   编辑资源分组
集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	续费   清空数据   编辑资源分组
集群	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	续费   清空数据   编辑资源分组
集群	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	续费   清空数据   编辑资源分组
配置变更中	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%)	企业版 PegaDB	pega.g4s1.micrr (60GB)	-	--	2023-08-16	续费   清空数据

8. 在实例列表的右上角，单击  刷新列表，可查看到实例状态显示为运行中。“支付方式”显示为预付费。

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

▼

🔍

🔄

⬇️

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

🔗 后付费转预付费

概述

您可以根据业务需求，变更实例的计费方式。以下将介绍后付费转预付费的操作。

注意事项

- 运行状态为已到期、重启中、迁移中的实例不支持后付费转预付费。

**\*\*后付费转预付费\*\***

- 登录[云数据库 Redis 管理控制台](#)。
- 在控制台页面左上角，选择实例所在地域，进入“云数据库 Redis-实例列表”页面。

实例列表

自动续费

+ 创建实例

释放

计费变更/取消

实例名称

▼

🔍

🔄

⬇️

自定义列

<input type="checkbox"/> 实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (50GB)	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache (1.4)	(2GB)	-	--	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis (6.0)	cache.n1.medium (4GB)	-	--	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>   集群	● 配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micro (60GB)	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

- 找到目标实例，勾选实例名称/ID前的复选框，也可多选实例进行批量计费变更。
- 点击计费变更/取消按钮，进入计费变更页面，对实例进行计费模式的变更。

实例列表

+ 创建实例

释放

计费变更/取消

实例名称

搜索

刷新

下载

自定义列

<input type="checkbox"/>	实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D+可用区E	后付费	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr...	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 30天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4)	(2GB)	-	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4)	(2GB)	-	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis	cache.n1.medi...	(4GB)	-	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	配置变更中	可用区D	后付费	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr...	-	--	2023-08-16 14:48:38	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

5. 选择预付费购买时长，可以按月购买或按年购买，最小预付费时长为一个月。

6. 确认订单，点击下一步按钮，按照提示支付订单。

计费变更-后付费转预付费

1 计费变更

2 确认订单

3 订单支付

4 变更成功

预付费购买时长: 按月 1个月

ID	产品名称	变更前单价	变更前计费方式	时长	变更后单价	变更后计费方式	变更后到期时间	金额
	云数据库Redis	¥ 0.093716125/分钟/台	按时长	按月 1个月	¥ 2168.50/月/台	包年包月	2023-09-30 16:50:40	¥ 2168.50

请确认购买时长

取消

下一步

7. 后付费转变为预付费成功后，用户可以在“云数据库 Redis-实例列表”页面对其进行查看和管理。

8. 在实例列表的右上角，单击 刷新列表，可查看到后付费转预付费完成后，实例状态显示为运行中。“支付方式”显示为预付费。

产品服务 / 云数据库Redis-实例列表

实例列表

+ 创建实例

释放

计费变更/取消

实例名称

搜索

刷新

下载

自定义列

<input type="checkbox"/>	实例名称/ID	状态	可用区	支付方式	数据容量 (已用/总量)	引擎类型	节点规格	部署集	所属实例组	创建时间	操作
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 30天后到期	0.00MB/60GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr...	-	--	2023-08-30 16:45:54	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D+可用区E	预付费 30天后到期	0.00MB/50GB (0.00%) 容量存储型	企业版 PegaDB	pega.g4s1.micr...	-	--	2023-08-30 15:11:42	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4)	(2GB)	-	2023-08-29 14:43:04	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llxee... scs-bj-weo...	运行中	可用区D	预付费 29天后到期	0.00MB/4GB (0.00%) 高性能内存型	memcache	(1.4)	(2GB)	-	2023-08-29 10:14:08	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>
<input type="checkbox"/>	redis-llq1c... scs-bj-rim...	运行中	可用区D	预付费 25天后到期	0.00MB/4GB (0.00%) 高性能内存型	社区版 redis	cache.n1.medi...	(4GB)	-	2023-08-25 11:31:36	<a href="#">续费</a> <a href="#">清空数据</a> <a href="#">编辑资源分组</a>

变更实例

主从切换

概述

云数据库 Redis 提供手动主从切换的功能。本文为您介绍通过管理控制台进行主从切换的通用方法。

注意事项

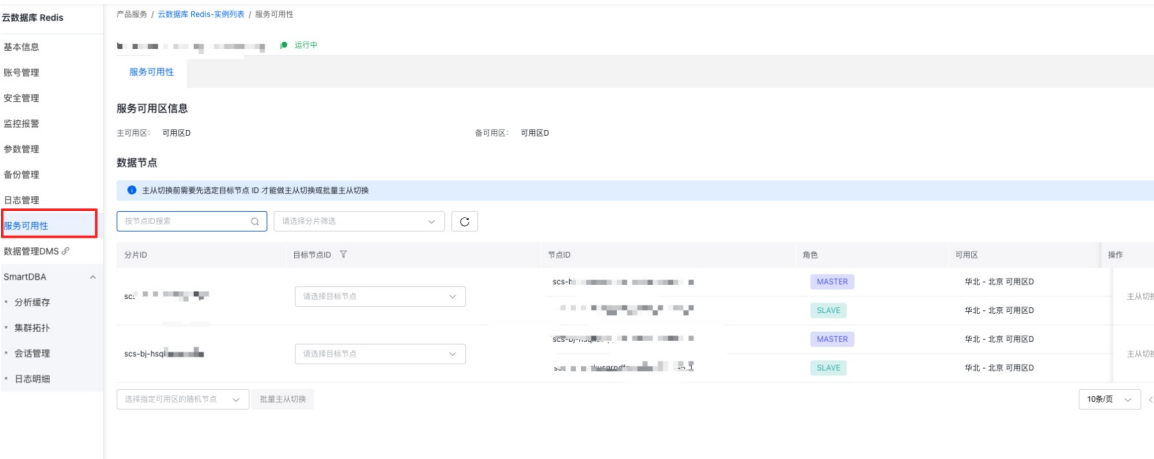
- 主节点角色将移交到其他可用区的从节点。如果所有节点都位于同一个可用区，则不支持该操作。
- 主从切换过程中可能会有1-2次闪断，请选择业务低峰期进行。

\*\*操作步骤\*\*

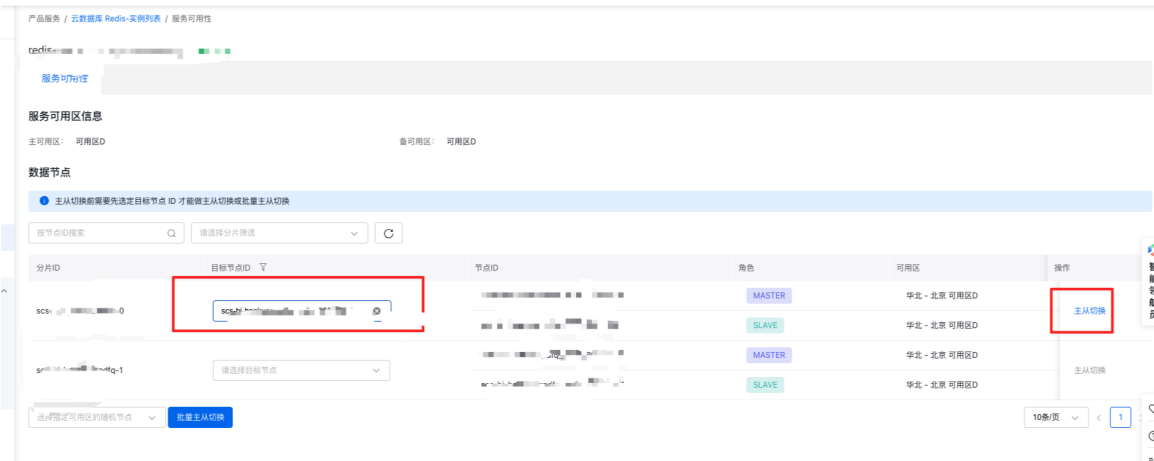
- 登录[云数据库 Redis 管理控制台](#)。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



- 在实例详情页的标签页导航栏中点击服务可用性。



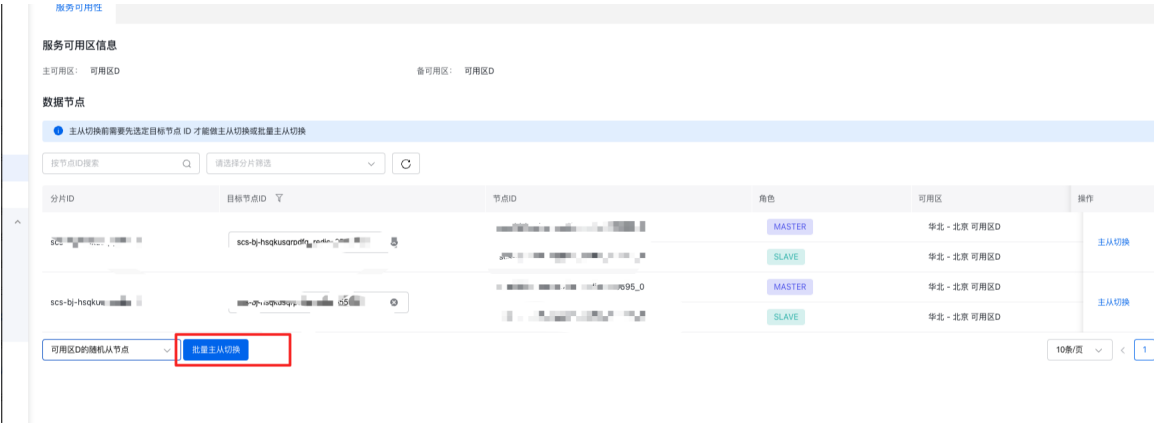
- 选取目标节点后，点击主从切换。



- 勾选我已了解风险并确认执行操作后点击确认。



6. 也可选择批量主从切换，快捷切换到可用区随机从节点。



修改实例域名

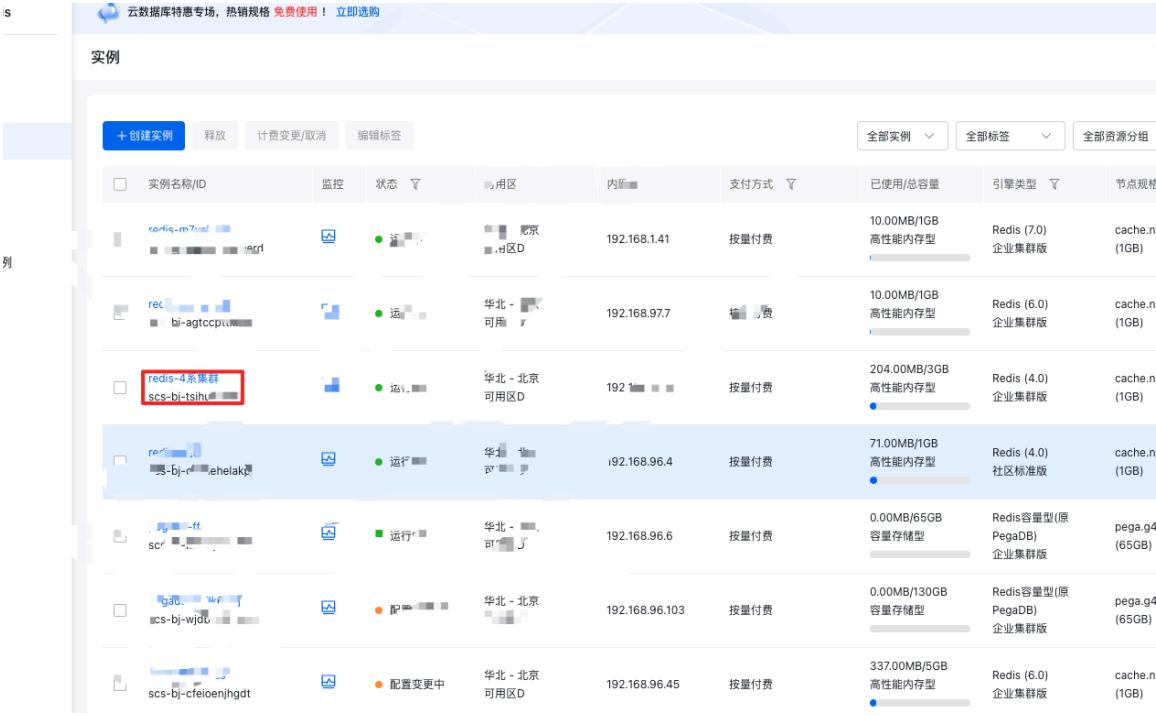
概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例域名的基本方法。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在Redis产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处**运行中**。
3. 点击“实例名称/ID”下面的目标**实例名称**，进入“实例详情”页面。





4. 点击**变更**，输入变更后域名。



5. 点击**确认**按钮，用户可以完成修改实例域名的操作。

### 相关 API

- API 接口：[变更实例](#)

### 🔗 设置维护窗口

#### 概述

本文为您介绍通过云数据库 Redis 管理控制台查看、设置维护窗口的基本方法。

#### 设置维护窗口

- 登录[云数据库 Redis 管理控制台](#)。
- 进入Redis产品服务的“云数据库 Redis-实例列表”页面。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。



4. “实例详情”页面维护信息可以查看到当前实例的维护窗口时间，点击维护信息右侧的**修改**按钮，进行维护周期和时间的修改。



5. 修改完成点击**确认**按钮，即可完成修改。



🔗 设置公网访问

概述

本文为您介绍通过云数据库 Redis 管理控制台绑定、解绑实例公网IP的基本方法。

说明：

- 若您当前暂无空闲的EIP，需要先创建一个EIP，用于绑定Redis实例。
- 若您的Redis实例已绑定公网IP，您可以调整EIP的带宽。
- 若您需要解绑公网IP，需要到“弹性公网IP”进行解绑操作。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在Redis产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处**运行中**。
3. 点击“实例名称/ID”下面的目标**实例名称**，进入“实例详情”页面。

4. 点击公网IP右侧的设置按钮，进行公网IP绑定设置。

本信息

号管理

全管理

控报警

数管理

分管理

务可用性

据管理DMS

artDBA

分析缓存

会话管理

日志明细

基本信息

实例ID: my-knvsfjwbakel

实例名称: my-knvsfjwbakel

实例组ID: my-knvsfjwbakel

创建时间: 2023-07-10 16:45

区域可用区: 华北 - 北京 可用区D

付费方式: 按量付费

网络信息

内网连接域名: redis.redis-cn-beijing-1.baidubce.com

公网连接域名: --

连接密码: --

内网端口: 6379

公网IP: 未分配

内网私密访问: 已关闭

内网IP: 192.168.96.65

公网端口: --

IPID: 192.168.96.65

维护信息

维护周期: 周一、周二、周三、周四、周五、周六、周日

维护时间: 03:00

持续时间: 1小时

配置信息

引擎类型: 企业集群版 Redis 容量型(原PegaDB)

节点规格: pegas4s1.micro(65GB)

分片数量: 1

存储类型: 容量存储型

总容量: 65GB

副本数量: 3

架构类型: 集群版

数据引擎: PegaDB 65GB

5. 选择您所需要的公网IP，进行绑定操作，点击确认即可。

修改实例端口

概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例端口的的基本方法。

说明：

- 支持修改内网端口和公网端口。
- 已绑定公网IP的实例支持修改公网端口。

操作步骤

- 登录云数据库 Redis 管理控制台。
- 在Redis产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处运行中。
- 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。

实例名称/ID	监控	状态	可用区	内网IP	支付方式	已使用/总容量	引擎类型	节点规格	副本数量	标签
my-knvsfjwbakel		运行中	华北 - 北京 可用区D	192.168.96.65	按量付费	10.00MB/1GB 高性能内存型	Redis (7.0) 企业集群版	cache.n1.micro (1GB)	2	-
my-knvsfjwbakel		运行中	华北 - 北京 可用区D	192.168.96.77	按量付费	10.00MB/1GB 高性能内存型	Redis (6.0) 企业集群版	cache.n1.micro (1GB)	2	-
my-knvsfjwbakel		运行中	华北 - 北京 可用区D	192.168.96.81	按量付费	10.00MB/1GB 高性能内存型	Redis (4.0) 企业集群版	cache.n1.micro (1GB)	2	-
scs-bj-cjunehehe		运行中	华北 - 北京 可用区D	192.168.96.81	按量付费	71.00MB/1GB 高性能内存型	Redis (4.0) 社区标准版	cache.n1.micro (1GB)	2	-
scs-bj-knvsfjwbakel		运行中	华北 - 北京 可用区D	192.168.96.65	按量付费	0.00MB/65GB 容量存储型	Redis容量型(原PegaDB) 企业集群版	pega.g4s1.micro (65GB)	3	-
pega-g4s1-micro		配置变更中	华北 - 北京 可用区D	192.168.96.65	按量付费	0.00MB/130GB 容量存储型	Redis容量型(原PegaDB) 企业集群版	pega.g4s1.micro (65GB)	2	-
pega-g4s1-micro		配置变更中	华北 - 北京 可用区D	192.168.96.45	按量付费	337.00MB/5GB 高性能内存型	Redis (6.0) 企业集群版	cache.n1.micro (1GB)	2	-
redis-fj		配置变更中	华北 - 北京 可用区D	192.168.96.42	按量付费	337.00MB/5GB 高性能内存型	Redis (6.0) 企业集群版	cache.n1.micro (1GB)	2	-

4. 点击端口旁边的变更操作，弹出输入框，输入变更后端口号。

224



5. 输入后，点击**确定**按钮，用户可以完成修改实例域名的操作。

变更配置

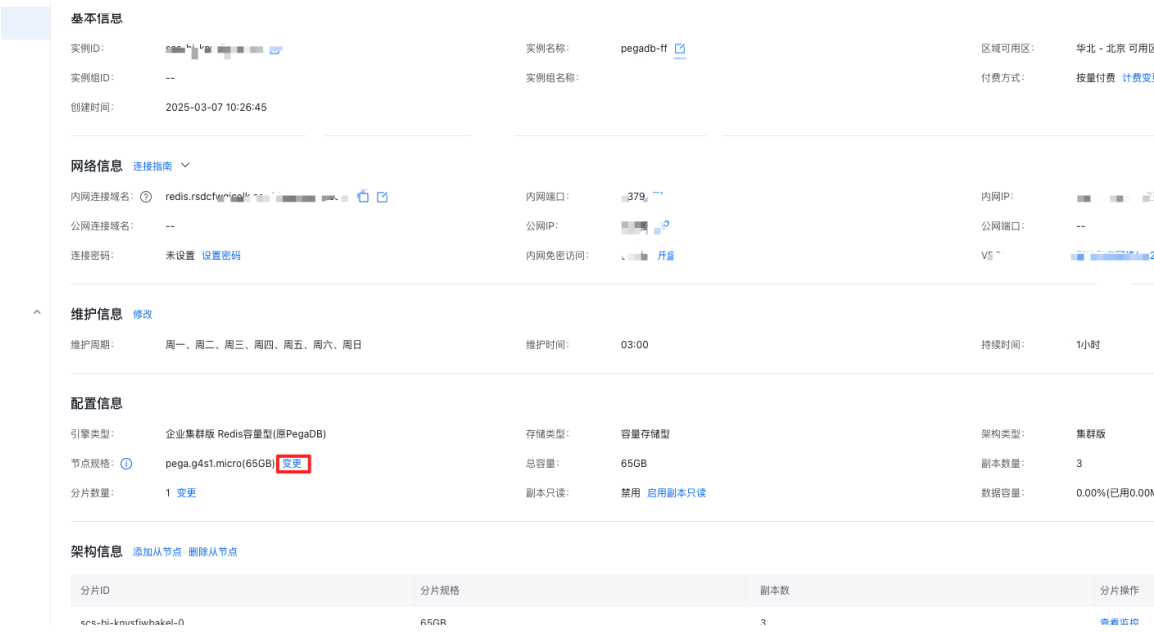
**概述** 本文为您介绍通过云数据库 Redis 管理控制台变更配置的基本方法。

注意事项

- 变更配置后历史备份将失效，需要重建备份。
- 集群版实例支持变更实例规格及分片数量。
- 变更配置后保证容量使用率在安全阈值，分片数不能少于 1 片。

操作步骤 变更节点规格

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在“云数据库 Redis-实例列表”中，选择需要变更配置的实例，点击实例名称，进入实例详情。
3. 在实例详情页面的【配置信息】模块，点击节点规格后的变更，进入“变更配置”页面。

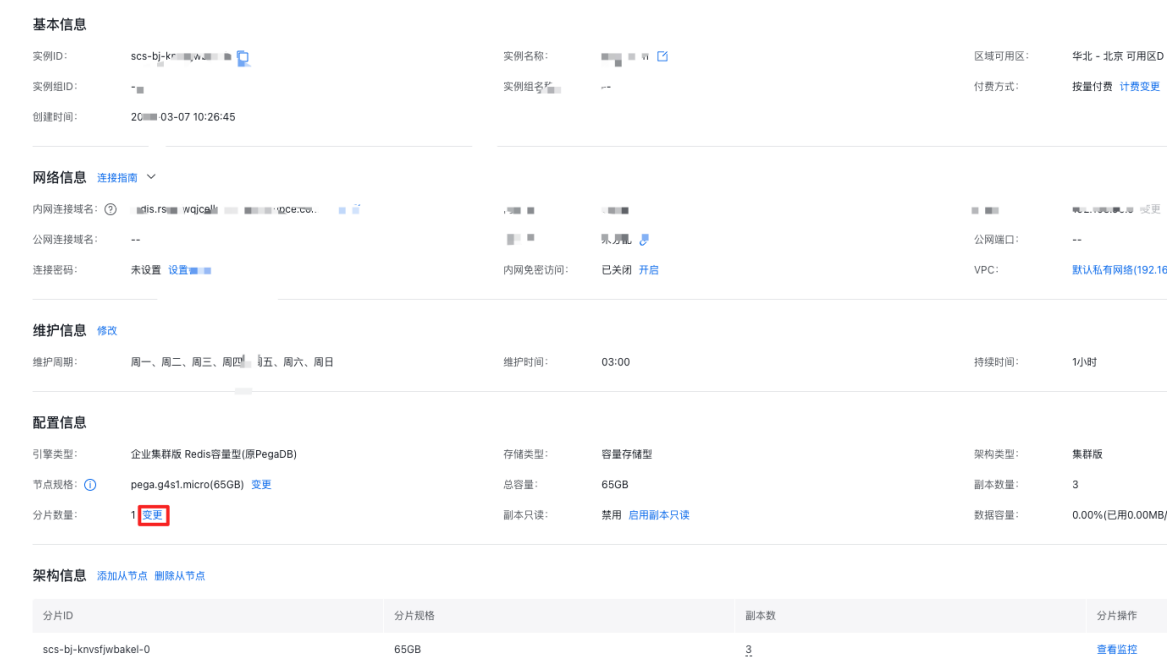


4.实例规格确认完成，然后在右侧订单栏点击下一步按钮，进行订单付费。完成订单付费后，实例配置将进行变更。待实例变配完成，实例状态更新为“运行中”，您可在实例详情查看实例的最新配置。

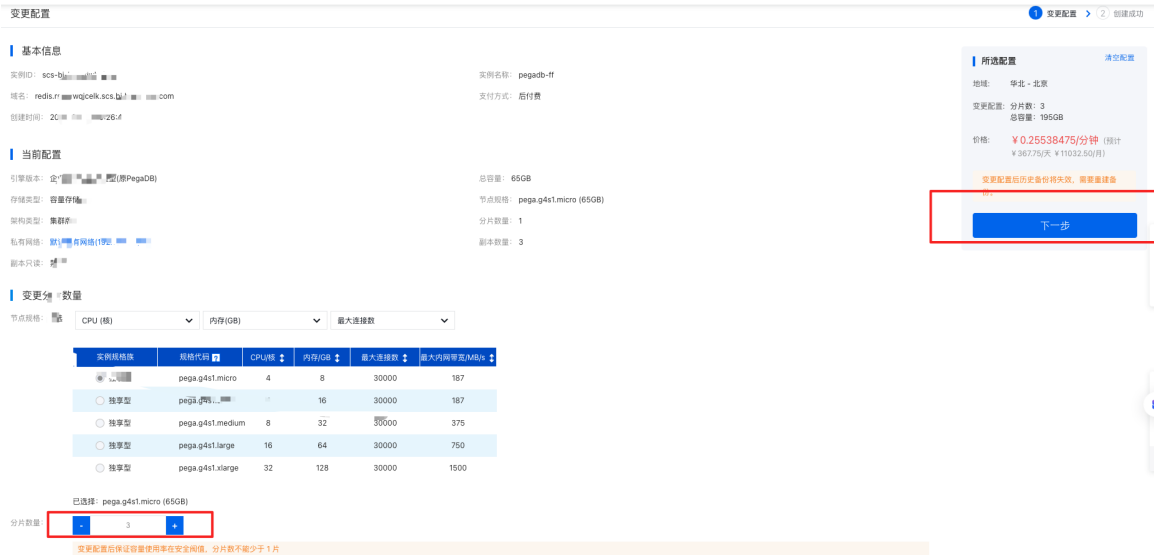


变更分片数量

- 1. 登录[云数据库 Redis 管理控制台](#)。
- 2. 在“云数据库 Redis-实例列表”中，选择需要变更配置的实例，点击实例名称，进入实例详情。
- 3. 在实例详情页面的【配置信息】模块，点击分片数量后的变更，进入“变更配置”页面。



- 4.分片数确认完成，然后在右侧订单栏点击下一步按钮，进行订单付费。完成订单付费后，实例配置将进行变更。待实例变配完成，实例状态更新为“运行中”，您可在实例详情查看实例的最新配置。



相关API

- API 接口：[变更实例](#)

修改实例名称

概述

本文为您介绍通过云数据库 Redis 管理控制台修改实例名称的基本方法。

操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 在Redis产品服务的“云数据库 Redis-实例列表”页面，确保该实例的状态处运行中。
3. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
4. 点击基本信息模块**变更**按钮，弹出文本框，输入变更后名称。



5. 点击文本框旁边的**确认**按钮，用户可以完成修改实例名称的操作。

相关 API

- API 接口：[修改实例名称](#)

SmartDBA

大Key分析

概述

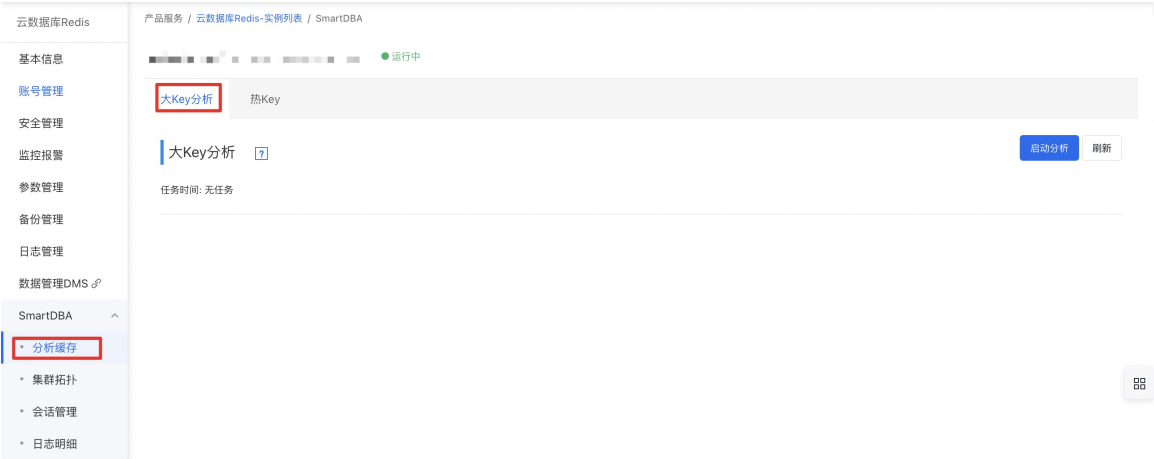
云数据库 Redis 提供分析缓存占用和大 Key 分布情况的功能。本文为您介绍通过管理控制台进行缓存分析的通用方法。

注意事项

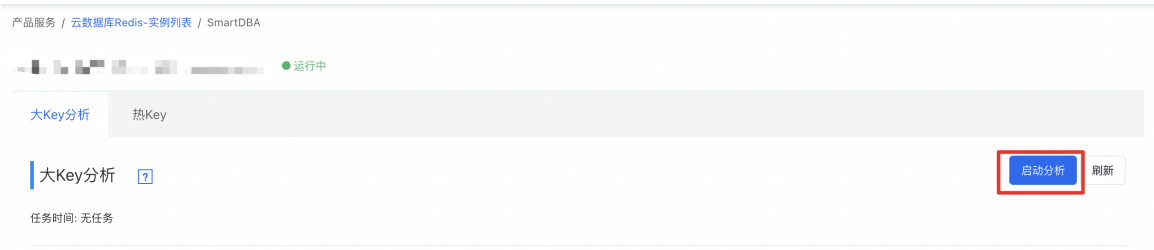
- 启动分析任务后，实例将处于缓存分析中的状态，期间控制台各项操作将被禁用。
- 分析任务完成后，将展示占用空间最大的前 100 个 Key。
- 不支持单副本的实例。

\*\*操作指南\*\*

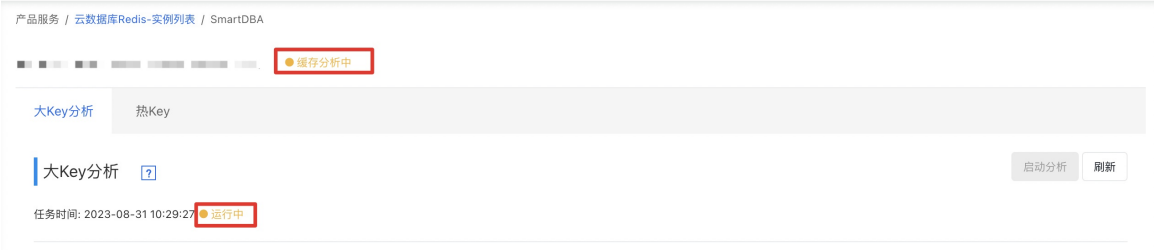
1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标**实例名称/ID**，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **SmartDBA**，默认选中**大Key**页签。



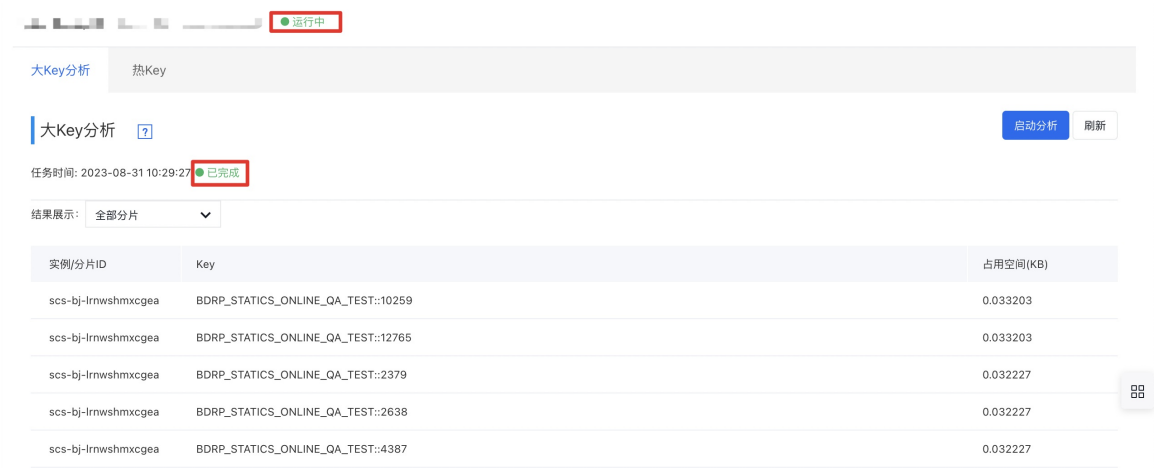
3. 在 **大Key** 页面中，点击“大Key”模块的**启动分析**按钮。



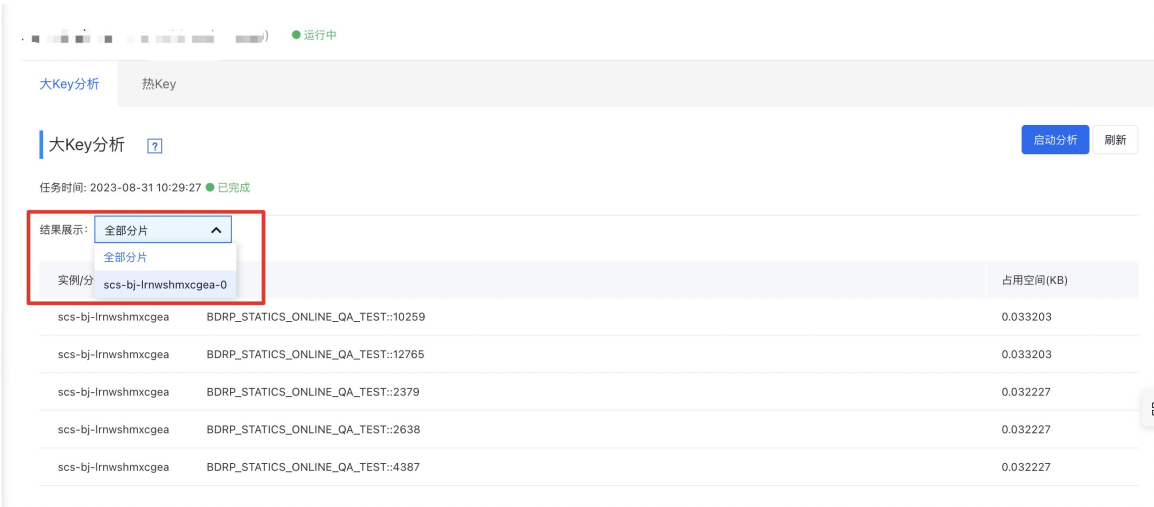
4. 分析任务状态变为**运行中**且实例状态显示为**缓存分析中**。



5. 当任务运行完毕，点击**刷新**按钮，分析任务状态变为**已完成**，分析结果将展示在下方。您可以在该页面再次启动新的分析任务，新任务将覆盖旧任务的分析结果。



5. 如果实例为集群版，可以点击结果展示处的下拉菜单，选择需要查询全部分片或某个具体分片。



热Key分析

概述

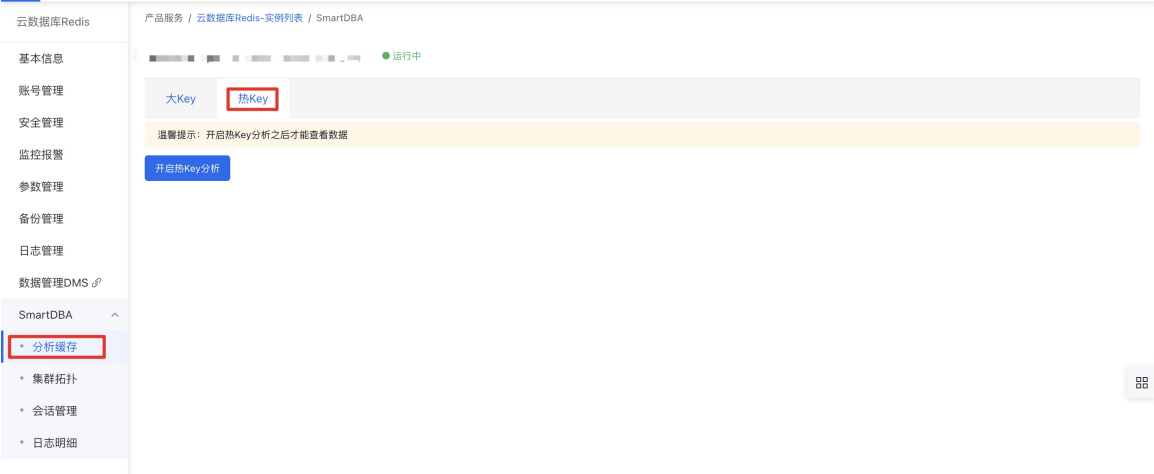
云数据库 Redis 提供热 Key 分布情况的功能。本文为您介绍通过管理控制台进行缓存分析的通用方法。

注意事项

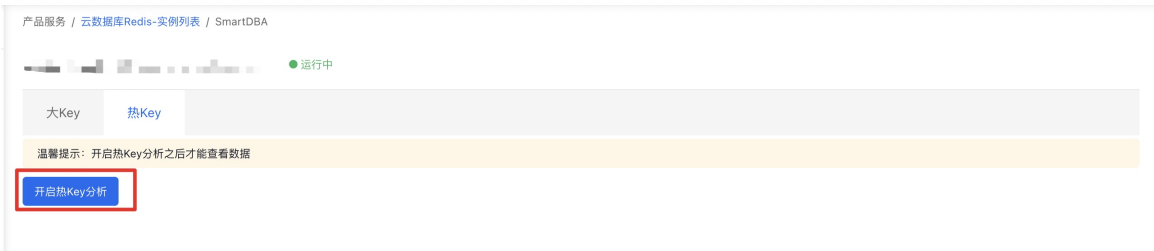
- 实时数据取最近60s的数据，当打开自动刷新后取最近30s的数据。
- 不支持单副本的实例。

开启热Key分析缓存

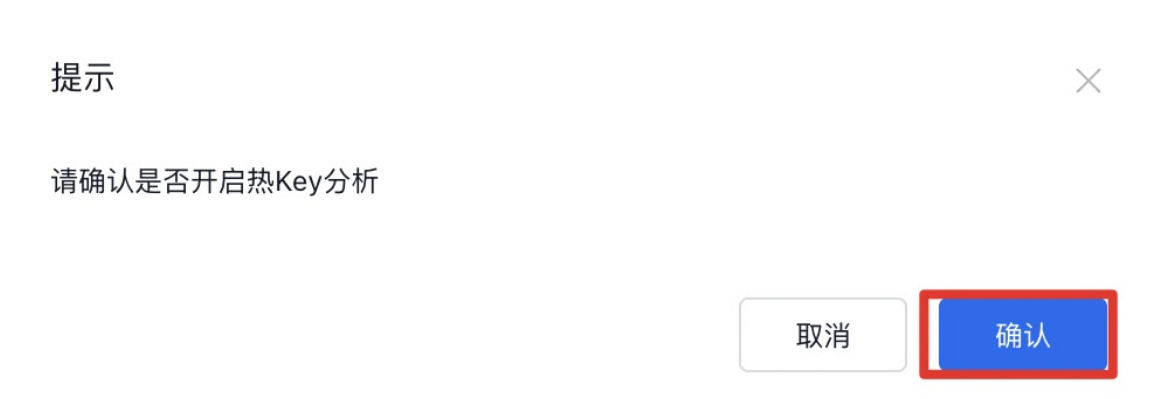
1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
2. 点击目标实例名称/ID，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 SmartDBA，默认选中大Key页签，点击热Key页签，切换热key页面。



4. 在 **热Key** 页面中，点击**开启热Key分析**按钮。



5.弹出提示窗口，点击**确定**按钮。



6. 打开**实时数据**和**历史数据**页签。



### 实时数据

以下操作前提是**开启热Key分析缓存**的操作步骤：

1. 点击**热Key**页签中的**实时数据**子页签，切换到**实时数据**列表，可以看到相关提示。





- 2. 在节点ID选择列表中可以指定实例或全部(全部实例)。
- 3. 在这一步，如果是自动刷新是已打开状态，则可以忽略此步骤，否则则需要开启它。



- 4. 打开后，系统30秒将更新次列表数据，需要用户手动点击刷新按钮。



历史数据

以下操作前提是开启热Key分析缓存的操作步骤：

- 1. 点击热Key页签中的实时数据子页签，切换到历史数据列表，可以看到相关提示。



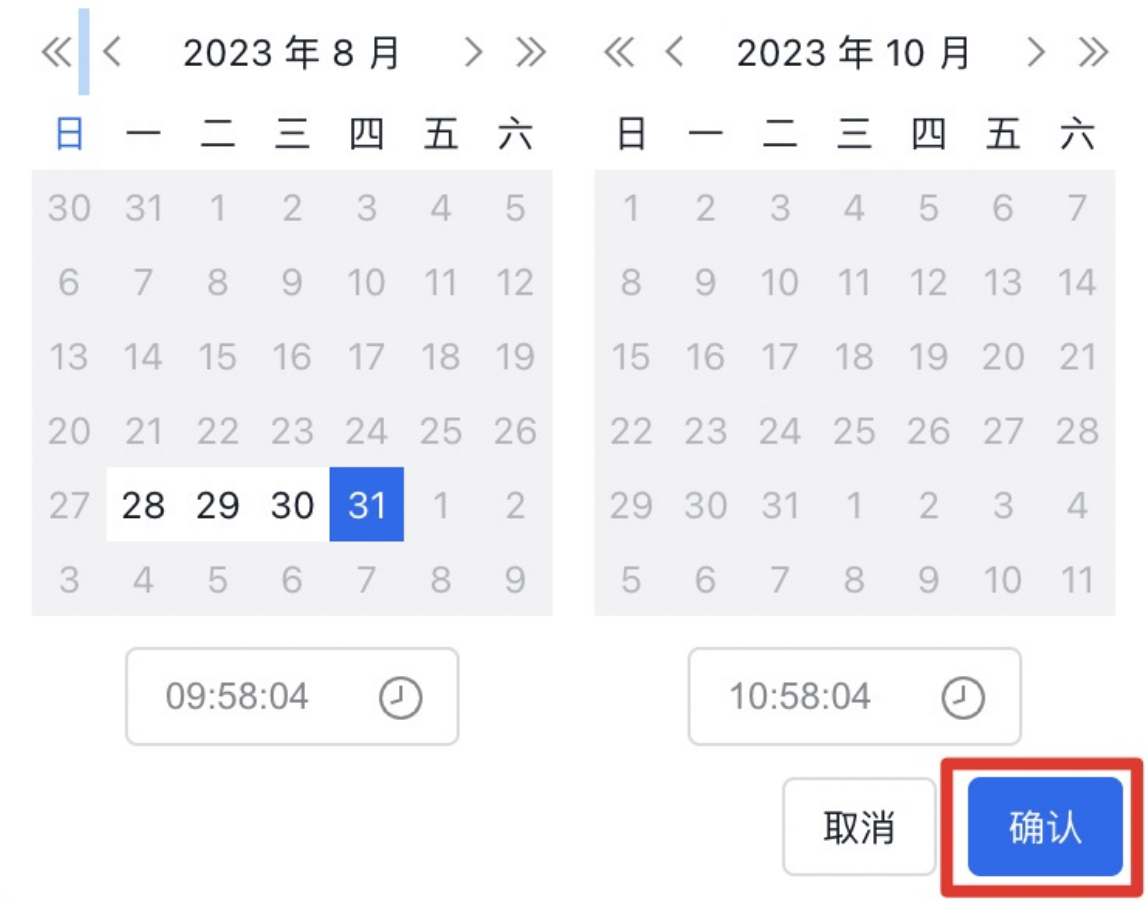
- 2. 在节点ID选择列表中可以指定实例或全部(全部实例)。



3. 在时间范围选择列表中可以点击框内日历图标。



4. 弹出日历组件，选择开始和结束范围，点击确定按钮。



5. 点击列表上的刷新按钮，列表将刷新数据。



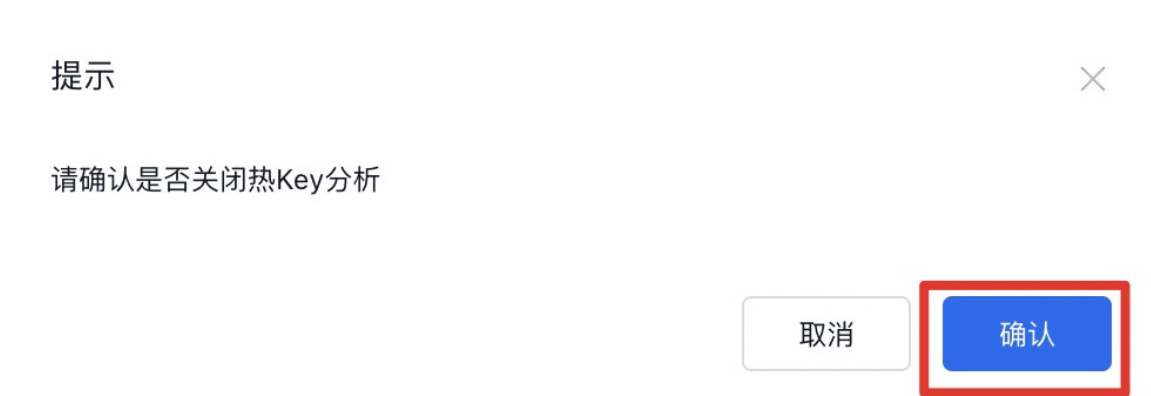
关闭热Key分析缓存

以下操作前提是开启热Key分析缓存的操作步骤：

- 1. 点击实时数据和历史数据页签最右侧关闭热Key分析按钮。



- 2. 弹出提示窗口，点击确定按钮。



- 3. 自动刷新后，会出现开启热Key分析按钮，表示热Key分析缓存已经关闭。



会话管理

概述

会话管理用于查看当前实例的会话状态、会话详情、会话统计等。

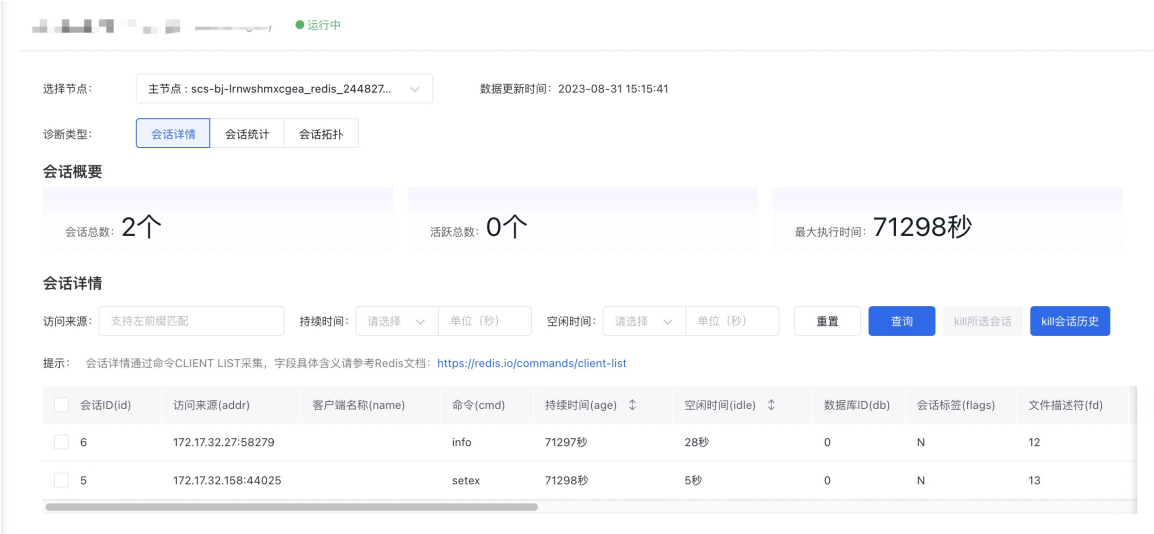
操作步骤

- 1. 登录云数据库 Redis管理控制台，进入实例列表页面。

2. 点击目标实例名称/ID，进入实例详情页面。
3. 在实例详情页的左侧导航栏中，点击 **SmartDBA**菜单下的**会话管理**菜单，打开会话管理页面，节点选择默认选择**集群节点信息**项，可查看集群会话概要。



- 4.在**选择节点**选择具体节点实例，，将出现**诊断类型**具体包含会话详情、会话统计、会话拓扑。例如：  
lrnwshmcgea\_redis\_244827\_0。



会话详情

点击**诊断类型**的**会话详情**按钮，出现会话详情页面，页面分为会话概要、会话详情两部分。

会话概要

会话概要包含当前会话会话总数、活跃总数、最大执行时间三个部分的统计，具体如下。



会话详情

会话详情包含会话列表，可以对会话筛选、会话进行kill、查看会话框历史等操作，具体如下。

会话详情

访问来源:

持续时间: 

请选择

单位 (秒)

空闲时间: 

请选择

单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示: 会话详情通过命令CLIENT LIST采集, 字段具体含义请参考Redis文档: <https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↕	空闲时间(idle) ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

• 会话筛选

在列表上放输入访问来源、持续时间、空闲时间等条件, 如果不加入筛选条件, 则查询所有, 完成并确认筛选条件后, 点击**查询**按钮。

访问来源:

持续时间: 

请选择

单位 (秒)

空闲时间: 

请选择

单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示: 会话详情通过命令CLIENT LIST采集, 字段具体含义请参考Redis文档: <https://redis.io/commands/client-list>

<input type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↕	空闲时间(idle) ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

- 访问来源是IP地址。
- 持续时间和空闲时间选择条件是大于 (>) 、小于 (<) 。
- 持续时间单位是输入条件是秒数。

• kill 所选会话

1. 在会话列表选择所需要kill的会话记录, 勾选当前行复选框, 点击会话列表右上角上 **kill所选会话**按钮。

访问来源:

持续时间: 

请选择

单位 (秒)

空闲时间: 

请选择

单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示: 会话详情通过命令CLIENT LIST采集, 字段具体含义请参考Redis文档: <https://redis.io/commands/client-list>

<input checked="" type="checkbox"/>	会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age) ↕	空闲时间(idle) ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input checked="" type="checkbox"/>	5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/>	6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

2. 弹出对话**kill**会话窗口, 确认当前会话后, 点击**确定**按钮。

kill会话

会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)	空闲时间(idle)	数据库ID(db)	会话标签(flags)	文件描述符(flags)	文件描述事件(events)
5	172.17.32.158:44025		setex	73327	2	0	N	N	r

取消

确定

• 会话历史

1. 点击会话列表右上角上 **会话历史**按钮。

访问来源: 

支持左前缀匹配

持续时间: 

请选择

单位 (秒)

空闲时间: 

请选择

单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示: 会话详情通过命令CLIENT LIST采集, 字段具体含义请参考Redis文档: <https://redis.io/commands/client-list>

<input type="checkbox"/> 会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)  ↕	空闲时间(idle)  ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/> 39604	172.17.32.158:30489		setex	1秒	1秒	0	N	13
<input type="checkbox"/> 6	172.17.32.27:58279		info	74947秒	48秒	0	N	12

2. 在弹出的会话历史窗口中，可以接近一小时、近24小时、近7天、近30天、自定义时间范围进行筛选。

kill会话历史记录

近一小时

近24小时

近7天

近30天

2023-08-01 16:17:09

—

2023-08-31 16:17:09

📅

会话ID(id)	访问来源(addr)	命令(cmd)	持续时间(age)	任务执行状态	任务执行结果
5	172.17.32.158:44025	setex	74939	SUCCESS	

关闭

• 会话重置。

1. 在列表右上角，点击充值按钮，可重置所有条件进行查询。

会话详情

访问来源: 

支持左前缀匹配

持续时间: 

请选择

单位 (秒)

空闲时间: 

请选择

单位 (秒)

重置

查询

kill所选会话

kill会话历史

提示: 会话详情通过命令CLIENT LIST采集, 字段具体含义请参考Redis文档: <https://redis.io/commands/client-list>

<input type="checkbox"/> 会话ID(id)	访问来源(addr)	客户端名称(name)	命令(cmd)	持续时间(age)  ↕	空闲时间(idle)  ↕	数据库ID(db)	会话标签(flags)	文件描述符(fd)
<input type="checkbox"/> 5	172.17.32.158:44025		setex	73327秒	2秒	0	N	13
<input type="checkbox"/> 6	172.17.32.27:58279		info	73326秒	121秒	0	N	12

会话统计

点击诊断类型的会话统计按钮，出现回话统计页面，分别用户维度统计、访问来源维度统计、数据库维度统计三部分统计列表及图表数据。

• 用户维度统计

字段包含用户、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时长（秒）。

产品服务 / 云数据库 Redis-实例列表 / SmartDBA-会话管理

主节点: scs-b...redis\_244827...

数据更新时间: 2023-08-31 16:39:57

会话详情

会话统计

会话拓扑

用户维度统计

用户	活跃会话数	总会话数	活跃会话平均时长 (秒)	活跃会话总时长 (秒)
	0	2	0	0

用户活跃会话数占比

• 访问来源维度统计

字段包含来源IP、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时（秒）。



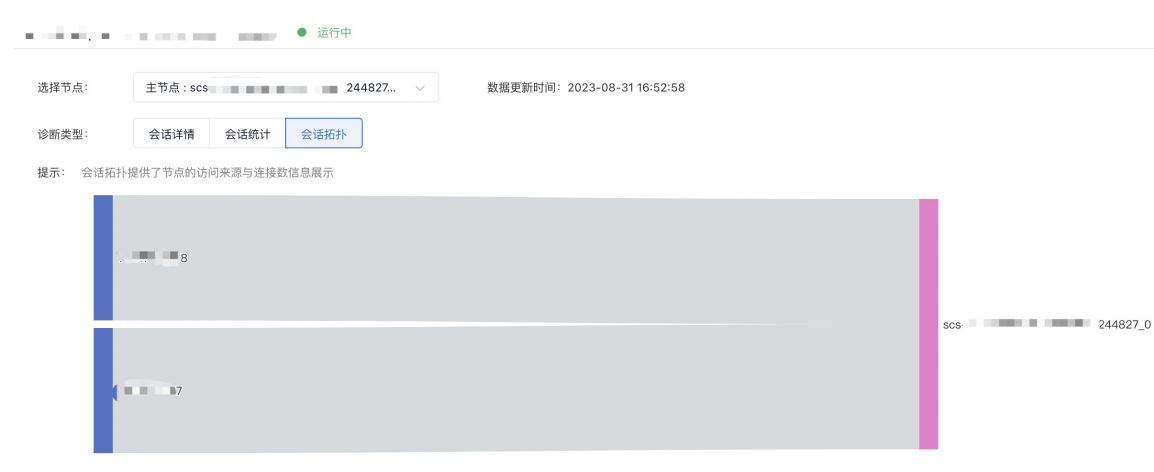
访问来源维度统计

字段包含数据库、活跃会话数、总会话数、活跃会话平均时长（秒）、活跃会话总时（秒）。



会话拓扑

点击诊断类型的会话拓扑按钮，打开会话拓扑图页面，用户可以通过切换节点ID方式来查看相应实例的拓扑图。



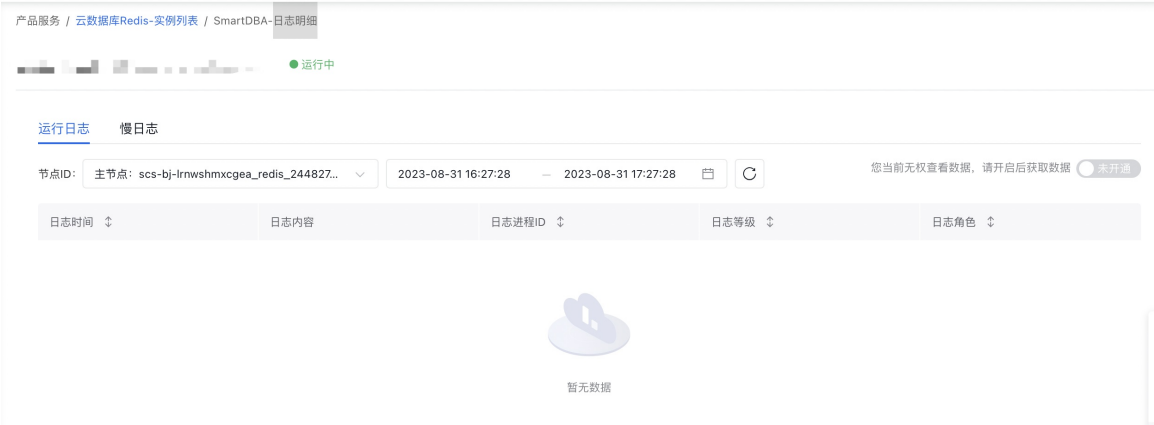
日志明细

概述

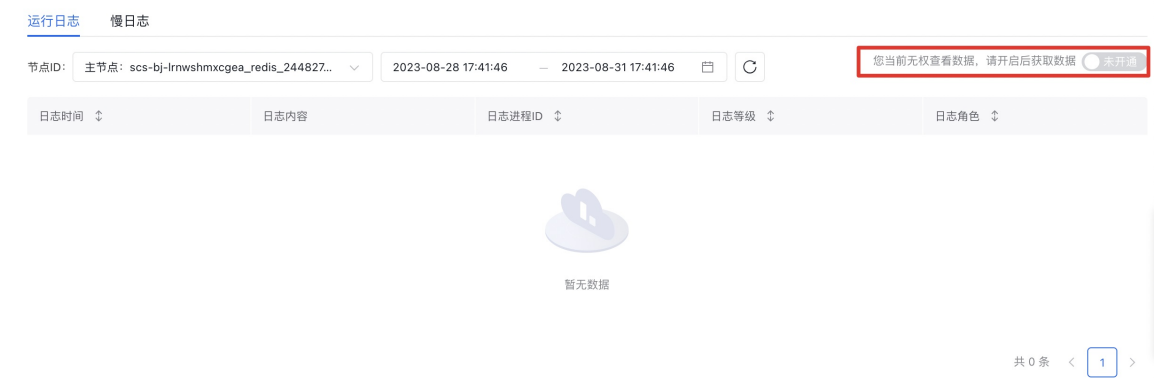
查看当前redis实例运行日志和慢日志列表数据。

运行日志-操作指南

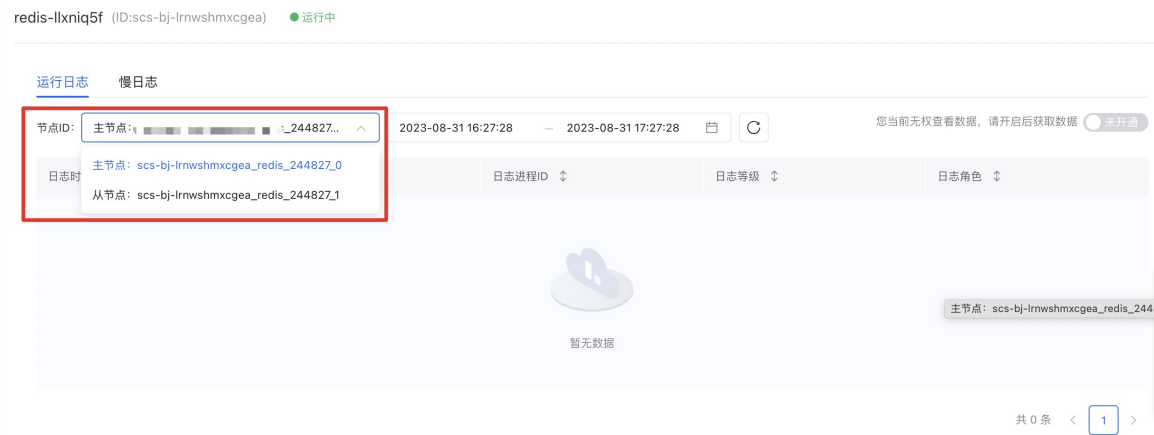
- 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 点击目标实例名称/ID，进入实例详情页面。
- 在实例详情页的左侧导航栏中，点击 SmartDBA，菜单展开的明细日志菜单，进入明细日志页面，在页面中默认选中运行日志页签。



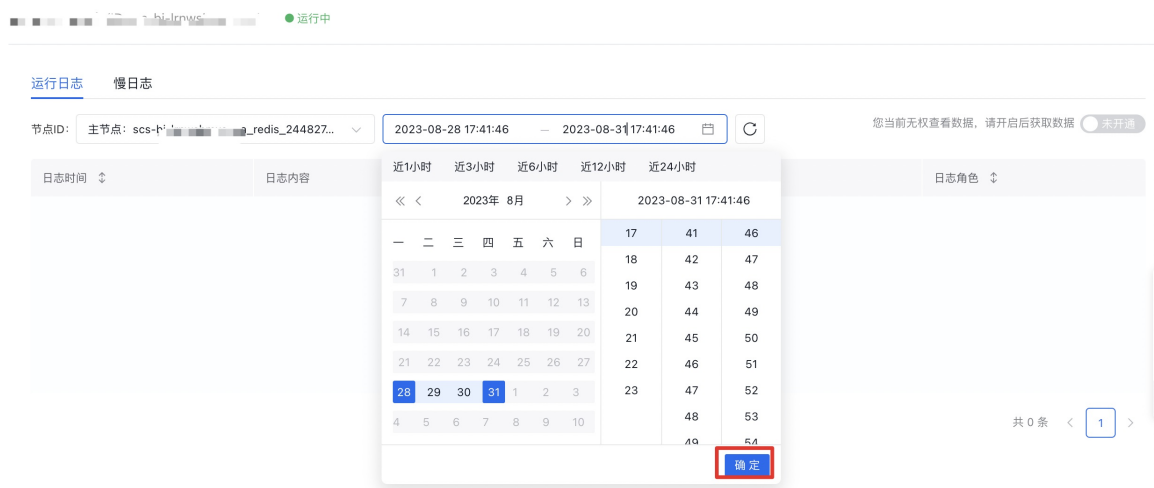
4. 如果你有开通运行日志的权限，点击右侧开通查看数据权限，没有则忽略下面操作步骤。



5. 在节点ID下拉列表框中，选择实例。

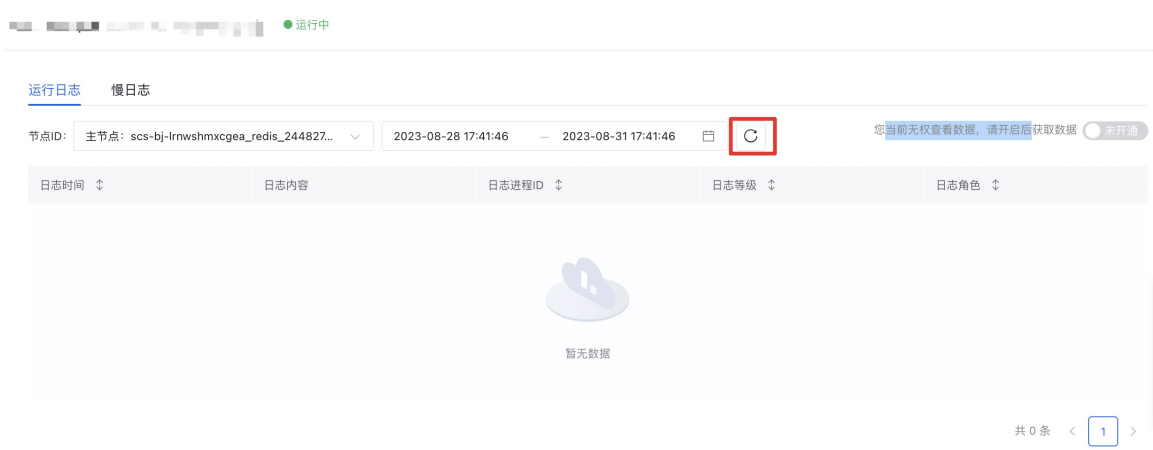


6. 在日历框中范围点击右侧日历图标，弹出日历组件,选择开始时间和结束时间，完成后，点击确定按钮，即可刷新列表数据。



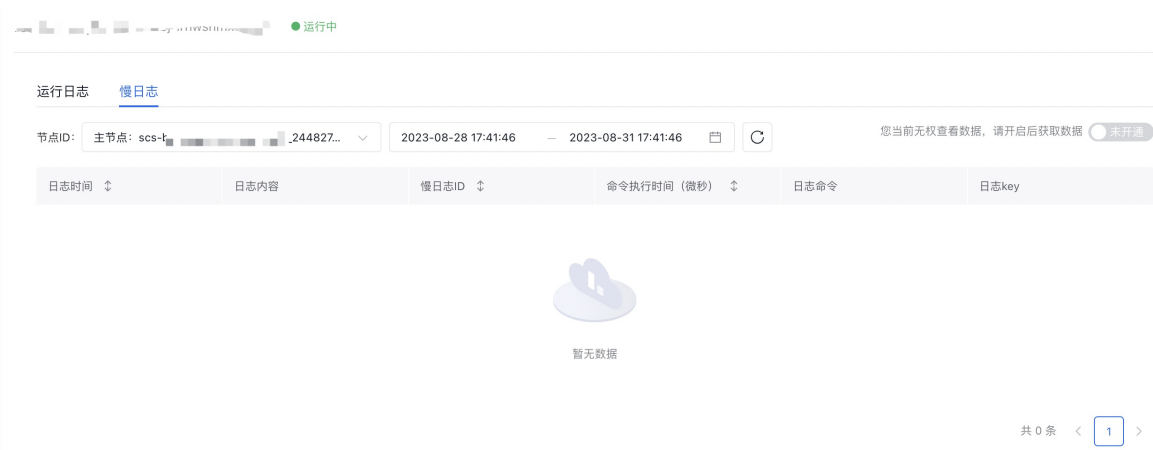


7. 若想刷新列表，则点击日历组件右侧刷新按钮，即可刷新列表。若不想刷新列表，此操作忽略。

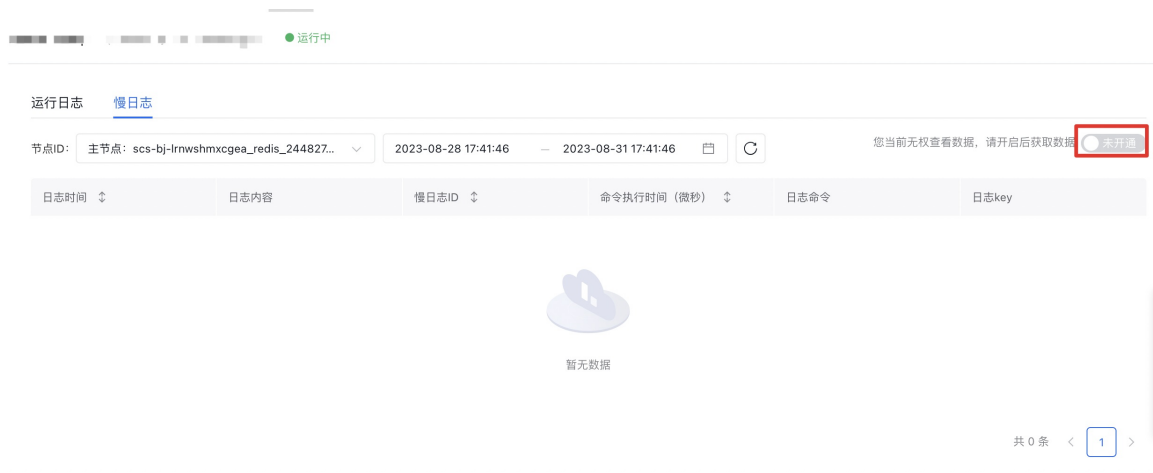


慢日志-操作指南

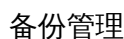
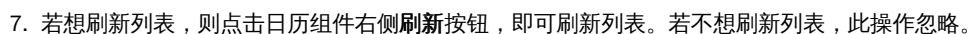
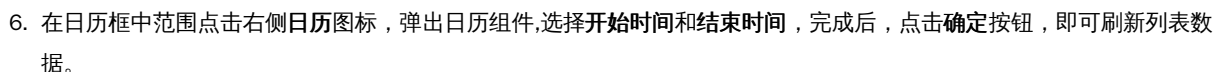
- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称/ID，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 **SmartDBA**，菜单展开的**明细日志**菜单，进入明细日志页面，在页面中默认选中**慢日志**页签。



4. 如果你有开通慢行日志的权限，点击右侧开通查看数据权限，没有则忽略下面操作步骤。



5. 在节点ID下拉列表框中，选择实例。



## 概述

1. 登录[云数据库 Redis](#) 管理控制台，进入实例列表页面。
2. 点击目标实例名称，进入实例详情页面。

- 3. 在实例详情页的左侧导航栏中，点击**备份管理**。
- 4. 在备份管理页面，点击**手动备份**按钮。



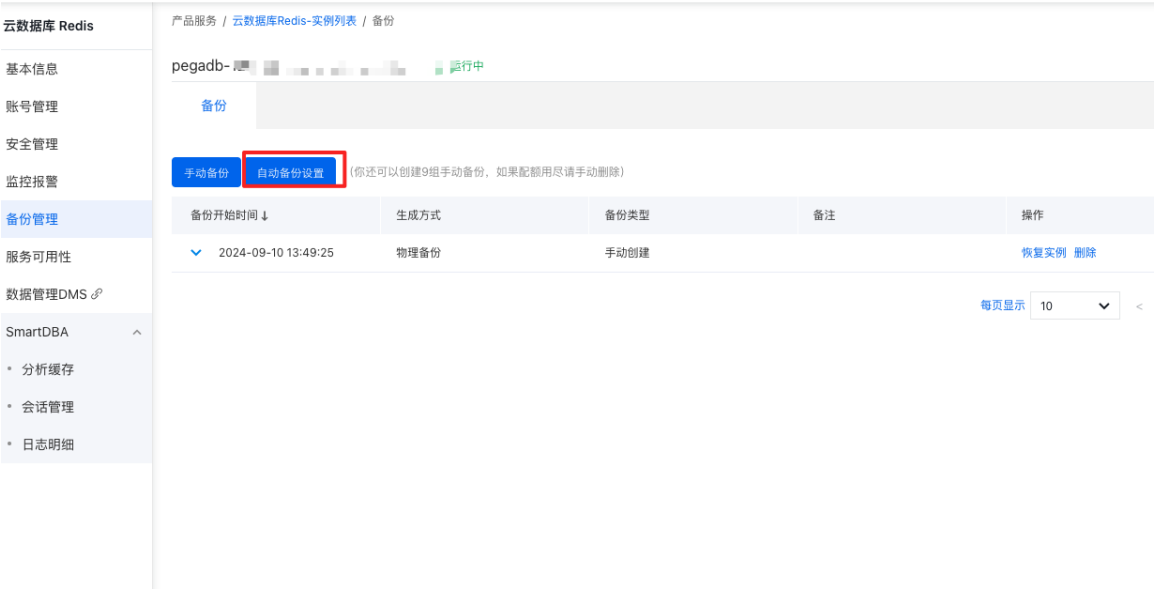
- 5. 弹出备份对话框，输入备注信息，点击**确定**按钮。



- 6.开始备份的时候，实例状态显示为**备份中**，备份完成后实例状态显示为**运行中**。

自动备份

- 1. 登录**云数据库 Redis** 管理控制台，进入实例列表页面。
- 2. 点击目标**实例名称**，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 **备份管理**。
- 4. 在备份管理页面，点击**自动备份**按钮。



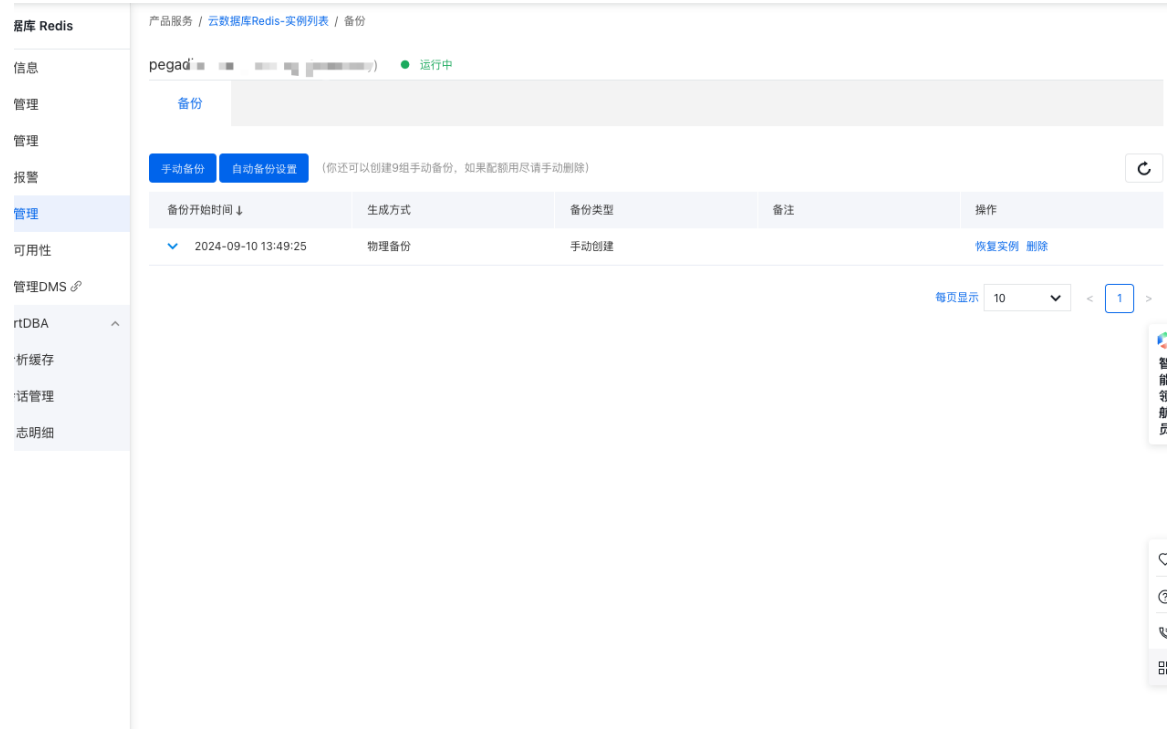
5. 弹出备份对话框，选择自动时间（执行时间），重复日期、及备份保留时长，选择完成后，点击**确定**按钮。



6.开始备份的时候，实例状态显示为**备份中**，备份完成后实例状态显示为**运行中**。

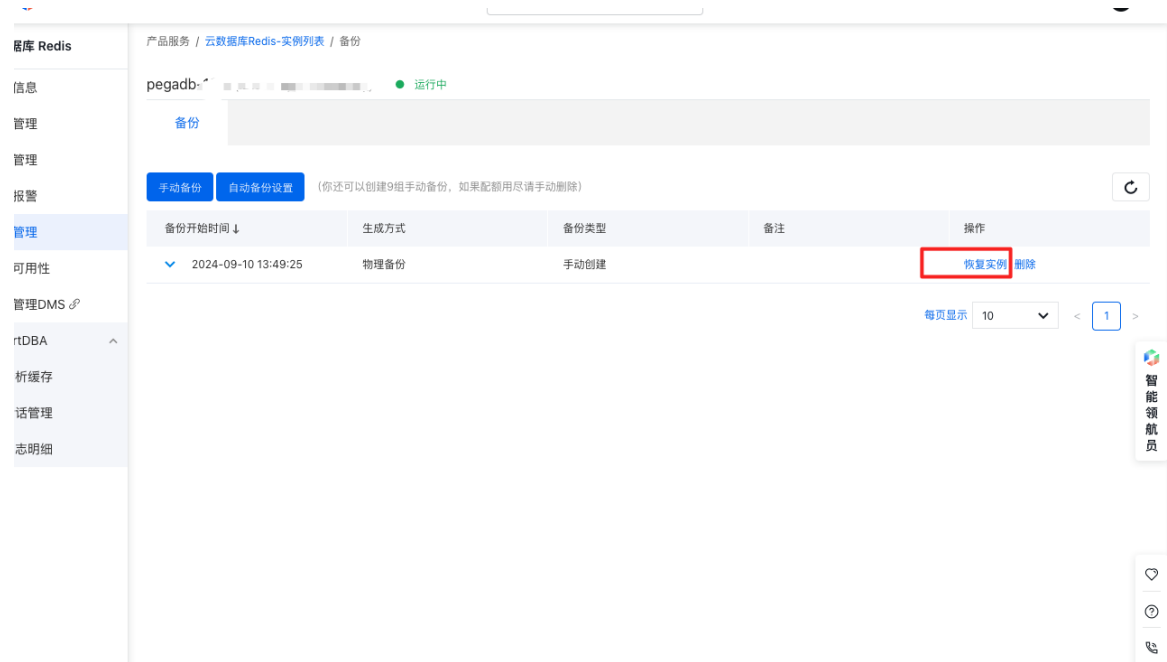
备份列表

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标**实例名称**，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击**备份管理**，进入到备份列表页面。

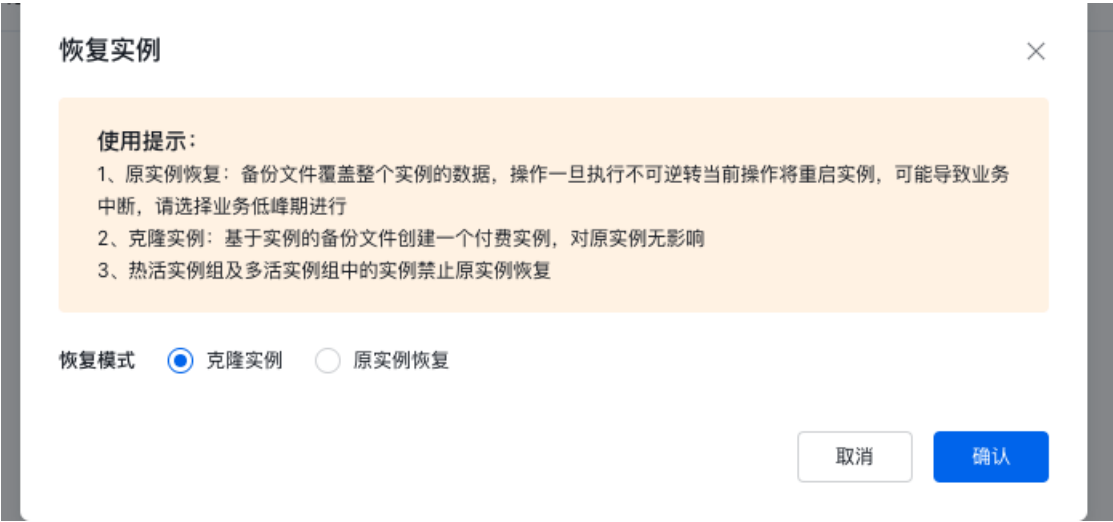


恢复实例

- 1. 登录云数据库 Redis 管理控制台，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击备份管理，进入到备份列表页面。 4.在备份列表页面，点击恢复实例按钮。



- 5. 在弹出恢复实例的对话框，选择恢复模式（克隆实例或原有实例），选择完成后，点击确定按钮。



说明：

- 1. 原实例恢复：备份文件覆盖整个实例的数据，操作一旦执行不可逆转当前操作将重启实例，可能导致业务中断，请选择业务低峰期进行。
- 2. 克隆实例：基于实例的备份文件创建一个付费实例，对原实例无影响。
- 3. 热活实例组及多活实例组中的实例禁止原实例恢复。
- 4. 备份文件失效问题：分片数变更后历史备份文件失效。

参数管理

参数配置

**概述** 您可以根据业务场景对PegaDB实例的参数进行自定义配置，以提升实例的性能与安全性。本文介绍各参数的详细说明以及操作步骤。

注意事项

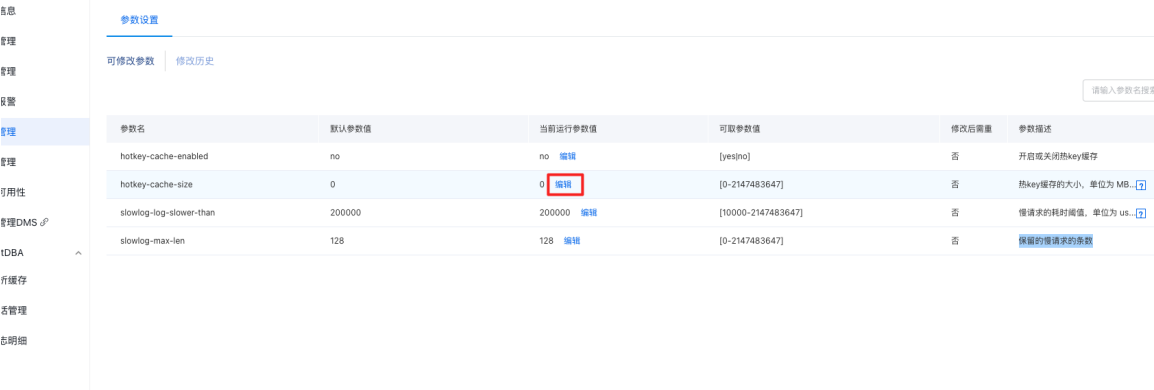
- 部分参数在提交修改后会自动重启实例。在设置参数时，您需要关注目标参数的**修改后需重启**列，更多信息请参见**参数以及说明**。

参数及说明

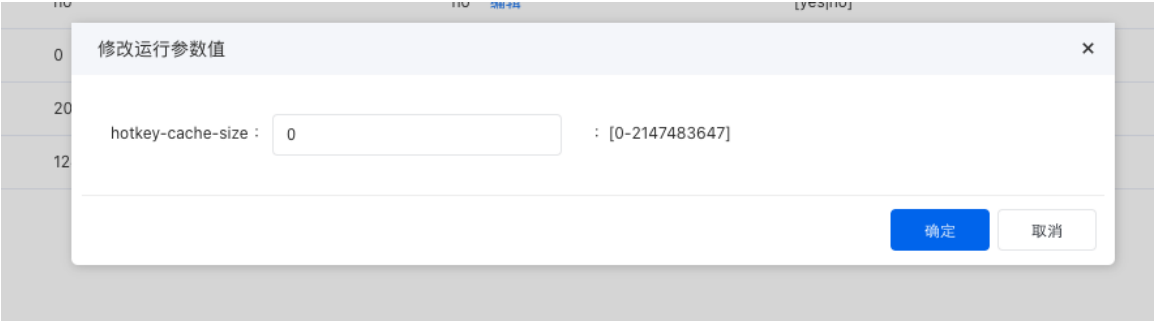
参数名	默认参数值	可取参数值	修改后需重启	参数描述
hotkey-cache-enabled	no	[yes、no]	否	开启或关闭热key缓存
hotkey-cache-size	0	[0-2147483647]	否	热key缓存的大小，单位为 MB
slowlog-log-slower-than	200000	[-1-10000000]	否	慢请求的耗时阈值，单位为 us
slowlog-max-len	128	[0-2147483647]	否	保留的慢请求的条数

操作步骤

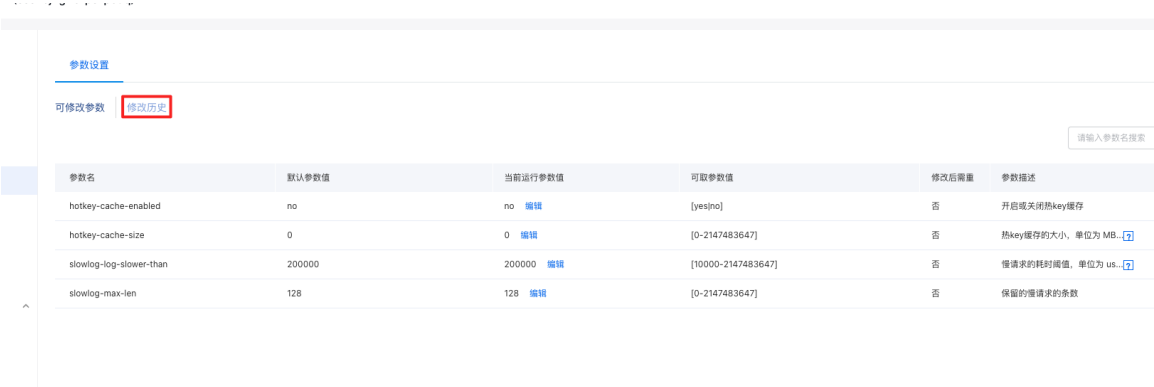
- 1. 登陆[云数据库 Redis](#)，进入实例列表页面。
- 2. 点击目标实例名称，进入实例详情页面。
- 3. 在实例详情页的左侧导航栏中，点击 **参数管理**。
- 4. 在参数管理页面中，选择可修改参数页签，点击目标参数“当前运行参数值”列的**编辑**。



5.在弹出窗口中修改参数值，单击**确定**按钮。



6.点击**修改历史**，可查看参数修改历史记录。



## 版本升级

### 🔗 小版本升级（PegaDB）

#### 概述

为提升用户体验，云数据库Redis容量型（PegaDB）会不定期地发布小版本，用于丰富云产品功能或修复已知缺陷。您可以在控制台进行升级小版本的操作。也可以通过设置维护时间实现在业务低峰期进行升级任务。 **注意事项**

- 该操作将升级实例到最新的内核版本，系统会自动检测实例的小版本，如果小版本升级按钮处于无法单击的状态，表示该实例已经是最新的小版本。
- 由于各地域版本发布进度可能有所差异，小版本发布情况以当前控制台显示为准。
- 只有运行中的实例才能进行升级小版本。
- 升级过程中实例将重启，可能会有分片数\*5秒内的闪断，请尽量选择业务低峰期进行，并确保应用具有自动重连机制。
- 对于超过16分片的实例，由于分片数较多，重启使用串型化处理，时间较长，请耐心等待。
- 单副本实例不支持升级小版本。

#### 操作步骤

1. 登录[云数据库 Redis 管理控制台](#)。
2. 点击“实例名称/ID”下面的目标实例名称，进入“实例详情”页面。
3. 点击后，跳转到实例创建的基本信息界面，在配置信息的版本后点击[升级小版本](#)。

基本信息

实例ID: scs-\*\*\*\*-\*\*\*\*-\*\*\*\*-\*\*\*\*

实例名称: pds-\*\*\*\*-\*\*\*\*-\*\*\*\*-\*\*\*\*

区域可用区: 华北 - 北京 可用区D [迁移](#)

实例组ID: --

实例组名称: --

付费方式: [按量付费](#) [计费变更](#)

创建时间: 2025-04-\*\*\*\* 15:26:18

网络信息 [连接指南](#)

内网连接域名: [dis.yun\\*\\*\\*\\*.com](#) [内网端口: \\*\\*\\*\\*](#)

公网连接域名: --

公网IP: 未分配

公网端口: --

连接密码: 未设置 [设置密码](#)

内网免密访问: 已关闭 [开启](#)

VPC: [test-rds-vpc](#) [92.168.0.0/16](#)

维护信息 [修改](#)

维护周期: 周一、周二、周三、周四、周五、周六、周日

维护时间: 03:00

持续时间: 1小时

配置信息

引擎类型: 企业集群版 Redis [\(原PegaDB\)](#)

版本: 2.0(2.2.8) [小版本升级](#)

存储类型: 容量存储型

架构类型: 集群版

节点规格: [pega.g4s1.micro\(65GB\)](#) [变更](#)

总容量: 65GB

副本数量: 2

分片数量: 1 [变更](#)

副本只读: [禁用](#) [启用副本只读](#)

4.点击后，会跳出弹窗提示，弹窗信息会展示当前版本、目标版本、小版本文档链接、执行时间等信息，勾选我已了解风险并确认执行操作后，点击确认即可执行升级任务。

## 性能调优

### 如何处理Redis集群数据倾斜

#### 背景

在Redis集群中，少数分片节点的空间使用率或CPU使用率、带宽使用率、延时等性能指标明显高于其他数据分片，该Redis集群可能已产生数据倾斜。数据倾斜严重时，会导致集群在整体使用率不高的情况下，响应时间上升、写入失败等异常情况。

#### 为什么会产生数据倾斜？

数据倾斜分为空间倾斜和访问倾斜：空间倾斜可分为 key 数量倾斜和 key 大小倾向。

**key 数量倾向**比较常见的场景为 Redis 集群开启了 hashtag，业务的 key 名包含了 hashtag 大括号，在无意识下大量的 key 名中的 {} 里面包含相同的内容，这些 key 经过集群代理的转发，集中落在少数分片上，造成这些分片 key 数量明显高于其他分片。

**key 大小倾斜**比较常见的场景为业务使用了复杂数据类型，例如 LIST 类型，业务在无意识下，向少数 LIST 类型 key 添加了较多数量的元素，其元素个数明显高于其他 LIST 类型 key，就出现在各分片 key 数量相当的情况下，个别分片的空间使用率明显高于其他分片。

访问倾斜可分为CPU消耗型访问倾斜和IO消耗型访问倾斜。

**CPU消耗型访问倾斜**一般出现在内存型 Redis 集群，比较典型的场景为业务使用了计算复杂度较高的命令，例如 ZSET 类型 key 的元素排序命令，当 key 内部元素数量较大时，排序命令的执行会消耗大量的CPU资源，整个命令的执行耗时较长，容易引发慢请求甚至超时断链。

**IO消耗型访问倾斜**一般出在业务频繁调用结果集较大的命令的场景。例如业务频繁调用 hgetall 命令，若碰到 HASH key 的元素数量较大，单次命令的结果集就会很大，当访问频次较高时，容易将实例网络IO打满，造成网络阻塞。另外一种比较典型场景是热key，例如在秒杀场景下，业务会超过频访问个别key，容易引发 CPU 和 网络IO 资源打满。

#### 如何确认是否存在数据倾斜？



查看实例的集群监控、分片监控、节点监控，查看慢日志。

🔗 如何处理数据倾斜 ？

		可能的原因	短期方案	长期方案
空间倾斜	key数量倾斜	hashtag。	实例容量升配。	业务分析改造。
	key大小倾斜	个别复杂数据类型key包含元素个数较多。	实例容量升配。	使用平台的大key分析功能找到大key，业务对大key进行拆分改造。
访问倾斜	CPU消耗型访问倾斜	热key。	开启从只读，提交工单开启代理层热key 缓存。	业务设计优化和访问逻辑优化。
		高复杂度的命令。	开启从只读。	查看慢日志找到复杂命令，业务针对性优化。
	IO 消耗型访问倾斜	热key。	开启从只读，提交工单开启代理层热key 缓存，提交工单升配网络IO。	开启平台热key分析出热 key，业务针优化。
		大key。	开启从只读，提交工单升配网络IO。	开启平台大key分析出大key，业务针优化。
		结果集大的命令。	开启从只读，提交工单升配网络IO。	查看慢日志找到复杂命令，业务针对性优化。

排查Redis实例流量使用率高的问题

Redis实例作为更靠近应用服务的数据层，通常会执行较多的数据存取并消耗网络带宽。不同的实例规格对应的最大带宽有所不同，当超过该规格的最大带宽时，将对应用服务的数据访问性能造成影响。

🔗 步骤一：查询流量使用率

查询实例在指定时段的流量使用量。具体操作，请参见[查看监控数据](#)。

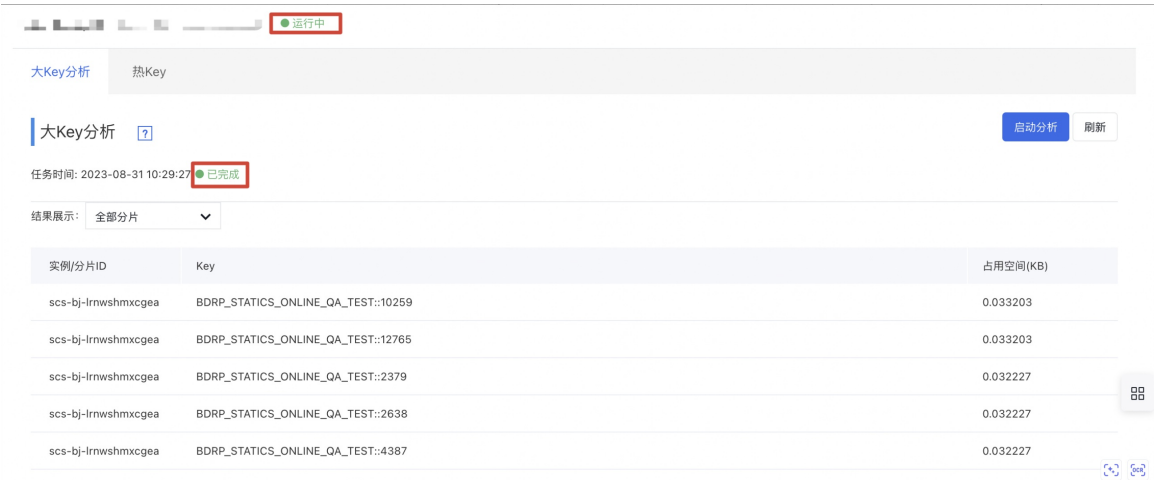
**说明** 需关注的监控指标为节点网络输入量和节点网络输出量。

🔗 步骤二：优化流量使用率

1. 当业务的访问量与预期带宽消耗不匹配，例如流量使用率的增长趋势和QPS的增长趋势明显不一致。您可以通过缓存分析功能，发现实例中存在的大Key。具体操作，请参见[大Key分析](#)。

对大Key（通常大于10 KB）进行优化，例如将大Key拆分、减少对大Key的访问、删除不必要的大Key等。

图 1.大Key分析示例



- 2.经过上述步骤优化后，流量使用率依旧较高，可评估迁移至企业集群版实例，通过多分片以承载更大的网络流量。具体操作，请参见[redis实例迁移方案](#)。

**说明** 在正式升级实例的规格前，您可以先购买一个按量付费的实例，测试要升级到的目标规格是否能够满足业务的负载需求，测试完成后可将其释放。关于如何释放实例，请参见[释放实例](#)。

## 常见Latency（时延）事件的处理建议

本文向您介绍常见Latency事件超过阈值时间的原因及处理建议，您可参考本文了解并处理对应的时延问题。

### 背景信息

原生Redis在2.8.13版引入时延监控（Latency Monitoring）特性，基于事件机制帮助您发现和排查可能的时延问题。该功能仅支持获取最近160秒的数据，且只存取每秒内时延最高的事件。

### 常见Latency事件

Latency事件	事件描述	常见超时原因	处理建议
Eviction Cycle	一次逐出周期的耗时，包含逐出数据的选择、删除操作，及后台线程等待的时间。	可能性较多，需要更进一步分析EvictionDel、EvictionLazyFree事件的耗时。	检查业务是否存在不必要的数 据，并进行优化。扩容实例规 格，尽量避免到达Maxmemory触 发内存逐出。
Eviction Del	在逐出周期中删除Key（键）的耗时。	通常在逐出大Key时产生。	尽量避免使用大Key。* 开 启lazyfree-lazy-eviction参数，使 用异步逐出模式。扩容实例规 格，尽量避免到达Maxmemory触 发内存逐出。
Eviction LazyFree	在逐出周期中，等待后台线程释放内存的耗时。	如果触发逐出时没有符合条件的Key，且 后台线程正在释放内存（例如删除大 Key）时，则需要等待至后台线程释放内 存至Maxmemory以下或后台线程释放内 存结束。	根据业务需求，修 改maxmemory-policy参数，调整 内存逐出策略。扩容实例规格， 尽量避免到达Maxmemory触发内 存逐出。
Expire Cycle	一次清理过期Key周期的耗时。	通常在删除大Key时产生。	尽量避免使用大Key。开启 lazyfree-lazy-expire参数，使用异 步逐出模式。
Expire Del	在清理过期Key周期中，删除Key的耗 时。	通常在删除大Key时产生。	尽量避免使用大Key。开启 lazyfree-lazy-expire参数，使用异 步逐出模式。
AofWrite	写入AOF（AppendOnly File）的耗时。每 次成功写入AOF文件后，会记录AofWrite 事件以及AofWriteAlone、 AofWriteActiveChild、 AofWritePendingFsync三者中的一种事 件。	可能性较多，需要更进一步分析 AofWriteAlone、AofWriteActiveChild、 AofWritePendingFsync事件的耗时。	如果对数据持久化没有要求，建 议可以关闭appendonly参数，关 闭AOF持久化。
AofWriteAlone	一次正常写入AOF文件的耗时。	数据写入量较大，或磁盘性能存在瓶 颈。	如果对数据持久化没有要求，建 议可以关闭appendonly参数，关 闭AOF持久化。
AofWriteActiveChild	写入AOF文件的耗时，写入过程中存在其 他子进程也在向磁盘写数据等情况。	写入AOF文件过程中，其他子进程也在 向磁盘写数据。	如果对数据持久化没有要求，建 议可以关闭appendonly参数，关 闭AOF持久化。
AofWritePendingFsync	写入AOF文件的耗时，写入过程中存在后 台线程正在向磁盘写数据的情况。	写入AOF文件过程中，存在后台线程正 在向磁盘写数据。	如果对数据持久化没有要求，建

fsync	与AOF写入的耗时，与AOF持久化进程正在执行fsync。	与AOF写入的耗时，与AOF持久化进程正在执行fsync。	可以关闭appendonly参数，关闭AOF持久化。
Command	写常规命令（未被标为fast）的耗时。	通常是特殊命令造成，例如执行KEYS命令，遍历所有数据。	使用 <a href="#">查询慢日志</a> 功能，确定具体的慢命令。使用 <a href="#">大Key分析</a> 功能，排查实例是否存在大Key。
FastCommand	被标为fast的命令（命令的时间复杂度为O(1)和O(log N)）的耗时。	通常是对大Key执行命令产生，例如执行GET命令，拷贝大量数据。	使用 <a href="#">查询慢日志</a> 功能，确定具体的慢命令。使用 <a href="#">大Key分析</a> 功能，排查实例是否存在大Key。
Fork	调用Fork操作的耗时。	通常在AOF Rewrite（重写）时产生。	如果是AOF Rewrite导致，且对数据持久化没有要求，建议关闭appendonly参数，关闭AOF持久化，关闭后，通常可以解决此类问题。

更多信息，请参见[Redis Diagnosing latency issues](#)。

### 排查Redis实例内存使用率高的问题

当Redis内存不足时，可能导致Key频繁被逐出、响应时间上升、QPS（每秒访问次数）不稳定等问题，进而影响业务运行。如果发现Redis内存占满或收到内存告警，可参考本文判断内存占用是否长期过高、内存占用是否突然上升、是否发生内存倾斜，并通过拆分大Key，设置过期策略，升级规格等方法解决问题。

#### 内存使用率高的现象分类:

内存使用率高，通常分为以下三种情况：

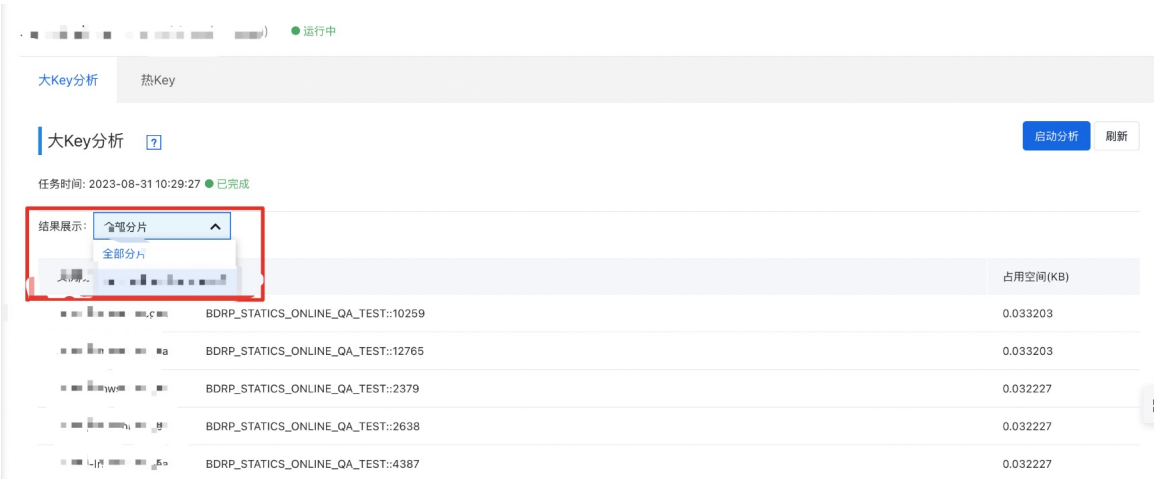
- 内存使用率在较长一段时间内一直处于**较高水位**。通常，当内存使用率超过95%时需要及时关注。
- 内存使用率一直较低，但从某个时间点开始**突然上升至较高水位**，甚至达到100%。
- 实例整体的内存使用率较低，但某个**数据分片节点**的内存使用率**接近100%**。

请根据不同情况，分别采取措施降低内存使用率。

#### 内存使用率长期处于高水位的解决办法

- 查询现有的Key是否符合业务预期，及时清理无用的Key。
- 通过缓存分析功能，分析大Key分布。具体操作，请参见[大Key分析](#)。
  - 对大Key进行评估，然后在业务侧对大Key进行拆分。

图 1. 大Key分析示例



b. 执行 `info keyspace` 命令，分析Key是否有合理的TTL策略。

3.根据业务需求，设置合理的数据逐出策略（即调整`maxmemory-policy`参数的值）。具体操作，请参见[设置实例参数](#)。

**说明** 云数据库Redis默认的数据逐出策略为`volatile-lru`，更多信息，请参见[参数说明](#)。

4.根据业务需求，设置合理的过期Key主动删除的执行频率（即调整`hz`参数的值）。具体操作，请参见[设置实例参数](#)。

**说明** `hz`的取值建议在100以内，如果该值过大将对CPU的使用率产生较大影响。

5.经过上述步骤优化后，内存使用率依旧较高，可评估升级至更大内存的规格，以承载更多数据并改善性能。具体操作，请参见[变更配置](#)。

**说明** 在正式升级实例的规格前，您可以先购买一个按量付费的实例，测试要升级到的目标规格是否能够满足业务的负载需求，测试完成后可将其释放。关于如何释放实例，请参见[释放实例](#)。

## 🔗 内存使用率突然上升的解决办法

**问题原因** 内存使用率突然升高的主要原因如下：

- 短时间内大量写入新数据。
- 短时间内大量创建新连接。
- 突发访问产生大量流量超过网络带宽，导致输入缓冲区和输出缓冲区积压。
- 客户端处理速度跟不上Redis的处理速度，导致输出缓冲区积压。 **解决方案** 请依次排查内存使用率突然升高的原因，并参考对应的解决方案解决问题。 **排查是否写入大量新数据 排查方法：**

查看Redis实例监控的**节点网络输入量**、**写请求**。如果节点网络输入量与写请求的趋势与内存使用率趋势一致，说明大量的数据写入导致内存使用率突然升高。

**解决方案：**

1. 通过设置Key的过期时间自动清理不再需要的Key，或手动删除不再需要的Key。
2. 升级实例规格，通过增加内存容量缓解内存使用率升高的问题。详情请参见[变更配置](#)。
3. 如果您的实例为标准版，扩容内存规格后仍无法解决内存使用率高的问题，可以考虑升级为企业集群版，将数据分布到多个数据分片节点上，减轻单个数据分片节点的内存压力。详情请参见[redis实例迁移方案](#)。

**排查是否创建大量新连接 排查方法：**

查看Redis实例监控的**节点实例连接数**。如果节点实例连接数突然增长，且与内存使用率趋势一致，说明大量新建连接导致内存使用率突然升高。 **解决方案：**

1. 排查是否存在连接泄漏。
2. 设置连接超时时间（即调整`timeout`参数的值），自动关闭空闲连接。详情请参见[设置实例参数](#)。

**排查是否突发流量导致输入和输出缓冲区积压 排查方法：**

1. 查看实例监控的**节点网络输出量**，如果节点网络输出量突然增长到高位，可能存在缓存积压。
2. 执行**MEMORY STATS**命令，查看**clients.normal**占用的内存是否过多。说明**clients.normal**反映了所有普通客户端连接的输入和输出缓冲区所占用的内存总量。

**解决方案：**

1. 排查业务流量突发原因。
2. 升级实例规格，以保证输入输出缓冲区的使用。详情请参见[变更配置](#)。

排查是否因客户端性能问题导致输出缓冲区积压

排查方法：

在Redis中，执行MEMORY DOCTOR命令，查看big\_client\_buf的值。当big\_client\_buf=1时，代表至少有一个客户端的输出缓冲区占用内存较大。

解决方案：

执行CLIENT LIST命令，查看哪个客户端的输出缓冲区内存占用量（omem）较大。排查该客户端应用是否存在性能问题。

🔗 数据分片节点内存使用率高的解决方法

**现象** 如果Redis实例为集群版，您可能从以下一种或几种现象发现数据分片节点的内存使用率高：

- 收到了云监控的内存使用率告警。告警信息中显示某个数据节点的内存使用率高，超过了阈值。
- 在实例监控页面查看实例的内存使用率和数据节点的内存使用率，发现实例的内存使用率不高，但某个数据分片节点的内存使用率较高。**问题原因** 如果实例的内存使用率不高，但某个数据分片节点的内存使用率较高，说明发生了内存倾斜。**解决方案** 检查是否存在大Key，并拆分大Key

**查找大Key:**通过[大Key分析](#)找出大Key。

**拆分大Key:**例如将含有数万成员的一个HASH Key拆分为多个HASH Key，并确保每个Key的成员数量在合理范围。在Redis集群架构中，拆分大Key能对数据分片间的内存平衡起到显著作用。更多方法请参见[避免大Key热Key实践指南](#)。**检查是否使用了Hash Tag** 如果使用了Hash Tag，请根据业务实际情况，评估将一个Hash Tag拆分为多个Hash Tag，使数据更加均匀地分布在不同的数据分片节点上。**扩容实例规格** 扩容实例规格，可以增加实例每个分片的内存，是改善内存倾斜的临时解决方案，具体操作请参见[变更配置](#)。

**重要** 在成功升级实例规格后，会改善内存倾斜问题，但可能也引起带宽倾斜或CPU倾斜。

🔗 附录1：Redis内存占用介绍

Redis的内存占用主要由以下三部分组成：

内存占用	说明
链路内存（动态）	主要包括Input Buff、Output Buff、JIT Overhead、Fake Lua Link、Lua执行缓存等，例如可执行INFO命令，通过返回结果的Clients中查看客户端缓存信息。   <b>说明</b> Input buff与Output buff与每个客户端的连接有关，通常较小。当执行客户端Range类操作或大Key收发较慢时，Input buff与Output buff占用的内存会增大，从而影响数据区，甚至会造成内存溢出OOM（Out Of Memory）。
数据内存	用户数据区，即实际存储的Value信息，通常作为重点分析的对象。
管理内存（静态）	启动时较小且相对恒定，该区域由管理数据的Hash内存开销、Repl-buff与aof-buff的内存开销（约32 MB~64 MB）等构成。说明 当Key数量特别多时（例如几亿个），会占用较大的内存。

**说明** 大部分OOM场景是由于动态内存管理失效引起，例如限时请求堆积导致动态内存快速上升、过于复杂或不合理的Lua脚本也可能导致OOM。

🔗 附录2：查看内存占用情况的其他方法

使用[MEMORY STATS](#)命令查看内存占用情况。

使用[MEMORY DOCTOR](#)命令查看内存诊断建议。



使用[MEMORY USAGE](#)命令查看指定Key占用的内存。

## 排查Redis实例CPU使用率高的问题

Redis CPU使用率升高可能是由于以下三种原因：高并发、高吞吐的业务消耗较多CPU资源，如果CPU资源未达到瓶颈，属于正常业务场景；业务运行超预期，Redis实例的CPU资源无法满足业务需求，可通过增加分片数（企业集群版）、副本数来解决资源瓶颈；使用不当，例如高消耗命令、热Key、大Key等，导致CPU使用率异常升高。当平均CPU使用率高于70%、连续5分钟内的CPU平均峰值使用率高于90%时，您需要及时关注并排查该问题，以保障应用的稳定运行。

### 🔗 哪些因素会导致CPU使用率异常升高

- **高消耗命令**：即时间复杂度为O(N)的命令，其中N为较大值，例如KEYS、HGETALL或使用MGET、MSET、HMSET、HMGET一次操作大量Key等。通常情况下，命令的时间复杂度越高，在执行时会消耗越多的资源，从而导致CPU使用率上升。由于命令执行单元为单线程的特性，Redis在执行高消耗命令时会引发排队导致应用响应变慢。极端情况下，甚至可能导致实例被整体阻塞，引发应用超时中断或流量跳过缓存层直接到达后端的数据库侧，引发雪崩效应。

说明：关于各命令对应的时间复杂度信息，请参见[Redis官网](#)。

- **热Key**：某个或某部分Key的请求访问次数显著超过其他Key时，代表此时可能产生了热Key。热Key将会消耗Redis的大量CPU资源，从而影响其他Key的访问时延。并且，在企业集群版架构中，如果热Key较为集中地分布在部分数据分片节点，可能会导致CPU使用率倾斜（个别分片的CPU使用率远超其他分片）。
- **大Key**：大Key会占用更多的内存，同时，对大Key的访问会显著增加Redis的CPU负载和流量。大Key在一定程度上更容易形成热点从而造成CPU使用率高。如果大Key较为集中地分布在部分数据分片节点，可能会导致CPU使用率倾斜、带宽使用率倾斜及内存使用率倾斜。
- **短连接**：频繁地建立连接，导致Redis实例的大量资源消耗在连接处理上。
- **AOF**：百度Redis实例默认开启了AOF（append-only file），当实例处于高负载状态时，AOF的写盘行为将会导致CPU使用率升高及实例整体的响应时延增加。

### 🔗 CPU使用率异常升高的现象分类

CPU使用率高，主要分为以下三种现象：

- 在某个时间段，**CPU使用率突然升高**，甚至到100%。原因排查和解决方案，请参见下文。
- 某个数据节点的**CPU使用率较高**，而其他数据节点的CPU使用率较低。原因排查和解决方案，请参见下文。
- 某个Proxy节点的**CPU使用率较高**，而其他Proxy节点的CPU使用率较低。原因排查和解决方案，请参见下文。请根据不同现象，分别采取措施降低CPU使用率。

### 🔗 CPU使用率突然升高

如果实例全局的CPU使用率升高，可参考以下步骤排查并优化。 **\*\*排查并禁用高消耗命令\*\*** **\*\*排查步骤\*\***

1. 通过性能监控功能，确认CPU使用率高的具体时间段。具体操作，请参见[查看监控数据](#)。
  2. 通过下述方法，找出高消耗的命令：
- 慢日志功能会记录执行超过指定时间阈值的命令。根据指定时间段和节点的慢查询语句和执行时长，可找出执行时间较长的高消耗命令。具体操作，请参见[慢日志](#)。

图 1. 慢日志查询示例

![image.png](http://bjhw-bce-online-public0.bjhw.baidu.com:8109/proxy-external/?url=http://bce-cdn.bj.bcebos.com/doc/bce-doc/SCS/image\_65fa2ef.png)

**\*\* 解决办法 \*\***

- 评估并禁用高风险命令和高消耗命令，例如FLUSHALL、KEYS、HGETALL等。具体操作，请参见[禁用危险命令](#)。
- 可选：根据业务情况，选择下述方法对实例进行调整：
  - 调整实例[开启副本只读](#)，对高消耗命令或应用进行分流。

### \*\* 排查并优化短连接\*\* \*\*排查步骤\*\*

1. 通过性能监控功能，确认CPU使用率高的具体时间段。具体操作，请参见[查看监控数据](#)。
2. 在性能监控页面，查看是否有CPU使用率较高，连接数较高，但QPS（每秒访问次数）未达到预期的现象。如果有，说明可能存在短连接，请参考下文的解决方法。

### \*\*解决方法\*\*

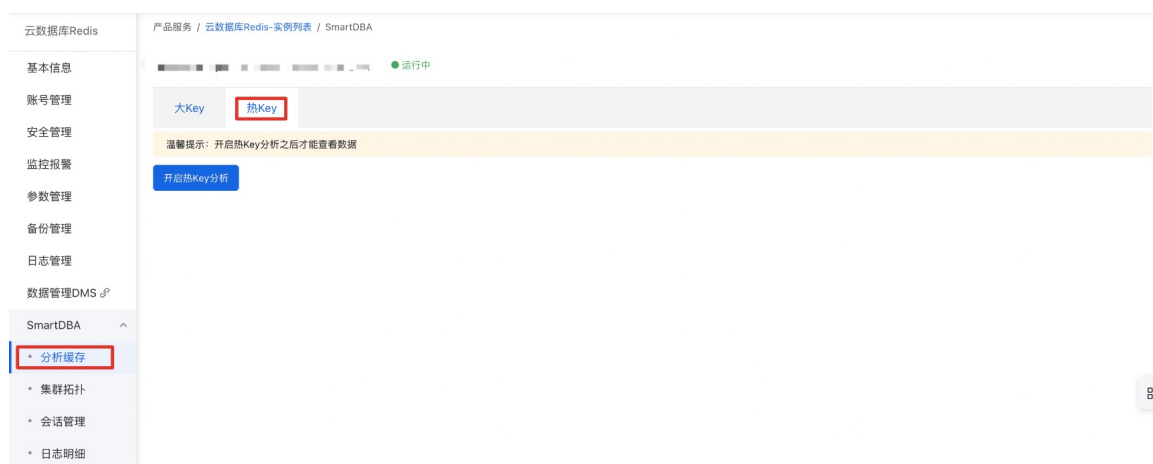
- 将短连接调整为长连接，例如使用JedisPool连接池连接。 **\*\*关闭AOF\*\*** Redis实例默认开启了AOF（append-only file），当实例处于高负载状态时，频繁地执行AOF会一定程度上导致CPU使用率升高。 **\*\*评估服务能力\*\*** 经过上文的步骤排查优化后，在业务正常运行的情况下，还是经常遇到实例整体的负载较高（平均CPU使用率在70%以上），可能存在性能瓶颈。首先，应排查是否存在异常的业务访问，例如异常的命令、来自某台应用主机的异常访问等，此类情况需要从业务上进行优化。如果均为正常访问，此时的高负载是正常业务行为，为保障业务平稳运行，建议升级实例的规格，或将其迁移到[企业集群版](#)并考虑[开启副本只读](#)。关于如何迁移到企业集群版，请参考[redis实例迁移方案](#)。

**说明** 为保障业务的正常运行，在正式升级实例前，建议先购买一个按量付费的实例，完成业务负载和兼容性测试，测试完成后可将其释放。

## 🔗 数据节点CPU使用率不一致

如果Redis实例为[企业集群版](#)，您发现实例的部分数据分片节点的CPU使用率高，而其他数据分片节点的CPU使用率较低，可参考以下步骤排查并优化。 **\*\*排查并优化热点Key\*\* \*\*排查步骤\*\***

1. 通过性能监控功能，确认CPU使用率高的具体时间段。具体操作，请参见[查看监控数据](#)。
2. 在实例详情页的左侧导航栏中，点击 **SmartDBA**，点击**热Key**页签，切换热key页面，单击**开启热Key分析**按钮。具体操作，请参见[热key分析](#)。



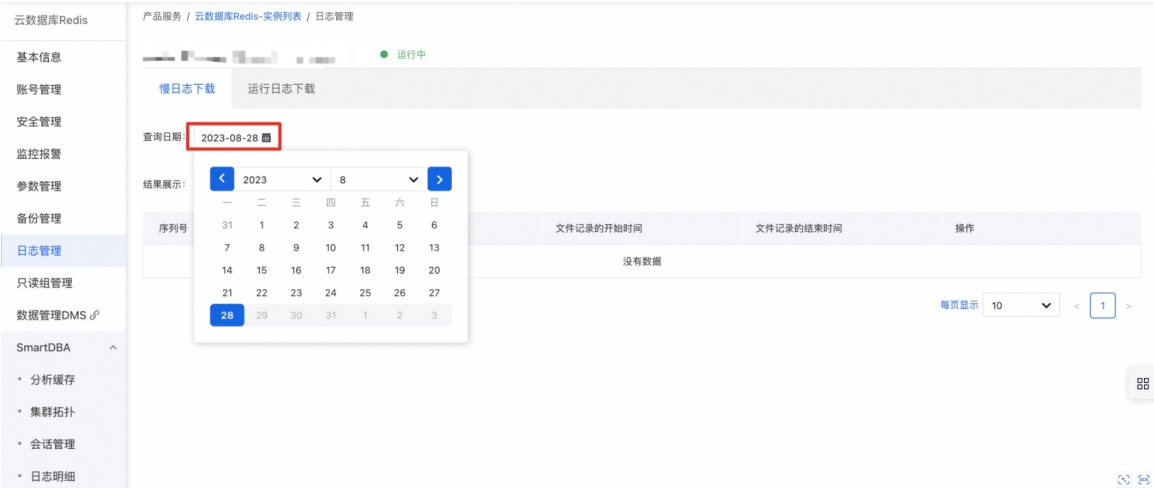
**\*\*解决**

**方法\*\*** 如果热Key的产生来自于读请求，您可以将[实例开启副本只读](#)来降低每个数据分片的读请求压力。

**说明** 在请求量极大的场景下，从副本和主副本之间会产生不可避免的延迟，此时会有读取到脏数据的问题。因此，在读、写压力都较大且对数据一致性要求很高的场景下，开启副本只读并不是最优方案。 **\*\*排查并禁用高消耗命令\*\* \*\*排查步骤\*\***

1. 通过性能监控功能，确认CPU使用率高的具体时间段。具体操作，请参见[查看监控数据](#)。
2. 通过下述方法，找出高消耗的命令：
  - 慢日志功能会记录执行超过指定时间阈值的命令。根据指定时间段和节点的慢查询语句和执行时长，可找出执行时间较长的高消耗命令。具体操作，请参见[慢日志](#)。

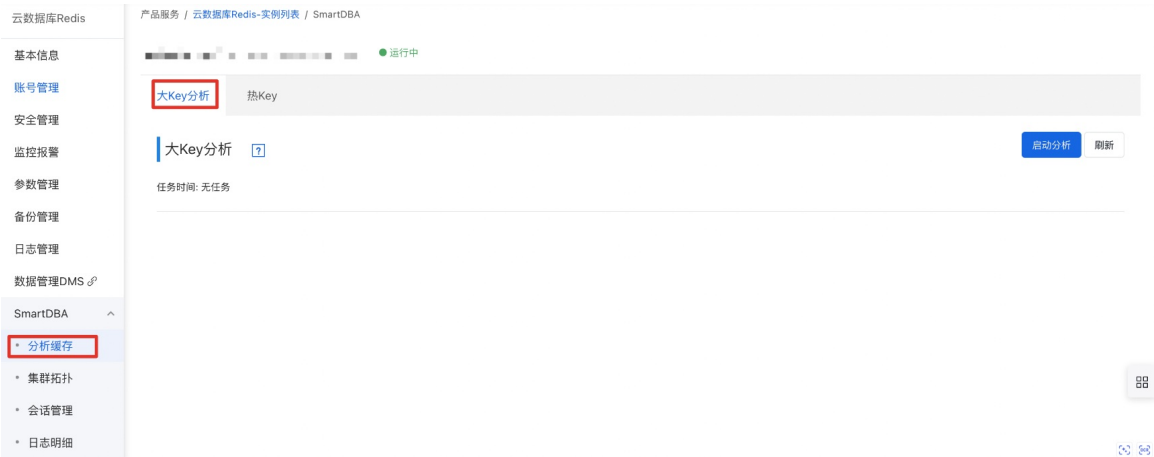
图 1. 慢日志查询示例



**\*\*解决办法\*\*** 评估并禁用高风险命令和高消耗命令，例如FLUSHALL、KEYS、HGETALL等。具体操作，请参见[禁用危险命令](#)。

**\*\*排查并优化大Key\*\*** **\*\*排查步骤\*\***

1. 通过性能监控功能，确认CPU使用率高的具体时间段。具体操作，请参见[查看监控数据](#)。
2. 在实例详情页的左侧导航栏中，点击 SmartDBA，默认选中大Key页签，单击启动分析按钮。具体操作，请参见[大Key分析](#)。



**\*\*解决方法\*\*** 根据业务的实际情况，将大Key拆分为小Key，以分散请求压力。

**代理节点CPU使用率不一致**

如果Redis实例为[企业集群版](#)，您发现实例的部分Proxy节点的CPU使用率高，而其他代理分片节点的CPU使用率较低，可参考以下步骤排查并优化。 **\*\*排查步骤\*\*** 在监控报警的代理监控页面，确认实例连接数、ProxyQPS是否均衡。具体操作，请参见[查看监控数据](#)。 **\*\*解决方法\*\*** 根据连接数使用率是否均衡，采取下述措施：

- 不均衡：重启业务所属的客户端或Proxy节点使连接重分配。重启Proxy节点，请联系百度智能云同学。
- 均衡：通常因pipeline或batch的操作规模过大引起，需要减少对应的操作规模，例如将其拆分为多个操作来执行。

**相关文档**

- 查询过去一个月内指定时间段的CPU使用率、内存使用率、流量使用率等监控指标，请参见[查看监控数据](#)。
- 设置CPU使用率报警，当CPU使用率超过阈值，系统将发送告警信息到告警接收人。请参见[设置报警规则](#)。
- 处理大Key、热Key的详细方法，请参见[避免大Key热Key实践指南](#)。

# API参考



## API简介

Redis API依据BCE API规范进行设计，符合简单易用原则，对外提供 Restful API。

Redis API对于每个 HTTP 请求，认证签名放在 Authorization 头域中，后端统一认证。且后端在 Response 头域中会添加 x-bce-request-id，作为请求唯一标识，方便追踪定位问题。

如果您是初次调用百度智能云产品的API，可以观看[API入门视频指南](#)，快速掌握调用API的方法。

## 使用须知

### API 认证机制

用户可以通过两种方式与百度智能云进行交互，包括认证方式和匿名方式。对于认证方式，需要通过使用 Access Key Id / Secret Access Key 加密的方法来验证某个请求的发送者身份。Access Key Id（AK）用于标示用户，Secret Access Key（SK）是用户用于加密认证字符串和百度智能云用来验证认证字符串的密钥，其中 SK 必须保密，只有用户和百度智能云知道。

当百度智能云接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。鉴权认证机制的详细内容请参见[鉴权认证机制](#)。

### 排版约定

排版格式	含义
< >	变量
[ ]	可选项
{ }	必选项
	互斥关系
等宽字体 Courier New	屏幕输出

### 时间与日期

日期与时间的表示有多种方式。为统一起见，除非是约定俗成或者有相应规范的，凡需要日期时间表示的地方一律采用 UTC 时间，遵循 ISO 8601，并做以下约束：

- 表示日期一律采用 YYYY-MM-DD 方式，例如 2016-06-01 表示 2016年6月1日
- 表示时间一律采用 hh:mm:ss 方式，并在最后加一个大写字母 Z 表示 UTC 时间。例如23:00:10Z表示 UTC 时间 23点0分10秒。
- 凡涉及日期和时间合并表示时，在两者中间加大写字母 T，例如 2016-06-01T23:00:10Z 表示 UTC 时间2016年6月1日23点0分10秒

## 通用说明

### API 服务域名

Redis API 的服务域名包括：

Region	EndPoint	Protocol
北京	redis.bj.baidubce.com	HTTP and HTTPS
保定	redis.bd.baidubce.com	HTTP and HTTPS
广州	redis.gz.baidubce.com	HTTP and HTTPS
苏州	redis.su.baidubce.com	HTTP and HTTPS
武汉	redis.fwh.baidubce.com	HTTP and HTTPS
香港	redis.hkg.baidubce.com	HTTP and HTTPS

通信协议

API 调用遵循 HTTP 协议，各 Region 采用不同的域名，具体域名为 redis.{region}.baidubce.com。数据交换格式为 JSON，所有 request/response body 内容均采用 UTF-8 编码。

请求参数

请求参数包括如下4种：

参数类型	说明
URI	通常用于指明操作实体,如:POST /v{version}/instance/{instanceId}
Query参数	URL中携带的请求参数
HEADER	通过HTTP头域传入,如:x-bce-date
Request Body	通过JSON格式组织的请求数据体

返回值

返回值分为两种形式：

返回内容	说明
HTTP STATUS CODE	如200,400,403,404等
Response Body	JSON格式组织的响应数据体

API 版本号

参数	类型	参数位置	描述	是否必须
version	string	URL参数	API版本号，当前API版本为v1	是

认证机制

所有API的安全认证一律采用Access Key与请求签名机制。 Access Key由Access Key ID和Secret Access Key组成，均为字符串。

对于每个HTTP请求，使用下面所描述的算法生成一个认证字符串。提交认证字符串放在Authorization头域里。服务端根据生成算法验证认证字符串的正确性。认证字符串的格式为bce-auth-v{version}/{accessKeyId}/{timestamp}/{expirationPeriodInSeconds}/{signedHeaders}/{signature}。

- version是正整数。
- timestamp是生成签名时的UTC时间。
- expirationPeriodInSeconds表示签名有效期限。
- signedHeaders是签名算法中涉及到的头域列表。头域名之间用分号（;）分隔，如host;x-bce-date。列表按照字典序排列。（本API签名仅使用host和x-bce-date两个header）

- signature是256位签名的十六进制表示，由64个小写字母组成。

🔗 密码加密传输规范定义

所有涉及密码的接口参数都需要加密，禁止明文传输。密码一律采用AES 128位加密算法进行加密，用SK的前16位作为密钥，加密后生成的二进制字节流需要转成十六进制，并以字符串的形式传到服务端。具体步骤如下：

- byte[] bCiphertext= AES(明文,SK)
- String strHex = HexStr(bCiphertext)

🔗 幂等性

当调用创建接口时如果遇到了请求超时或服务器内部错误，用户可能会尝试重发请求，这时用户通过clientToken参数避免创建出比预期要多的资源，即保证请求的幂等性。

幂等性基于clientToken，clientToken是一个长度不超过64位的ASCII字符串，通常放在query string里，如http://redis.bj.baidubce.com/v1/instance?clientToken=be31b98c-5e41-4838-9830-9be700de5a20。

如果用户使用同一个clientToken值调用创建接口，则服务端会返回相同的请求结果。因此用户在遇到错误进行重试的时候，可以通过提供相同的clientToken值，来确保只创建一个资源；如果用户提供了一个已经使用过的clientToken，但其他请求参数（包括queryString和requestBody）不同甚至url Path不同，则会返回IdempotentParameterMismatch的错误代码。

clientToken的有效期为24小时，以服务端最后一次收到该clientToken为准。也就是说，如果客户端不断发送同一个clientToken，那么该clientToken将长期有效。

使用 Redis API 的用户需要实名认证，没有通过实名认证的可以前往[百度开放云官网控制台](#)中的安全认证下的实名认证中进行认证。没有通过实名认证的用户请求将会得到一下错误提示码：

错误码	错误描述	HTTP状态码	中文解释
QualifyNotPass	The User has not pass qualify.	403	账号没有通过实名认证

错误返回

🔗 错误返回格式

Redis 错误响应符合BCE规范，统一为如下格式。后续各接口不再单独列出。

参数名	类型	说明
code	string	错误码
message	string	错误描述
requestId	string	本次请求的requestId

示例：

```
{
 "requestId": "ae2225f7-1c2e-427a-a1ad-5413b762957d",
 "code": "AccessDenied",
 "message": "Access denied."
}
```

其中，code为错误码，所有错误码取值来源BCE公共错误码和Redis专有错误码。

🔗 BCE 公共错误码

错误返回码	错误消息	状态码	说明
-------	------	-----	----

AccessDenied	Access denied.	403 Forbidden	无权限访问对应的资源。
InappropriateJSON	The JSON you provided was well-formed and valid, but not appropriate for this operation.	400 Bad Request	请求中的JSON格式正确，但语义上不符合要求。如缺少某个必需项，或值类型不匹配等。出于兼容性考虑，对于所有无法识别的项应直接忽略，不应该返回这个错误。
InternalServerError	We encountered an internal error Please try again.	500 Internal Server Error	所有未定义的其他错误。在有明确对应的其他类型的错误时（包括通用的和服务自定义的）不应该使用。
InvalidAccessKeyId	The Access Key ID you provided does not exist in our records.	403 Forbidden	Access Key ID不存在。
InvalidHTTPAuthHeader	The Access Key ID you provided does not exist in our records.	400 Bad Request	Authorization头域格式错误。
InvalidHTTPRequest	There was an error in the body of your HTTP request.	400 Bad Request	HTTP body格式错误。例如不符合指定的Encoding等。
InvalidURI	Could not parse the specified URI.	400 Bad Request	URI形式不正确。例如一些服务定义的关键词不匹配等。对于ID不匹配的问题，应定义更加具体的错误码，如NoSuchKey。
MalformedJSON	The JSON you provided was not well-formed.	400 Bad Request	JSON格式不合法。
InvalidVersion	The API version specified was invalid.	404 NotFound	URI的版本号不合法。
OptInRequired	A subscription for the service is required.	403 Forbidden	没有开通对应的服务。
PreconditionFailed	The specified If-Match header doesn't match the ETag header.	412 PreconditionFailed	详见Etag。
RequestExpired	Request has expired. Timestamp date is <Data>.	400 Bad Request	请求超时。要改成x-bce-date。若请求中只有Date，需将Date转成datetime。
IdempotentParameterMismatch	The request uses the same client token as a previous, but non-identical request.	403 Forbidden	clientToken对应的API参数不一样。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access	400 Bad	Authorization头域中附带的签名和服务端验证不一

otMatch	Key and signing method. Consultthe service documentation for details.	Reques t	致。
---------	-----------------------------------------------------------------------	-------------	----

错误码

参考各接口错误码。

## API规格

公共请求头

下表列出了所有 Redis API 所携带的公共头域。HTTP 协议的标准头域不在此处列出。

头域 (Header)	是否必须	说明
Authorization	必须	包含Access Key与请求签名
Content-Type	可选	application/json; charset=utf-8
x-bce-date	可选	表示日期的字符串，符合BCE API规范
x-bce-content-sha256	可选	表示内容部分的SHA256签名的十六进制字符串。这里内容指HTTP Request Payload Body。即Content部分在被HTTP encode之前的原始数据

公共头域将在每个Redis API中出现，是必需的头域，其中x-bce-content-sha256头域只出现在POST和PUT请求中。

POST、PUT、DELETE等请求数据放在request body中。

公共响应头

下表列出了所有 Redis API 的公共响应头域。HTTP 协议的标准响应头域不在此处列出。

头域 (Header)	说明
Content-Type	只支持JSON格式， application/json; charset=utf-8
x-bce-request-id	Redis后端生成，并自动设置到响应头域中

其中，request id使用UUID version4由Redis服务生成。

## 订单服务

### 创建实例

请求说明

- 本接口用于创建一个或多个Redis实例。
- 创建实例需要实名认证，没有通过实名认证的可以前往百度开放云官网控制台中的安全认证下的实名认证中进行认证。
- 创建计费方式为按量付费的实例需要账户现金余额+通用代金券大于100；包年包月方式的实例则需要账户现金余额大于等于实例费用。
- 支持批量创建，且如果创建过程中有一个实例创建失败，所有实例将全部回滚。
- 创建接口为异步创建，可通过查询指定实例详情接口查询实例状态。
- 当前版本为V2。

请求结构

```
POST /v2/instance?clientToken={clientToken} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "billing": {
 "paymentTiming": "paymentTiming",
 "reservation": {
 "reservationLength":reservationLength
 }
 },
 "purchaseCount": purchaseCount,
 "instanceName": "instanceName",
 "nodeType": "nodeType",
 "shardNum": shardNum,
 "proxyNum": proxyNum,
 "clusterType": "clusterType",
 "port": port,
 "engineVersion": "engineVersion",
 "vpclId": "vpclId",
 "replicationInfo": [
 {
 "availabilityZone": "availabilityZone",
 "subnetId": "subnetId",
 "isMaster": isMaster
 }
],
 "tags": [
 {
 "tagKey": "tagKey",
 "tagValue": "tagKey"
 }
],
 "resourceGroupId": "resourceGroupId",
 "confTpl": "confTpl",
 "autoBackupConfig": "Tue,Wed,Thu,Fri,Sta,Sun,Mon;16:20:01;5"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数	类型	是否必须	参数位置	描述
clientToken	String	是	URL看参数	幂等性Token，是一个长度不超过64位的ASCII字符串
billing	Billing	是	Request Body参数	计费相关参数。结构及取值参考 <a href="#">计费信息</a>
instanceName	String	是	Request Body参数	用户自定义实例名称，要求： 1) 支持大小写字母、数字、中文以及_/.等特殊字符，必须以字母或者中文开头 2) 长度限制为1-64
nodeType	String	是	Request Body参数	实例规格。 Redis内存型和Redis容量型时为必填项。具体规格通过 <a href="#">获取实例规格列表</a> 接口获取。 Memcache时为非必填项，此参数无效。

port	int	是	Request Body参数	端口号可选1025-7041、7043-22221、22223-65534，创建后支持更改
engine	int	否	Request Body参数	引擎类型，默认值为2。memcache:1 ,redis:2, Redis容量型(原PegaDB):3
engineVersion	String	是	Request Body参数	数据库版本： Redis企业集群版：4.0、5.0、6.0 Redis社区标准版：4.0、5.0、6.0、7.0
storeType	int	否	Request Body参数	存储类型，默认为0。高性能内存:0、容量型存储：3
enableReadOnly	int	否	Request Body参数	副本只读，默认值为2。 打开：1、关闭：2
purchaseCount	int	是	Request Body参数	购买个数，最大不超过10，默认1
shardNum	int	否	Request Body参数	分片个数，默认值为1
proxyNum	int	是	Request Body参数	代理节点数，目前支持的取值：主从版：0 集群版：代理节点数=分片个数 集群版分片数为1时，代理节点数量为2
clusterType	String	是	Request Body参数	集群类型： 企业版集群："cluster" 主从版："master_slave" mamcache："default"
diskFlavor	int	否	Request Body参数	单分片存储空间,引擎类型为3，即为Redis容量型(原PegaDB)时传入
diskType	String	否	Request Body参数	存储类型:目前支持 "cds"
vpclId	String	否	Request Body参数	vpclId，如果不提供则属于默认vpc 可通过调用查询VPC列表接口查询可用的VPC 如果传入该参数，则副本信息replicationInfo字段必传，并且副本信息中的子网需要是该vpclId所属子网
replicationInfo	List <ReplicationMap>	否	Request Body参数	副本信息。 需要创建多少副本，该数组就传多少项，数组中的一项代表一个副本 可用区和子网所属关系须一致 子网和vpclId所属关系须一致 主节点有且仅有一个 memcache仅支持单个副本 Redis和Redis容量型最多支持10副本 Redis集群版和Redis容量型最少2副本，Redis标准版支持单副本 可用区和子网只能是一一对应的关系

				部署集ID列表。该参数不传，默认为共享集群。
autoRenewTimeUnit	String	否	Request Body参数	按月付费或者按年付费 月是"month"，年是"year"
autoRenewTime	int	否	Request Body参数	自动续费的时间 按月是1-9 按年是 1-3
bgwGroupId	String	否	Request Body参数	blb专属集群Id。该参数不传，默认为共享集群。
clientAuth	String	否	Request Body参数	密码长度8~16位，至少包含字母、数字和特殊字符中两种。允许的特殊字符包括\$^*()_+=,密码需要加密传输，禁止明文传输，详情请参考 <a href="#">密码加密传输规范定义</a>
tags	List <Tag>	否	Request Body参数	标签键值对列表
confTpl	String	否	Request Body参数	指定的参数模版ID
resourceGroupId	String	否	Request Body参数	资源分组ID。资源分组信息可通过 <a href="#">资源管理接口</a> 查询。
autoBackupConfig	String	否	Request Body参数	备份配置。格式："\${备份周期};\${备份时间};\${备份时长}"。 备份周期取值参考：周一：'Mon', 周二：'Tue', 周三： 'Wed', 周四：'Thu',周五：'Fri', 周六：'Sta', 周日：'Sun'。 备份时间为UTC时间，如16:00:00，北京时间为00:00:00。 备份时间：取值1-15
deployIdList	List<String>	否	Request Body参数	部署集ID列表。建议仅传一个部署集ID，即此列表长度不能大于1。部署集ID通过 <a href="#">部署集列表接口</a> 获取。

ReplicationMap

参数名称	类型	描述	
availabilityZone	String	可通过调用获取可用区列表接口查询可用区列表。zoneName命名规范是“国家-region-可用区序列”，小写，例如北京可用区A为"cn-bj-a"	
subnetId	String	子网Id	
isMaster	Int	是否是主节点 1代表主 0代表从。主节点有且仅有一个。	

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	类型	描述
instanceIds	List<String>	由InstanceId组成的数组
orderId	String	订单ID



## 请求示例

```
POST /v2/instance?clientToken=be31b98c-5e41-4838-9830-9be700de5a20 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de

{
 "billing": {
 "paymentTiming": "Prepaid",
 "reservation": {
 "reservationLength": 1
 }
 },
 "purchaseCount": 1,
 "instanceName": "scs_name",
 "nodeType": "cache.n1.medium",
 "shardNum": 1,
 "proxyNum": 0,
 "clusterType": "master_slave",
 "replicationNum": 2,
 "port": 6379,
 "engineVersion": "5.0",
 "vpclId": "vpc-1n1wqx4iuu",
 "replicationInfo": [
 {
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-rvv87cdd0gv9",
 "isMaster": 1
 },
 {
 "availabilityZone": "cn-bj-b",
 "subnetId": "sbn-rvv87cdd0gv9",
 "isMaster": 0
 }
],
 "autoBackupConfig": "Tue,Wed,Thu,Fri,Sat,Sun,Mon;16:20:01;5"
}
```

## 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "instanceIds": ["scs-bj-cxisuflkquj"],
 "orderId": "edab21e8b38c45deb28926279ff4fab"
}
```

## 变更配置

### 请求说明

- 本接口用于变更一个redis实例的规格。
- 变更实例需要实名认证，没有通过实名认证的可以前往百度开放云官网控制台中的安全认证下的实名认证中进行认证。
- 变更计费方式为后付费的实例需要账户现金余额+通用代金券大于100；预付费方式的实例则需要账户现金余额大于等于实例费用。

- 变更配置操作只允许对单一实例进行操作，变更过程失败，操作回滚，费用退回。
- 只有实例正常运行状态时才可以进行扩缩容操作，变更接口为异步变更，可通过查询指定实例详情接口查询实例状态。
- 当前版本为v1。

请求结构

```
PUT /v1/instance/{instanceId}/change?clientToken={clientToken} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "billing": {
 "paymentTiming": "paymentTiming",
 },
 "engineVersion": "engineVersion",
 "nodeType": "nodeType",
 "shardNum": "shardNum"
 "diskFlavor": diskFlavor
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数	类型	是否必须	参数位置	描述
instanceId	String	是	URL参数	实例ID
clientToken	String	是	URL参数	幂等性Token，是一个长度不超过64位的ASCII字符串
billing	billing	是	RequestBody参数	计费相关参数。结构及取值参考 <a href="#">计费信息</a>
engineVersion	String	是	RequestBody参数	引擎版本。例如redis 3.2/4.0/5.0/6.0 等
nodeType	String	否	RequestBody参数	参见附录中 <a href="#">实例规格</a>
shardNum	Integer	否	RequestBody参数	分片个数
diskFlavor	Integer	否	RequestBody参数	存储空间

- nodeType和shardNum只能传入其中1个参数，不能同时变更

返回头域

除公共头域外，无其它特殊头域。

请求示例

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/change HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de

{
 "billing": {
 "paymentTiming": "paymentTiming",
 },
 "engineVersion": "engineVersion",
 "nodeType": "nodeType",
 "diskFlavor": 1
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
ContentType: application/json
```

释放实例

请求说明

- 释放单个Redis实例，释放后实例将进入回收站，保留7天后自动删除所有物理资源。
- 可以从回收站内恢复实例或者彻底删除实例,预付费实例需要通过续费实例接口进行恢复。
- 当前版本为v1。

请求结构

```
DELETE /v1/instance/{instanceId} HTTP/1.1
HOST: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceId	String	是	URL参数	实例Id

返回头域

除公共头域，无其它特殊头域。

请求示例

```
DELETE /v1/instance/redis-linkhx2G HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

错误码

错误码	错误描述	HTTP状态码	中文解释
BadRequest	Bad request parameters or illegal request	400	请求有误
OperationDenied	Prepay type cannot be deleted	409	预付费且未过期的实例不能释放
InstanceNotExistException	Instance not found	404	实例未找到或非自己的实例
InstanceDeleteFailed	Instance {instanceId} delete failed	403	删除scs实例失败
ServiceUnavailable	Service is unavailable	503	内部服务器错误

获取创建实例价格

描述

- 本接口用于获取创建指定规格Redis实例的价格,包年包月计费单位为元/月，按量付费计费单位为元/分钟。
- 当前版本为v1

请求结构

```
POST /v1/price
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "nodeType": "pega.g4s1.micro",
 "clusterType": "cluster",
 "engine": 3,
 "shardNum": 2,
 "replicationNum": 2,
 "instanceNum": 1,
 "diskFlavor": 50,
 "chargeType": "postpay",
 "period": 1,
 "timeUnit": "YEAR"
}
```

请求参数

参数	类型	是否必须	参数位置	描述
clientToken	String	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串。
engine	Integer	否	RequestBody参数	引擎类型。默认值为2。取值范围如下： Memcache: 1 Redis内存型: 2 Reids 容量型(原 PegaDB): 3
clusterType	String	否	RequestBody参数	集群类型。默认值为master_slave。取值范围如下： Redis内存型标准版：master_slave Redis内存型集群版：cluster Redis容量型（Pegadb）集群版：cluster Memcache: default
nodeType	String	否	RequestBody参数	实例规格。 Redis内存型和Redis容量型时为必填项。具体规格通过 <a href="#">获取实例规格列表接口</a> 获取。 Memcache时为非必填项，此参数无效。
cacheInstanceType	Integer	否	RequestBody参数	Memcache的节点规格。单位GB。 取值范围：1、2、4、8、16、32、64。
shardNum	Integer	否	RequestBody参数	分片个数，默认为1
replicationNum	Integer	否	RequestBody参数	副本数量。默认值为1。取值范围1-10。 Redis内存型和Redis容量型时有效。 Memcache时无效。
instanceNum	Integer	否	RequestBody参数	购买数量，默认为1
diskType	String	否	RequestBody参数	存储磁盘类型，购买Reids 容量型(原 PegaDB)时有效。默认值为cloud_hp1。
diskFlavor	Integer	否	RequestBody参数	单分片存储磁盘容量大小，购买Reids 容量型(原 PegaDB)时为必填项。
chargeType	String	否	RequestBody参数	计费类型。默认为按量付费postpay。取值范围如下： 包年包月：prepay。 按量付费：postpay。
period	Integer	否	RequestBody参数	计费周期。计费类型为包年包月时必填，单位为月，默认为1
timeUnit	String	否	RequestBody参数	计费单位。包年包月时默认值为MONTH；按量付费时默认值为MINUTE。取值范围如下，区分大小写： 年：YEAR 月：MONTH 日：DAY 小时：HOUR。仅按量付费支持此项。 分钟：MINUTE。仅按量付费支持此项。

## 返回头域

除公共头域外，无其它特殊头域。

## 返回参数

参数名称	类型	描述
price	Float	实例价格。包年包月计费单位为元/月，按量付费计费单位为元/分钟。 如果设置了timeUnit,则是以timeUnit为单位的价格。
catalogPrice	Float	目录价。单位同price一致。

请求示例

```
POST http://redis.bj.baidubce.com/v1/price
HOST: redis.bj.baidubce.com
ContentType: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "nodeType": "pega.g4s1.micro",
 "clusterType": "cluster",
 "engine": 3,
 "shardNum": 2,
 "replicationNum": 2,
 "instanceNum": 1,
 "diskFlavor": 50,
 "chargeType": "postpay",
 "period": 1,
 "timeUnit": "YEAR"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 150
{
 "price": 2558,
 "catalogPrice": 2558
}
```

获取实例变更配置价格

请求说明

- 本接口用于获取已有Redis实例变更为指定配置的价格,预付费计费单位为元/月,后付费计费单位为元/分钟。
- 当前版本为V1。
- Memcache仅支持变更分片，不支持变更规格。

请求结构

```
POST /v1/instance/{instanceId}/price
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "chargeType": "postpay",
 "nodeType": "pega.g4s1.micro",
 "shardNum": 2,
 "replicationNum": 2,
 "diskFlavor": 50
}
```

请求参数

参数	类型	是否必须	参数位置	描述
clientToken	String	是	URL参数	幂等性Token，是一个长度不超过64位的ASCII字符串
instanceId	String	是	URL参数	实例Id
nodeType	String	否	RequestBody参数	节点规格，Redis内存型和Redis容量型此字段有效。 具体规格通过 <a href="#">获取实例规格列表</a> 接口获取。
shardNum	Integer	否	RequestBody参数	分片个数，默认为1
replicationNum	Integer	否	RequestBody参数	副本数量，默认为1。 Redis内存型和Redis容量型此字段有效。
diskFlavor	Integer	否	RequestBody参数	单分片容量，单位GB。 Redis容量型此字段有效。
chargeType	String	否	RequestBody参数	计费类型。预付费prepay/后付费postpay,计费类型需与原实例相同
period	Integer	否	RequestBody参数	计费周期。计费类型为预付费必填,单位为月
changeType	String	否	RequestBody参数	变更类型。 modifyType：Redis标准版变配为集群版时的询价标识，此时为必填字段。

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
price	Float	实例价格。预付费计费单位为元/月,后付费计费单位为元/分钟

请求示例

```
POST http://redis.bj.baidubce.com/v1/instance/scs-bj-jrmyipizxoeq/price
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "chargeType": "postpay",
 "nodeType": "pega.g4s1.micro",
 "shardNum": 2,
 "replicationNum": 2,
 "diskFlavor": 50
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 150
{
 "price": 0.1064375
}
```

## 后付费转预付费API

### 请求说明

通过调用API接口来执行后付费转预付费。

- 后付费转预付费立即生效。
- 默认自动支付，请保证账户有足够余额。

### 请求结构

```
POST /v1/instance/toPrepay HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "duration":1,
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
duration	Integer	Body	是	转预付费的时长，单位月。最小值1。
instanceIds	List<String>	Body	是	集群ID列表。

### 返回头域

除公共头部，无其它特殊头部。

### 返回参数

参数名称	参数类型	描述
orderId	String	订单ID。

### 请求示例



```
POST /v1/instance/toPrepay HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "duration":1,
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "orderId": "xxx"
}
```

预付费转后付费API

请求说明

通过调用API接口来执行预付费转后付费。

- 预付费转后付费延迟生效。仅在预付费实例到期后自动变更为后付费，生成转换任务的实例不支持进行其他订单相关操作（比如变配、续费等）。
- 仅预付费实例支持变更
- 处于待变更状态的实例不支持重复变更
- 开通自动续费的实例不支持计费变更
- 已到期的实例不支持计费变更

请求结构

```
POST /v1/instance/toPostpay HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceIds	List<String>	Body	是	集群ID列表。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
orderId	String	订单ID。多个集群同时变更会产生多个订单，订单以英文逗号分隔。

请求示例

```
POST /v1/instance/toPostpay HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "orderId": "xxx"
}
```

取消预付费转后付费API

请求说明

通过调用API接口来取消计费变更。

- 只有计费变更没有生效之前可以取消。

请求结构

```
POST /v1/instance/cancelToPostpay HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceIds	List<String>	Body	是	集群ID列表。

返回头域

除公共头部，无其它特殊头部。

#### 返回参数

无。

#### 请求示例

```
POST /v1/instance/cancelToPostpay HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

## 续费API

## 请求说明

对于预付费实例，通过调用API接口来执行续费操作。

- 本接口用于对已有的预付费实例进行续费。
- 如果预付费实例在回收站中,续费后会从回收站中恢复。
- 续费接口为异步变更，可通过查询指定实例详情接口查询实例状态。
- 当前版本为v1。

#### 请求结构

```
POST /v1/instance/renew HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "duration":1,
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

#### 请求头域

除公共头域外，无其它特殊头域。

#### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceIds	List<String>	Body	是	集群ID列表。
duration	Integer	Body	是	转预付费的时长，单位月。最小值1。

🔗 返回头域

除公共头部，无其它特殊头部。

🔗 返回参数

参数名称	参数类型	描述
orderId	String	订单ID。

🔗 请求示例

```
PUT /v1/instance/scs-bj-ozgdsbsqqnxi/clusterTypeChange HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "duration":1,
 "instanceIds": [
 "scs-bj-woxuukeunrju",
 "scs-bj-rfdlbypsafhi"
]
}
```

🔗 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "orderId": "xxx"
}
```

## 实例服务

### 获取实例列表

请求说明

- 只能查看属于自己账号的实例列表。
- 当前版本为v2。

请求结构

```
GET /v2/instance?marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
marker	String	否	URL参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	String	否	URL参数	每页包含的最大数量(主实例)，最大数量通常不超过1000，缺省值为1000
instanceIds	String	否	URL参数	集群ID查询。支持多个实例ID，以英文逗号分隔。
vnetIp	String	否	URL参数	内网IP查询。

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	类型	描述
marker	String	批量获取列表的查询的起始位置，是一个由系统生成的字符串,起始值可传入-1
maxKeys	integer	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000
isTruncated	boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现
instances	List<InstanceModel>	由InstanceModel组成的数组

请求示例

```
GET /v2/instance?marker=-1&maxKeys=20 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:57:04Z/1800/host/633fd6aad177fce91e5e930c63886b5fb4da8938d9a02c9d13df4fdb8efdec5
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "marker": "-1",
 "isTruncated": true,
 "nextMarker": "1058",
 "maxKeys": 1,
 "instances": [
 {
 "instanceId": "scs-bj-cxisuflkquj",
 "instanceName": "post101",
 "instanceStatus": "Running",
 "engine": "redis",
 "engineVersion": "3.2",
 "vnetIp": "10.107.231.11",
 "domain": "redis.ruqpwjmvrsyt.scs.bj.bce.qa.sandbox.baidu.com",
 "port": "6379",
 "instanceCreateTime": "2018-11-13T05:37:49Z",
 "capacity": 1,
 "usedCapacity": 0.06,
 "paymentTiming": "Postpaid",
 "zoneNames": [
 "cn-bj-a",
 "cn-bj-c"
]
 }
]
}
```

获取实例详情

请求说明

- 查询单个实例的详细信息。
- 当前版本为v2。

请求结构

```
GET /v2/instance/{instanceId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceId	String	是	URL参数	待查询的实例ID

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	类型	描述
instanceId	String	实例ID
instanceName	String	实例名称
instanceStatus	String	<a href="#">实例状态</a>
clusterType	String	引擎类型为Redis，取值： 企业版集群: “cluster” 主从: “master_slave”
engine	String	引擎类型，redis,memcache,Redis 容量型(原 PegaDB)
engineVersion	String	引擎版本
nodeType	String	<a href="#">节点规格</a>
storeType	int	存储类型 0:高性能存储,1:分级存储
shardNum	int	分片数量
vnetIp	String	私网IP
domain	String	域名
port	String	内网端口
instanceCreateTime	Datetime	实例创建时间
instanceExpireTime	Datetime	实例到期时间
capacity	Float	总容量，单位GB
usedCapacity	Double	已用容量，单位GB
paymentTiming	String	付费方式 包年包月：Prepaid，按量付费：Postpaid
zoneNames	List<String>	可用区list
replicationNum	int	副本数
replicationInfo	List < <a href="#">ReplicationMap</a> >	副本信息
diskFlavor	int	存储容量
enableReadOnly	int	副本只读开关
vpId	String	VPC的ID
subnets	List<Subnet>	子网列表
autoRenew	Boolean	是否自动续费，是：“true”，否：“false”
entrance	String	统一读入口地址。开启统一读入口后生效。

Subnet

参数名称	参数类型	参数描述
name	String	子网名
subnetId	String	子网Id
zoneName	String	子网所在可用区
cidr	String	子网cidr
vpId	String	所属vpc的id

ReplicationMap

参数名称	参数类型	参数描述	
availability Zone	String	可通过调用获取可用区列表接口查询可用区列表。zoneName命名规范是"国家-region-可用区序列"，小写，例如北京可用区A为"cn-bj-a"	
subnetId	String	子网Id	
isMaster	Int	是否是主节点 1代表主 0代表从	

请求示例

```
GET /v2/instance/scs-bj-cxisuftlkquj HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
```

返回示例



```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "engine": "redis",
 "instanceStatus": "Running",
 "domain": "redis.qftampylobsp.scs.bj.bce.qa.sandbox.baidu.com",
 "vpcId": "vpc-j1vaxw1cx2mw",
 "nodeType": "cache.n1.medium",
 "subnets": [
 {
 "subnetId": "sbn-hvq6kecs1jj6",
 "cidr": "192.168.0.0/20",
 "name": "系统预定义子网",
 "zoneName": "cn-bj-a"
 }
],
 "instanceId": "scs-bj-zpkrihnhwtca",
 "instanceName": "redis-khbr3qqr",
 "clusterType": "master_slave",
 "paymentTiming": "Postpaid",
 "instanceNum": 1,
 "vnetIp": "192.168.15.87",
 "instanceCreateTime": "2020-11-10T11:31:42Z",
 "engineVersion": "6.0",
 "capacity": 4,
 "storeType": 0,
 "zoneNames": [
 "cn-bj-a"
],
 "replicationNum": 2,
 "replicationInfo": [{
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-rvv87cdd0gv9",
 "isMaster": 1
 }],
 "port": 6379,
 "usedCapacity": 0
}
```

修改实例名称

请求说明

- 修改实例名称。

请求结构

```
PUT /v1/instance/{instanceId}/rename?clientToken={clientToken} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "instanceName": "instanceName"
}
```

请求参数 | 参数 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----|-----|-----|

-----| | clientToken | String | 是 | Query参数 | 幂等性Token，是一个长度不超过64位的ASCII字符串。 | | instanceId | String | 是 | URL参数 | 实例ID | | instanceName | String | 是 | RequestBody参数 | 新的实例名称。要求：

1. 支持大小写字母、数字以及\_ / .等特殊字符，必须以字母开头；

- 2. 长度限制为1-64 ；
- 3. 和自己其他实例的名称不重复 |

返回头域 除公共头域外，无其它特殊头域。

返回参数 无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/rename?
clientToken=CKi1yQEIirbJAQiktskBCKmdygEI74fLAQiWocsBCIagzQEIucrNAQjQy80BCJLPzQEY+MnNAQ HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-
21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "instanceName":"instanceName"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

修改实例域名

请求说明

- 修改实例域名。

请求结构

```
PUT /v1/instance/{instanceId}/renameDomain?clientToken={clientToken} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "domain":"domain"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数 | 参数 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----|-----| |
clientToken | String | 是 | URL参数 | 幂等性Token，是一个长度不超过64位的ASCII字符串。 | | instanceId | String | 是 | URL参
数 | 实例ID | | domain | String | 是 | RequestBody参数 | 新的实例域名。域名命名规范：
1.由小写字母和数字组成；
2.以小写字母开头；
3.长度在3-30之间。 |

返回头域

除公共头域外，无其它特殊头域。

## 返回参数

无特殊返回参数。

## 请求示例

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/renameDomain HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "domain":"domain"
}
```

## 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

## 重启实例API

### 请求说明

- 重启单个redis实例,不支持重启memcache。
- 重启后Redis实例状态变更为重启中，重启成功后变更为运行中。
- 重启过程中可能会有分片数\*5秒内的闪断，请选择业务低峰期进行，并确保应用具有自动重连机制。
- 注：对于单副本实例，实例重启将导致数据丢失，请谨慎操作。

### 请求结构

```
PUT /v1/instance/{instanceId}/restart HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

**请求参数** | 参数 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----|-----| |  
instanceId | String | 是 | Path | 实例ID | | isDefer | Boolean | 否 | Body | 执行时机。

- false：立即执行
- true：维护时间内执行 |

### 返回头域

除公共头域外，无其它特殊头域。

### 返回参数

无特殊返回参数。 **请求示例**

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/restart HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

清空实例API

请求说明

- 清空单个Redis实例的数据。
- 对Redis实例，清空后表现为数据占用内存下降，数据被清空。

请求结构

```
PUT /v1/instance/{instanceId}/flush HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "password":"password",
 "dbIndex": 0
}
```

请求头域

除公共头域外，无特殊其他头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
password	String	Body	是	实例密码，如果没有设置密码，传递空字符串，密码需要加密传输，禁止明文传输，详情请参考 <a href="#">密码加密传输规范定义</a>
dbIndex	Integer	Body	否	数据库索引，取值为：[0,255]
isFlushExpired	Boolean	Body	否	true=只清理过期数据, false=清理所有数据
isDefer	Boolean	Body	否	维护时间窗口执行, 仅isFlushExpired=true时可用。     • true：维护时间执行，false：立即执行。

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-ykfzgtgpxjny/flush HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "password":"password",
 "dbIndex": 0
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

获取实例规格列表

请求说明

获取实例规格列表。

请求结构

```
GET /v2/nodetypes HTTP/1.1
HOST: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

无

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
defaultNodeTypeList	List<NodeTypeItem>	Redis内存型标准版规格
clusterNodeTypeList	List<NodeTypeItem>	Redis内存型集群版规格
pegaClusterNodeTypeList	List<NodeTypeItem>	Redis容量型（PegaDB）集群版规格

NodeTypeItem Object详情

参数名称	参数类型	参数描述
nodeType	String	节点规格
instanceFlavor	Integer	节点容量
cpuNum	Integer	cpu数量
networkThroughputInGbps	float	网络吞吐
peakQps	Integer	参考最大Qps
maxConnections	Integer	最大连接数
allowedNodeNumList	List<int>	允许的分片数量
minDiskFlavor	Integer	Pegadb规格最小的磁盘大小，单位GB。
maxDiskFlavor	Integer	Pegadb规格最小的磁盘大小，单位GB。

请求示例

```
GET /v2/nodetypes HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T06:13:25Z/1800/host/2feac87a586a66c048be0b6ef2d02dfca7d65cf9b6c9e6b9efe20bac95b7c7a
```

返回示例

```
HTTP/1.1 200 OK
```

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 150
{
"clusterNodeTypeList": [
{
"instanceFlavor": 32.0,
"nodeType": "cache.n1.2xlarge",
"allowedNodeNumList": [
1,
2,
3,
4,
5,
6,
8,
12,
16,
24,
32,
40,
48,
64,
80,
96,
128,
160,
192,
256
],
"cpuNum": 2,
"networkThroughputInMB": 48,
"networkThroughputInGbps": 0.048,
"peakQps": 60000,
"maxConnections": 60000,
"allowedReplicationNumList": [
1,
2,
3,
4,
5,
6,
7,
8,
9,
10
],
"minDiskFlavor": 0,
"maxDiskFlavor": 0,
"defaultDiskFlavor": 0
}
],
"defaultNodeTypeList": [
{
"instanceFlavor": 0.25,
"nodeType": "cache.n1.nano",
"allowedNodeNumList": [
1
],
"cpuNum": 2,
"networkThroughputInMB": 10,
"networkThroughputInGbps": 0.01
```

```
 "networkThroughputInMbps": 0.01,
 "peakQps": 60000,
 "maxConnections": 10000,
 "allowedReplicationNumList": [
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10
],
 "minDiskFlavor": 0,
 "maxDiskFlavor": 0,
 "defaultDiskFlavor": 0
 },
 {
 "instanceFlavor": 1.0,
 "nodeType": "cache.n1.micro",
 "allowedNodeNumList": [
 1
],
 "cpuNum": 2,
 "networkThroughputInMB": 10,
 "networkThroughputInGbps": 0.01,
 "peakQps": 60000,
 "maxConnections": 10000,
 "allowedReplicationNumList": [
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10
],
 "minDiskFlavor": 0,
 "maxDiskFlavor": 0,
 "defaultDiskFlavor": 0
 }
],
"pegaClusterNodeTypeList": [
 {
 "instanceFlavor": 8.0,
 "nodeType": "pega.g4s1.micro",
 "allowedNodeNumList": [
 1,
 2,
 4,
 6,
 8,
 12,
 16,
 24,
 32,
 48,
 64,
 80,
 96,
 112,
 128,
 144,
 160,
 176,
 192,
 208,
 224,
 240,
 256,
 272,
 288,
 304,
 320,
 336,
 352,
 368,
 384,
 400,
 416,
 432,
 448,
 464,
 480,
 496,
 512,
 528,
 544,
 560,
 576,
 592,
 608,
 624,
 640,
 656,
 672,
 688,
 704,
 720,
 736,
 752,
 768,
 784,
 800,
 816,
 832,
 848,
 864,
 880,
 896,
 912,
 928,
 944,
 960,
 976,
 992,
 1008,
 1024,
 1040,
 1056,
 1072,
 1088,
 1104,
 1120,
 1136,
 1152,
 1168,
 1184,
 1200,
 1216,
 1232,
 1248,
 1264,
 1280,
 1296,
 1312,
 1328,
 1344,
 1360,
 1376,
 1392,
 1408,
 1424,
 1440,
 1456,
 1472,
 1488,
 1504,
 1520,
 1536,
 1552,
 1568,
 1584,
 1600,
 1616,
 1632,
 1648,
 1664,
 1680,
 1696,
 1712,
 1728,
 1744,
 1760,
 1776,
 1792,
 1808,
 1824,
 1840,
 1856,
 1872,
 1888,
 1904,
 1920,
 1936,
 1952,
 1968,
 1984,
 2000,
 2016,
 2032,
 2048,
 2064,
 2080,
 2096,
 2112,
 2128,
 2144,
 2160,
 2176,
 2192,
 2208,
 2224,
 2240,
 2256,
 2272,
 2288,
 2304,
 2320,
 2336,
 2352,
 2368,
 2384,
 2400,
 2416,
 2432,
 2448,
 2464,
 2480,
 2496,
 2512,
 2528,
 2544,
 2560,
 2576,
 2592,
 2608,
 2624,
 2640,
 2656,
 2672,
 2688,
 2704,
 2720,
 2736,
 2752,
 2768,
 2784,
 2800,
 2816,
 2832,
 2848,
 2864,
 2880,
 2896,
 2912,
 2928,
 2944,
 2960,
 2976,
 2992,
 3008,
 3024,
 3040,
 3056,
 3072,
 3088,
 3104,
 3120,
 3136,
 3152,
 3168,
 3184,
 3200,
 3216,
 3232,
 3248,
 3264,
 3280,
 3296,
 3312,
 3328,
 3344,
 3360,
 3376,
 3392,
 3408,
 3424,
 3440,
 3456,
 3472,
 3488,
 3504,
 3520,
 3536,
 3552,
 3568,
 3584,
 3600,
 3616,
 3632,
 3648,
 3664,
 3680,
 3696,
 3712,
 3728,
 3744,
 3760,
 3776,
 3792,
 3808,
 3824,
 3840,
 3856,
 3872,
 3888,
 3904,
 3920,
 3936,
 3952,
 3968,
 3984,
 4000,
 4016,
 4032,
 4048,
 4064,
 4080,
 4096,
 4112,
 4128,
 4144,
 4160,
 4176,
 4192,
 4208,
 4224,
 4240,
 4256,
 4272,
 4288,
 4304,
 4320,
 4336,
 4352,
 4368,
 4384,
 4400,
 4416,
 4432,
 4448,
 4464,
 4480,
 4496,
 4512,
 4528,
 4544,
 4560,
 4576,
 4592,
 4608,
 4624,
 4640,
 4656,
 4672,
 4688,
 4704,
 4720,
 4736,
 4752,
 4768,
 4784,
 4800,
 4816,
 4832,
 4848,
 4864,
 4880,
 4896,
 4912,
 4928,
 4944,
 4960,
 4976,
 4992,
 5008,
 5024,
 5040,
 5056,
 5072,
 5088,
 5104,
 5120,
 5136,
 5152,
 5168,
 5184,
 5200,
 5216,
 5232,
 5248,
 5264,
 5280,
 5296,
 5312,
 5328,
 5344,
 5360,
 5376,
 5392,
 5408,
 5424,
 5440,
 5456,
 5472,
 5488,
 5504,
 5520,
 5536,
 5552,
 5568,
 5584,
 5600,
 5616,
 5632,
 5648,
 5664,
 5680,
 5696,
 5712,
 5728,
 5744,
 5760,
 5776,
 5792,
 5808,
 5824,
 5840,
 5856,
 5872,
 5888,
 5904,
 5920,
 5936,
 5952,
 5968,
 5984,
 6000,
 6016,
 6032,
 6048,
 6064,
 6080,
 6096,
 6112,
 6128,
 6144,
 6160,
 6176,
 6192,
 6208,
 6224,
 6240,
 6256,
 6272,
 6288,
 6304,
 6320,
 6336,
 6352,
 6368,
 6384,
 6400,
 6416,
 6432,
 6448,
 6464,
 6480,
 6496,
 6512,
 6528,
 6544,
 6560,
 6576,
 6592,
 6608,
 6624,
 6640,
 6656,
 6672,
 6688,
 6704,
 6720,
 6736,
 6752,
 6768,
 6784,
 6800,
 6816,
 6832,
 6848,
 6864,
 6880,
 6896,
 6912,
 6928,
 6944,
 6960,
 6976,
 6992,
 7008,
 7024,
 7040,
 7056,
 7072,
 7088,
 7104,
 7120,
 7136,
 7152,
 7168,
 7184,
 7200,
 7216,
 7232,
 7248,
 7264,
 7280,
 7296,
 7312,
 7328,
 7344,
 7360,
 7376,
 7392,
 7408,
 7424,
 7440,
 7456,
 7472,
 7488,
 7504,
 7520,
 7536,
 7552,
 7568,
 7584,
 7600,
 7616,
 7632,
 7648,
 7664,
 7680,
 7696,
 7712,
 7728,
 7744,
 7760,
 7776,
 7792,
 7808,
 7824,
 7840,
 7856,
 7872,
 7888,
 7904,
 7920,
 7936,
 7952,
 7968,
 7984,
 8000,
 8016,
 8032,
 8048,
 8064,
 8080,
 8096,
 8112,
 8128,
 8144,
 8160,
 8176,
 8192,
 8208,
 8224,
 8240,
 8256,
 8272,
 8288,
 8304,
 8320,
 8336,
 8352,
 8368,
 8384,
 8400,
 8416,
 8432,
 8448,
 8464,
 8480,
 8496,
 8512,
 8528,
 8544,
 8560,
 8576,
 8592,
 8608,
 8624,
 8640,
 8656,
 8672,
 8688,
 8704,
 8720,
 8736,
 8752,
 8768,
 8784,
 8800,
 8816,
 8832,
 8848,
 8864,
 8880,
 8896,
 8912,
 8928,
 8944,
 8960,
 8976,
 8992,
 9008,
 9024,
 9040,
 9056,
 9072,
 9088,
 9104,
 9120,
 9136,
 9152,
 9168,
 9184,
 9200,
 9216,
 9232,
 9248,
 9264,
 9280,
 9296,
 9312,
 9328,
 9344,
 9360,
 9376,
 9392,
 9408,
 9424,
 9440,
 9456,
 9472,
 9488,
 9504,
 9520,
 9536,
 9552,
 9568,
 9584,
 9600,
 9616,
 9632,
 9648,
 9664,
 9680,
 9696,
 9712,
 9728,
 9744,
 9760,
 9776,
 9792,
 9808,
 9824,
 9840,
 9856,
 9872,
 9888,
 9904,
 9920,
 9936,
 9952,
 9968,
 9984,
 10000
]
 }
]
```

```
96,
128,
160,
192,
256,
320
],
"cpuNum": 4,
"networkThroughputInMB": 187,
"networkThroughputInGbps": 0.187,
"peakQps": 10000,
"maxConnections": 30000,
"allowedReplicationNumList": [
1,
2,
3,
4,
5,
6,
7,
8,
9,
10
],
"minDiskFlavor": 50,
"maxDiskFlavor": 320,
"defaultDiskFlavor": 64
}
]
}
```

获取可用区列表

请求说明

获取可用区列表。

请求结构

```
GET /v1/zone HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

无

返回参数

参数名称	类型	描述
zones	List <ZoneNames>	可用区列表

ZoneNames

参数名称	类型	描述
zoneNames	List <String>	可用区列表

请求示例



```
GET /v1/zone HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T06:05:41Z/1800/host/7767a07238b64a846fef59a0b919eec99d147e5fe23be98ff6bf90e32e889baf
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "zones": [
 {"zoneNames":["cn-bj-a"]},
 {"zoneNames":["cn-bj-b"]},
 {"zoneNames":["cn-bj-a","cn-bj-b"]}
]
}
```

获取子网列表

请求说明

获取子网列表。

请求结构

```
GET /v1/subnet?vpclId={vpclId}&zoneName={zoneName} HTTP/1.1
HOST: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
vpclId	String	否	Query参数	所属vpc的id
zoneName	String	否	Query参数	所属可用区的名称

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	类型	描述
subnets	List <Subnet>	子网列表

Subnet Object详情

参数名称	参数类型	参数描述
name	String	子网名
subnetId	String	子网Id
zoneName	String	子网所在可用区
cidr	String	子网cidr
vpclId	String	所属vpc的id

请求示例

```
GET /v1/subnet?vpclId=vpc-1n1wqxfu4iuu&zoneName=cn-bj-a HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T06:13:25Z/1800/host/2feac87a586a66c048be0b6ef2d02dfca7d65cf9b6c9e6b9efe20bac95b7c7a
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 150
{
 "subnets": [
 {
 "name": "系统预定义子网B",
 "subnetId": "sbn-4zt8126ea6c3",
 "cidr": "192.168.16.0/20",
 "zoneName": "cn-bj-b",
 "vpclId": "vpc-1n1wqxfu4iuu"
 }
]
}
```

绑定标签

请求说明

- 给实例绑定标签。

请求结构

```
PUT /v1/instance/{instanceId}/bindTag HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必需	参数位置	描述
instanceId	String	是	RequestBody参数	虚拟机实例ID
changeTags	List<TagModel>	是	RequestBody参数	待解绑的标签列表

返回头域

除公共头域，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-zpkrihnhtwca/bindTag HTTP/1.1
Host: redis.cd.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2023-08-13T07:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 1214cca7-4ad5-451d-9215-71cb844c0a50
Content-Type: application/json;charset=UTF-8
```

解绑标签

请求说明

- 解绑实例上定义的标签。
- 可以同时解绑多个标签。

请求结构

```
PUT /v1/instance/{instanceId}/unBindTag HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数 | 参数名称 | 类型 | 是否必需 | 参数位置 | 描述 | |-----|-----|-----|-----| | instanceId | String | 是 | RequestBody参数 | 虚拟机实例ID | | changeTags | List | 是 | RequestBody参数 | 待解绑的标签列表，具体数据格式参见下述TagModel |

返回头域

除公共头域，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-cxisuftlkquj/unBindTag HTTP/1.1
Host: redis.cd.baidubce.com
Content-Type: application/json
Authorization: authorization string
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 1214cca7-4ad5-451d-9215-71cb844c0a50
Content-Type: application/json;charset=UTF-8
```

从回收站中批量恢复实例

请求说明

- 使用以下代码可以从回收站中批量恢复实例,批量恢复仅支持后付费实例。
- 回收站中的预付费实例请通过续费实例接口恢复。
- 当前版本为V2。

请求结构

```
POST /v2/recycler/recover HTTP/1.1
HOST: redis.bj.baidubce.com
Authorization: authorization string
{
 "instanceIds" : ["scs-bj-arkfmjcblyty"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceIds	List	是	RequestBody	实例Id列表

返回头域

除公共头域，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/recycler/recover HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
{
 "instanceIds" : ["scs-bj-arkfmjcblyty"]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

错误码

错误码	错误描述	HTTP状态码	中文解释
BadRequest	Bad request parameters or illegal request	400	请求有误
InstanceNotExistException	Instance not found	404	实例未找到或非自己的实例
ServiceUnavailable	Service is unavailable	503	内部服务器错误

释放回收站中的实例

请求说明

- 从回收站中批量释放Redis实例，释放后实例所使用的物理资源都被收回，相关数据全部丢失且不可恢复。
- 当前版本为V2。

请求结构

```
POST /v2/recycler/delete HTTP/1.1
HOST: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceIds	List<String>	是	RequestBody	实例Id列表

返回头域

除公共头域，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/recycler/delete HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2021-04-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "instanceIds" : ["scs-bj-arkfmjcblyty"]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

错误码

错误码	错误描述	HTTP状态码	中文解释
BadRequest	Bad request parameters or illegal request	400	请求有误
InstanceNotExistException	Instance not found	404	实例未找到或非自己的实例
InstanceDeleteFailed	Instance {instanceId} delete failed	403	删除scs实例失败
ServiceUnavailable	Service is unavailable	503	内部服务器错误

获取回收站内的实例列表

请求说明

- 获取当前回收站内的实例列表。
- 当前版本为V2。

请求结构

```
GET /v2/recycler/list?marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
marker	String	否	Query参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串,初始值可传入-1
maxKeys	int	否	Query参数	每页包含的最大数量(主实例)，最大数量通常不超过1000，缺省值为1000。

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	类型	描述
marker	String	批量获取列表的查询的起始位置，是一个由系统生成的字符串,起始值可传入-1
maxKeys	int	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000。
isTruncated	boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现
instances	List<InstanceModel>	由InstanceModel组成的数组

请求示例

```
GET /v2/instance?marker=-1&maxKeys=20 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:57:04Z/1800/host/633fd6aad177fce91e5e930c63886b5fb4da8938d9a02c9d13df4fdb8efdec5
```

返回示例

```
{
 "marker": "-1",
 "isTruncated": true,
 "nextMarker": "1058",
 "maxKeys": 1,
 "instances": [
 {
 "instanceId": "scs-bj-cxisuftlkquj",
 "instanceName": "post101",
 "instanceStatus": "Running",
 "isolatedStatus": "Isolated",
 "engine": "redis",
 "engineVersion": "3.2",
 "vnetIp": "10.107.231.11",
 "domain": "redis.ruqpwmvrsyt.scs.bj.bce.qa.sandbox.baidu.com",
 "port": "6379",
 "instanceCreateTime": "2018-11-13T05:37:49Z",
 "capacity": 1,
 "usedCapacity": 0.06,
 "paymentTiming": "Postpaid",
 "zoneNames": [
 "cn-bj-a",
 "cn-bj-c"
]
 }
]
}
```

集群状态检查

🔗 集群状态检查

接口描述

本接口用于检查实例所在集群的状态。

本接口为v1接口。

注意事项

本接口仅支持 Redis 和 Redis 容量型(原 PegaDB)引擎的实例，不支持 Memcache 引擎实例。

请求结构

```
GET /v{version}/instance/{instanceId}/status HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为1
instanceId	String	是	PATH参数	要检查的实例ID。

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
clusterStatus	String	用于描述集群状态，有两种取值。normal 集群状态正常；abnormal 集群状态不正常。
checkList	CheckList	检查项，每项中：Yes表示检查通过，No表示检查不通过。 所有项检查通过，clusterStatus为normal； 有任意一项检查不通过，clusterStatus为abnormal。

CheckList

参数名称	类型	描述
clusterInstanceStatus	String	集群实例状态是否正常。Yes表示检查通过，No表示检查不通过。
clusterTopology	String	集群拓扑是否正常。Yes表示检查通过，No表示检查不通过。
clusterRedisIsAlived	String	集群实例是否都存活。Yes表示检查通过，No表示检查不通过。
clusterDelay	String	集群是否存在延迟。Yes表示检查通过，No表示检查不通过。
leaderReadOnly	String	主节点只读检查是否通过。Yes表示检查通过，No表示检查不通过。

请求示例

```
GET http://redis.bj.baidubce.com/v1/instance/scs-bj-ybklwexvscdd/status
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "clusterStatus": "normal",
 "checkList": {
 "clusterInstanceStatus": "Yes",
 "clusterTopology": "Yes",
 "clusterRedisIsAlived": "Yes",
 "clusterDelay": "Yes",
 "leaderReadOnly": "Yes"
 }
}
```

统一读入口负载均衡

🔗 统一读入口负载均衡

接口描述

本接口用于创建统一读入口后进行重新负载均衡操作。

本接口为v2接口。

注意事项

请求结构

```
POST /v{version}/entrance/disconnect/{instanceId} HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	要操作的集群ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
success	Boolean	是否操作成功
result	Boolean	是否操作成功。和success一致。

请求示例



```
POST http://redis.bj.baidubce.com/v2/entrance/disconnect/scs-bj-udufxwaeltv
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true,
 "result": true
}
```

创建统一读入口

创建统一读入口

接口描述

本接口用于创建集群的统一读入口。不可重复创建。

本接口为v2接口。

创建完成后可通过查询集群详情接口查看统一读入口地址。

**注意事项**

本接口仅支持Redis社区标准版实例，并且集群存在可用的只读节点。

请求结构

```
POST /v{version}/entrance/create/{instanceId} HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	要操作的集群ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
success	Boolean	是否创建成功
result	Boolean	是否创建成功，和success一致

请求示例

```
POST http://redis.bj.baidubce.com/v2/entrance/create/scs-bj-udufxfwaeltv
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true,
 "result": true
}
```

拨测接口



接口描述

本接口用于拨测。

本接口为v1接口。

注意事项

不支持标准版单副本实例。

不支持Memcache实例。

请求结构

```
GET /v{version}/instance/{instanceId}/blbStatus HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为1。
instanceId	String	是	PATH参数	实例ID

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
alive	Boolean	实例blb状态。
errors	String	报错信息。

请求示例

```
GET http://redis.bj.baidubce.com/v1/instance/scs-bj-hohcysprhst/blbStatus
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "alive": true,
 "errors": ''
}
```

主从切换API

请求说明

主从切换，变更分片的主从节点关系，保障实例的高可用性。

请求结构

```
PUT /v1/instance/{instanceId}/switchMasterSlave HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "shards": [
 {
 "hashName": "",
 "nodeShowId": ""
 }
]
}
```

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
shards	List<SwitchMasterSlaveShard>	Body	是	切换的分片列表。

SwitchMasterSlaveShard参数信息

参数名称	参数类型	是否必须	描述
hashName	String	是	指定要切换的分片。可从实例详情接口RedisList字段中获取分片信息。
nodeShowId	String	是	指定该分片中要切换为主节点ID。可从实例详情接口RedisList字段中获取分片中节点ID的信息。

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-elahimuhxjia/switchMasterSlave HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "shards": [
 {
 "hashName": "scs-bj-elahimuhxjia-0",
 "nodeShowId": "scs-bj-elahimuhxjia_redis_67957_1"
 }
]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

修改维护时间窗口API

请求说明

通过调用API接口来修改维护时间窗口。

请求结构

```
PUT /v1/instance/{instanceId}/timeWindow HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
startTime	String	Body	是	开始时间。示例：04:00（即凌晨4点）
duration	Integer	Body	是	持续时间。取值范围：1-8（单位：小时）
period	List<Integer>	Body	是	维护周期。1-6分别代表周一到周六。0代表周日。

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-oqdgngbjfogjw/timeWindow HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "startTime": "04:30",
 "duration": 2,
 "period": [1,2,3,4,5,6,0]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

## 标准版升级集群版API

### 请求说明

通过调用API接口来将标准版实例升级为集群版。

- 不支持跨版本架构升级，例如，Redis 4.0 标准架构不支持升级至 Redis 5.0 集群架构。
- 不支持带了只读实例组的标准版实例升级到集群版，如需要升级需要先删除只读实例组才能升级。
- 热活或多活实例组中的实例不支持升级。
- 不支持单副本标准版升级架构。
- 升级后的总内存容量需要大于等于标准版总内存容量。
- 有维护时间内的任务未完成时不支持升级。
- 非运行中实例不支持升级。
- Redis7.0标准版不支持升级。

#### 请求结构

```
PUT /v1/instance/{instanceId}/clusterTypeChange HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

#### 请求头域

除公共头域外，无其它特殊头域。

#### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
isDefer	Boolean	Body	否	是否维护时间内执行。默认false。true：维护时间内执行；false：立即执行。
nodeType	String	Body	否	节点规格。    - 可以是新的规格，需要是Redis集群版的规格。   - 可以为空，默认为原实例规格，但是需要确保原实例规格为集群版的规格，否则执行失败。
shardNum	Integer	Body	否	分片数量。    - 可以设置新的分片数量；   - 可以为空，默认为原实例分片数。
replicationInfo	List<ReplicationItem>	Body	否	副本信息。    - 不支持减少副本，可以新增。   - 可以设置新的副本信息，需全量的副本信息。   - 可以为空，默认为原实例副本数。

ReplicationItem

参数名称	参数类型	描述
availabilityZone	String	可通过调用获取可用区列表接口查询可用区列表。zoneName命名规范是“国家-region-可用区序列”，小写，例如北京可用区A为“cn-bj-a”
subnetId	String	子网Id
isMaster	Integer	是否是主节点 1代表主 0代表从。主节点有且仅有一个。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
orderId	String	订单ID。

请求示例

```
PUT /v1/instance/scs-bj-ozgdsbsqnx/clusterTypeChange HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "nodeType": "cache.n1.small",
 "shardNum": 2,
 "replicationInfo": [
 {
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-t94a1xpb4tw4",
 "isMaster": 1
 },
 {
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-t94a1xpb4tw4",
 "isMaster": 0
 }
],
 "isDefer": false
}
```

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "orderId": "xxx"
}
```

## 迁移可用区API

#### 请求说明

云数据库Redis支持通过调用API将实例迁移到同地域内的其它可用区、子网。

- 迁移可用区功能不支持**热活实例组**、**多活实例组**、不支持只读节点。
- Redis**社区版**、**集群版**、**容量版**均支持可用区迁移功能。
- 主从节点同一个可用区下的子网需要保持一致。
- **Redis社区版只读节点**不支持迁移，但Redis社区版实例其它主从节点迁移时子网的选择会受到限制，即如果迁移到只读节点所在可用区时，子网信息默认需要与只读节点一致。
- 变更选取的可用区、子网不可与之前完全一致，即选取节点完全没变化时不支持变更。
- **开通公网**的实例不支持可用区迁移。
- 实例有待执行的**维护窗口**任务，不支持可用区迁移。

#### 请求结构

```
POST /v1/instance/{instanceId}/azoneMigration
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "isDefer": "true",
 "replicationInfo": [
 {
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-x94w3r601111",
 "isMaster": 1
 }
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID。
isDefer	Boolean	Body	是	是否维护时间内执行。    - true：维护时间内执行   - false：立即执行
replicationInfo	List<ReplicationItem>	Body	是	副本信息。需全量的副本信息，可以修改可用区和子网。

ReplicationItem

参数名称	参数类型	是否必须	描述
availabilityZone	String	是	可用区。示例：cn-bj-a
subnetId	String	是	子网ID。示例：sbn-x94w3r601111
isMaster	Integer	是	是否是主节点。主节点有且仅有一个。     1：主节点    0：从节点

返回头域

除公共头部，无其它特殊头部。

返回参数

无。

请求示例



```
POST /v1/instance/{instanceId}/azoneMigration
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "isDefer": "true",
 "replicationInfo": [
 {
 "availabilityZone": "cn-bj-a",
 "subnetId": "sbn-x94w3r601111",
 "isMaster": 1
 }
]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

IP变更API

请求说明

云数据库Redis支持通过调用API进行内网IP变更。

- VPC默认不变，可用区、子网默认选择与主节点一致。
- 修改网络地址生效后，会断开旧地址上的所有网络连接（旧地址不可用）。请确保业务使用连接地址（域名）连接实例。并确保迁移后DNS缓存及时刷新，且应用程序具备断线重连机制。
- Redis社区版、集群版、容量版均支持IP变更功能。
- 变更IP不支持热活实例组、多活实例组，只读节点。
- 变更IP不支持开通外网ip。
- BLB专属集群的实例不支持变更内网ip。

请求结构

```
POST /v1/instance/{instanceId}/azoneMigration/modifyEntrance
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "isDefer": "true"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID。
				是否维护时间内执行。
isDefer	Boolean	Body	是	• true：维护时间内执行    • false：立即执行

🔗 返回头域

除公共头部，无其它特殊头部。

🔗 返回参数

无。

🔗 请求示例

```
POST /v1/instance/{instanceId}/azoneMigration/modifyEntrance
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "isDefer": "true"
}
```

🔗 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

版本升级API

请求说明

- 通过调用API升级实例的内核版本。

请求结构

```
POST /v1/instance/{instanceId}/upgradeVersion HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无特殊其他头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
kernelVersion	String	Body	否	升级大版本时的版本号。非大版本升级时空。    - 升级版本要高于现在的版本。    3系标准版可以升级到6系、7系    4系可以升级到6系、7系    5系可以升级到6系、7系    6系可以升到7系
isDefer	Boolean	Body	否	执行时间。    - false：立即执行   - true：维护时间内执行

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v1/instance/scs-bj-ykfztgpxjny/upgradeVersion HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

## 账号管理

### 账号列表



账号列表

接口描述

本接口用于查询账号列表。

注意事项

请求结构

```
POST /v{version}/instance/{instanceId}/aclUserActions HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	集群ID。

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
success	Boolean	请求是否成功。
result	List<ListItem>	账号列表。

**ListItem** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | userName | String | 账号名称。 | | updateStatus | Integer | 账号状态。0：正常可用；1：创建中；2：修改中；4：删除中； | | extra | String | 备注信息。 | | userType | Integer | 账号权限。1：读写；2：只读； |

请求示例

```
GET http://redis.bj.baidubce.com/v2/instance/scs-bj-wqocnchhrqft/aclUserActions
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true,
 "result": [
 {
 "userName": "default",
 "updateStatus": 0,
 "extra": "",
 "userType": 1
 }
]
}
```

修改账号密码

修改账号密码

接口描述

本接口用于修改非default账号的密码。

请求结构

```
POST /v{version}/{instanceId}/aclUserActions/modifyPasswd HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "userName": "default",
 "clientAuth": "fb8862d028d5c17aeb419ef655f6b004"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	集群ID。
userName	String	是	RequestBody参数	要设置的账号名称。
clientAuth	String	是	RequestBody参数	账号密码。详情请参考 <a href="#">密码加密传输规范定义</a>

响应头域

除公共头域外，无其它特殊头域。

响应参数

无。

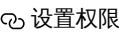
请求示例

```
POST http://redis.bj.baidubce.com/v2/instance/scs-bj-klwyvdooggfp/aclUserActions/modifyPasswd
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "userName": "default",
 "clientAuth": "fb8862d028d5c17aeb419ef655f6b004"
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

设置权限



接口描述

本接口用于设置指定账号的权限。

请求结构

```
POST /v{version}/{instanceId}/aclUserActions/authority HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "userName": "cli_test",
 "userType": 2
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。
instanceId	String	是	PATH参数	集群ID。
userName	String	是	RequestBody参数	要设置的账号名称。
userType	Integer	是	RequestBody参数	该账号设置的权限。1：读写；2：只读；

响应头域

除公共头域外，无其它特殊头域。

响应参数

无。

请求示例

```
POST http://redis.bj.baidubce.com/v2/instance/scs-bj-wqocnchhrqft/aclUserActions/authority
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "userName": "cli000",
 "userType": 2
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

删除账号



删除账号

接口描述

本接口用于删除账号。

请求结构

```
POST /v{version}/{instanceId}/aclUserActions/delete HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "userName": "cli{ttt}"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	集群ID。
userName	String	是	RequestBody参数	要删除的账号名称。

响应头域

除公共头域外，无其它特殊头域。

响应参数

无。

请求示例

```
POST http://redis.bj.baidubce.com/v2/instance/scs-bj-wqocnchhrqft/aclUserActions/delete
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "userName": "cli{ttt}"
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

创建账号

创建账号

接口描述

本接口用于创建集群账号。

请求结构

```
POST /v{version}/instance/{instanceId}/aclUserActions HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "userName": "name",
 "clientAuth": "fb8862d028d5c17aeb419ef655f6b004",
 "extra": "",
 "userType": 2
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
instanceId	String	是	PATH参数	集群ID。
userName	String	是	RequestBody参数	账号名称。由小写字母、数字、下划线组成、字母开头,字母或数字结尾,最长16个字符。
clientAuth	String	是	RequestBody参数	账号密码。详情请参考 <a href="#">密码加密传输规范定义</a>
extra	String	否	RequestBody参数	备注
userType	Integer	是	RequestBody参数	账号权限。1：读写；2：只读；

响应头域

除公共头域外，无其它特殊头域。

响应参数

无。

请求示例

```
POST http://redis.bj.baidubce.com/v2/instance/scs-bj-wqocnchhrqft/aclUserActions
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "userName": "name",
 "clientAuth": "fb8862d028d5c17aeb419ef655f6b004",
 "extra": "",
 "userType": 2
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

参数管理

获取参数列表

请求说明

- 查询Redis实例的配置参数和运行参数。

请求结构

```
GET /v1/instance/{instanceId}/parameter HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceId	String	是	URL参数	待查询的实例ID

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	类型	描述
parameters	List<parameter>	参数列表

parameter参数信息

参数名称	类型	描述
default	String	参数默认值
forceRestart	int	是否需要重启生效：1（重启生效,当前需要提交工单重启）0（无需重启，提交后即生效）
name	String	参数名称
value	String	参数当前值



## 请求示例

```
GET /v1/instance/scs-bj-ykfiztgpxjny/parameter HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

## 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "parameters": [
 {
 "name": "appendonly",
 "default": "yes",
 "value": "no",
 "forceRestart": "0"
 },
 {
 "name": "maxmemory-policy",
 "default": "volatile-ttl",
 "value": "volatile-ttl",
 "forceRestart": "0"
 },
 {
 "name": "hash-max-ziplist-entries",
 "default": "512",
 "value": "512",
 "forceRestart": "0"
 },
 {
 "name": "hash-max-ziplist-value",
 "default": "64",
 "value": "64",
 "forceRestart": "0"
 },
 {
 "name": "set-max-intset-entries",
 "default": "512",
 "value": "512",
 "forceRestart": "0"
 },
 {
 "name": "zset-max-ziplist-entries",
 "default": "128",
 "value": "128",
 "forceRestart": "0"
 },
 {
 "name": "zset-max-ziplist-value",
 "default": "64",
 "value": "64",
 "forceRestart": "0"
 },
 {
 "name": "timeout",
 "default": "0",
 "value": "0",
 "forceRestart": "0"
 }
]
}
```

```
 },
 {
 "name": "hz",
 "default": "10",
 "value": "10",
 "forceRestart": "0"
 },
 {
 "name": "disable_commands",
 "default": "flushall,flushdb",
 "value": "flushall,flushdb",
 "forceRestart": "1"
 },
 {
 "name": "notify-keyspace-events",
 "default": "",
 "value": "",
 "forceRestart": "0"
 }
]
}
```

修改参数

请求说明

- 修改redis实例参数值。

请求结构

```
PUT /v1/instance/{instanceId}/parameter clientToken={clientToken} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "parameter": {
 "name": "name",
 "value": "value"
 }
}
```

请求头域

除公共头域外，无其它特殊头域。

**请求参数** | 参数名称 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----|-----| |  
clientToken | String | 是 | URL参数 | 幂等性Token，是一个长度不超过64位的ASCII字符串 | | instanceId | String | 是 | URL参数 |  
实例Id | | parameter | instanceParam | 是 | RequestBody参数 | 需要修改的参数 |

**instanceParam参数信息** | 参数名称 | 类型 | 是否必须 | 描述 | |-----|-----|-----|-----| | name | String | 是 | 参数名称 | | value |  
String | 是 | 设置参数值 |

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-ykfztgpxjny/parameter HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "parameter": {
 "name": "appendonly",
 "value": "no"
 }
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

备份恢复

查看备份列表API

请求说明

- 查询某个实例备份列表。

请求结构

```
GET /v1/instance/{instanceId}/backup HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

**请求参数** | 参数名称 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----| | instanceId | string | 是 | Path | 集群ID |

返回头域

除公共头部，无其它特殊头部。

**返回参数** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | totalCount | Integer | 总数量。 | | backups | List<BatchBackupRecord> | 结果列表。 |

**BatchBackupRecord 备份信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | batchId | String | 备份ID | | backupType | String | 备份类型

- auto表示自动备份
- manual表示手动产生的备份 | | comment | String | 备注 | | startTime | String | 开始时间。格式：yyyy-MM-dd'T'HH:mm:ss'Z' | | recoverable | String | 是否可用于备份恢复。
- recoverable - 可用
- non-recoverable - 不可用 | | records | List<Record> | 按分片备份的列表。 |

**Record 单个备份记录** | 参数名称 | 参数类型 | 描述 | |-----|-----|-----| | backupRecordId | String | 备份记录ID。 | | backupId | String | 备份记录ID。 | | startTime | String | 开始时间。格式：yyyy-MM-dd'T'HH:mm:ss'Z' | | duration | Long | 持续时间 | | objectSize | Long | 文件大小 |

|backupType|String|备份类型。

- auto 自动备份
- manual 手动备份| |backupStatus|String|备份状态。
- initial 备份中
- success 成功
- doing 备份中
- failed 失败
- expired
- recovering 备份恢复中
- recover failed 备份恢复失败
- recover success 备份恢复成功| |shardName|String|分片名称| |comment|String|备注|

请求示例

```
GET /v1/instance/scs-bj-ykfztgpxjny/backup HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "backups": [
 {
 "backupType": "manual",
 "batchId": "1735873224099928902",
 "batchRecords": [
 {
 "backupId": "4976335",
 "backupRecordId": null,
 "backupStatus": "success",
 "backupType": "manual",
 "comment": "订单11",
 "duration": 1,
 "objectSize": 195,
 "shardName": "Shard-445560",
 "startTime": "2025-01-03T03:00:46Z"
 }
],
 "comment": "订单11",
 "recoverable": "recoverable",
 "startTime": "2025-01-03T03:00:46Z"
 }
],
 "totalCount": 2
}
```

## 获取备份策略API

### 请求说明

- 通过调用API查询某个集群备份策略。

### 请求结构

```
GET /v1/instance/{instanceId}/backup/policy HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID

### 返回头域

除公共头部，无其它特殊头部。

### 返回参数

参数名称	参数类型	描述
backupTime	String	备份时间。
backupDays	String	备份周期。
expireDay	Integer	备份文件保留时长，可选范围： 1-15天
isEncrypt	String	是否开启加密    - no 不开启   - yes 开启

### 请求示例

```
GET /v1/instance/scs-bj-ykfiztgpxjny/backup/policy HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "backupDays": "Mon,Tue,Wed,Thu,Fri,Sta,Sun",
 "backupTime": "0:38:54",
 "expireDay": 7,
 "isEncrypt": "no"
}
```

## 设置备份策略API

请求说明

- 通过调用API来设置redis实例自动备份策略。

请求结构

```
PUT /v1/instance/{instanceId}/backup/modifyPolicy HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "backupDays": "Sun,Mon,Tue,Wed,Thu,Fri,Sta",
 "backupTime": "01:05:00",
 "expireDay": 7,
 "isEncrypt": "yes"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
backupTime	String	Body	否	备份时间。
backupDays	String	Body	是	备份周期。
expireDay	Integer	Body	是	备份文件保留时长，可选范围： 1-15天
isEncrypt	String	Body	是	是否开启加密 no 不开启 yes 开启

返回头域

除公共头部，无其它特殊头部。

返回参数

无。

请求示例

```
PUT /v1/instance/scs-bj-ykfztgpxjny/modifyPolicy HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "backupDays": "Sun,Mon,Tue,Wed,Thu,Fri,Sta",
 "backupTime": "01:05:00",
 "expireDay": 7,
 "isEncrypt": "yes"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

删除手动备份API

请求说明

- 通过调用API来删除手动备份。

请求结构

```
DELETE /v1/instance/{instanceId}/backup/{batchId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
batchId	String	Path	是	备份ID。

返回头域

除公共头部，无其它特殊头部。

返回参数

无。

请求示例

```
DELETE /v1/instance/{instanceId}/backup/{batchId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

获取备份的URL API

请求说明

- 通过调用API来获取备份的URL。

请求结构

```
GET /v1/instance/{instanceId}/backup/{backupId}/url HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
backupId	Integer	Path	是	备份ID。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
url	String	下载地址。
urlExpiration	Integer	过期时间，单位秒。

请求示例

```
GET /v1/instance/{instanceId}/backup/{backupId}/url HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "url": "http://bos-sandbox-bj.baidu-int.com/scs-backup-rdb-bucket/expiration7/2680_11029_4f2082af-052d-4446-b517-787e259b26b0?authorization=bce-auth-v1%2Fce16d1f34a4e4b5481769e19322ff0d7%2F2019-08-29T11%3A42%3A33Z%2F1800%2Fhost%2Fbfc628b70b4e5f93b74903d6be86f47259849bfa8206afc07a23a0e688b4d6",
 "urlExpiration": 1800
}
```

修改备份备注API

请求说明

通过调用API来进行修改备份备注。

请求结构

```
PUT /v1/instance/{instanceId}/backup/{batchId}/comment HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
batchId	String	Path	是	备份ID
comment	String	Body	否	备注。字符长度限制0-256个字符。

返回头域

除公共头部，无其它特殊头部。

返回参数

无。

请求示例

```
PUT /v1/instance/{instanceId}/backup/{batchId}/comment HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

返回示例



```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

手动备份API

请求说明

- 通过调用API来进行手动备份。

请求结构

```
POST /v1/instance/{instanceId}/backup HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
comment	String	Body	否	备注。字符长度限制0-256个字符。

返回头域

除公共头部，无其它特殊头部。

返回参数

无。

请求示例

```
POST /v1/instance/{instanceId}/backup HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

网络安全

查询IP白名单

请求说明

- 查询允许访问实例的IP白名单。

请求结构

```
GET /v1/instance/{instanceId}/securityIp HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityIps": ["securityIps0","securityIps1"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
instanceId	String	是	URL参数	待查询的实例ID

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	类型	描述
securityIps	List<String>	IP白名单列表, 包括常规地址: 如192.168.0.1，CIDR地址: 如192.168.1.0/24，0.0.0.0/0代表允许所有地址

请求示例

```
GET /v1/instance/scs-bj-ykfiztgpxjny/securityIp HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "securityIps": [
 "0.0.0.0/0",
 "192.0.0.1",
 "192.168.0.1",
 "192.168.1.0/24"
]
}
```

增加IP白名单

请求说明

- 设置可以访问实例的IP白名单。

请求结构

```
PUT /v1/instance/{instanceId}/securityIp?clientToken={clientToken} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityIps": ["securityIps1"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数	参数名称	类型	是否必须	参数位置	描述	----- ----- ----- ----- -----
	clientToken	String	是	URL参数	幂等性Token，是一个长度不超过64位的ASCII字符串	

instanceId | String | 是 | URL参数 | 实例Id | | securityIps | List<String> | 是 | RequestBody参数 | IP白名单分组下的IP列表。包括  
常规地址: 如192.168.0.1，CIDR地址: 如192.168.1.0/24，0.0.0.0/0代表允许所有地址 |

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/securityIp HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c63bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "securityIps": ["192.0.0.1"]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

删除IP白名单

请求说明

- 设置可以访问实例的IP白名单。

请求结构

```
DELETE /v1/instance/{instanceId}/securityIp?clientToken={clientToken} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityIps": ["securityIps0", "securityIps1"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数 | 参数名称 | 类型 | 是否必须 | 参数位置 | 描述 | |-----|-----|-----|-----|-----|  
-----| | clientToken | String | 是 | URL参数 | 幂等性Token，是一个长度不超过64位的ASCII字符串 | |  
instanceId | String | 是 | URL参数 | 实例Id | | securityIps | List<String> | 是 | RequestBody参数 | IP白名单分组下的IP列表。包括  
常规地址: 如192.168.0.1，CIDR地址: 如192.168.1.0/24，0.0.0.0/0代表允许所有地址 |

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
DELETE /v1/instance/scs-bj-ykfiztgpjny/securitylp HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "securitylps": ["192.0.0.1"]
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

修改访问密码

请求说明

- 修改访问密码。

请求结构

```
PUT /v1/instance/{instanceId}/modifyPassword?clientToken={clientToken} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "password": "password"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
clientToken	String	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串
instanceId	String	是	URL参数	实例Id
password	String	是	Request Body参数	密码长度8~16位，至少包含字母、数字和特殊字符中两种。允许的特殊字符包括 \$^*()_+=,密码需要加密传输，禁止明文传输，详情请参考 <a href="#">密码加密传输规范定义</a>

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/instance/scs-bj-ykfiztgpxjny/modifyPassword HTTP/1.1
HOST: redis.bj.baidubce.com
ContentType: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c63bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "password": "!123wss!212"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

域名交换

请求说明

将源实例的域名与目标实例的域名交换。

请求结构

```
POST /v2/instance/swapDomain HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "sourceInstanceId": "sourceInstanceId",
 "targetInstanceId": "targetInstanceId"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必须	参数位置	描述
sourceInstanceId	String	true	RequestBody	源实例ID
targetInstanceId	String	true	RequestBody	目标实例ID

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/instance/swapDomain HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/4386541c7b154d12a322912235a5a5b9/2022-01-28T06:45:44Z/1800/host/310a6f098ac9f790ca6d52d8ab42c0bf9f88e16af212558dc654878e55dc726a
{
 "sourceInstanceId": "scs-source-instance-id",
 "targetInstanceId": "scs-target-instance-id"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

## 热活实例组

### 前置检查

#### 请求说明

热活实例组前置检查，检查数据配置及网络联通性，不通过检查不能创建。

- leader：主角色类型。
- follower：从角色类型。

#### 请求结构

```
POST /v2/group/check HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "leader" : {
 "leaderRegion" : "leaderRegion",
 "leaderId" : "leaderId"
 },
 "followers" : [{
 "followerId" : "followerId",
 "followerRegion" : "followerRegion"
 }]
}
```

#### 请求头域

除公共头域外，无其它特殊头域。

#### 请求参数

参数名称	类型	是否必选	参数位置	描述
leader	Leader	是	RequestBody参数	主角色信息
followers	List<FollowersItem>	否	RequestBody参数	从角色信息

#### Leader

参数名称	类型	是否必选	描述
leaderId	String	是	热活实例组主角色实例ID
leaderRegion	String	是	热活实例组主角色所在地域

#### FollowersItem

参数名称	类型	是否必选	描述
followerId	String	是	从角色实例ID
followerRegion	String	是	从角色实例所在地域

**返回头域** 除公共头域外，无其它特殊头域。

#### 返回参数

参数名称	类型	描述
connectionResults	List< <a href="#">ConnectionResults</a> >	如果存在多个实例，检查主从、从从之间的网络联通性
leaderResult	LeaderResult	主角色实例数据配置检查结果
followerResult	List< <a href="#">FollowerResult</a> >	从角色实例数据配置检查结果

ConnectionResults

参数名称	类型	描述
sourceId	String	源端集群ID
sourceRole	String	源端集群角色
targetId	String	目标端集群ID
targetRole	String	目标端集群角色
connectable	Boolean	是否连通（true:连通 false：不连通）

LeaderResult

参数名称	类型	描述
version	Boolean	版本是否为集群版（true:是，false：否）
clusterStatus	Boolean	集群状态是否是running（true:是，false：否）
replicationNum	Boolean	副本数（true:是，false：否）（主角色副本数需为2）
flavor	Boolean	单分片容量（true:是，false：否）（主角色单分片在8G及以下）
joined	Boolean	是否已加入热活实例组（true：是，false：否）
noPasswd	Boolean	是否无密码（true：是，false：否）
noSecurityGroup	Boolean	是否无vpc侧安全组（true：是，false：否）
isHitX1	Boolean	是否是新架构（true:是 false：否）

FollowerResult

参数名称	类型	描述
followerId	String	从角色ID
noData	Boolean	是否没有数据（true:是，false: 否）
version	Boolean	版本是否为集群版（true:是，false：否）
clusterStatus	Boolean	集群状态是否是running（true:是，false：否）
shardNum	Boolean	分片数是否和主一致（true:是，false：否）
replicationNum	Boolean	副本数（true:是，false：否）（主角色副本数需为2）
flavor	Boolean	单分片容量（true:是，false：否）（主角色单分片在8G及以下）
joined	Boolean	是否已加入热活实例组（true：是，false：否）
noPasswd	Boolean	是否无密码（true：是，false：否）
noSecurityGroup	Boolean	是否无VPC侧安全组（true：是，false：否）
isHitX1	Boolean	是否是新架构（true:是 false：否）

请求示例

```
POST /v2/group/check HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de

{
 "leader" : {
 "leaderRegion" : "bj",
 "leaderId" : "scs-bj-bdtzbvImghbu"
 },
 "followers" : [{
 "followerId" : "scs-bdbl-dzkqigawuhzy",
 "followerRegion" : "bd"
 }]
}
```

### 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5

{
 "leaderResult": {
 "version": true,
 "clusterStatus": true,
 "replicationNum": true,
 "flavor": true,
 "joined": false,
 "noPasswd": true,
 "noSecurityGroup": true,
 "isHitX1": true
 },
 "followerResult": [
 {
 "followerId": "scs-bdbl-dzkqigawuhzy",
 "noData": false,
 "version": false,
 "engineVersion": true,
 "clusterStatus": false,
 "shardNum": false,
 "replicationNum": true,
 "flavor": true,
 "joined": false,
 "noPasswd": true,
 "noSecurityGroup": true,
 "isHitX1": true
 }
],
 "connectionResults": [
 {
 "sourceId": "scs-bj-bdtzbvImghbu",
 "sourceRole": "leader",
 "targetId": "scs-bdbl-dzkqigawuhzy",
 "targetRole": "follower",
 "connectable": true
 }
]
}
```



## 创建热活实例组

### 请求说明

创建热活实例组。

### 请求结构

```
POST /v2/group/create HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "leader" : {
 "groupName" : "groupName",
 "leaderRegion" : "leaderRegion",
 "leaderId" : "leaderId"
 }
}
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必选	参数位置	描述
leader	Leader	是	RequestBody参数	主角色信息

#### Leader

参数名称	类型	是否必选	描述
groupName	String	是	热活实例组名称。规则：支持大小写字母、数字以及_-.等特殊字符，长度6~32
leaderId	String	是	主角色实例ID
leaderRegion	String	是	主角色地域

### 返回头域

除公共头域外，无其它特殊头域。

### 返回参数

参数名称	类型	描述
groupId	String	热活实例组ID

### 请求示例

```
POST /v2/group/create HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "leader" : {
 "groupName" : "hotGroupName",
 "leaderRegion" : "bj",
 "leaderId" : "scs-bdbl-dzkqigawuhzy"
 }
}
```

### 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "groupId" : "scs-group-qplfjtozixy"
}
```

获取热活实例组列表

请求说明

获取热活实例组列表。

请求结构

```
POST /v2/group/list HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 pageNo: 1,
 pageSize: 10
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	描述	是否必填
pageSize	int	每页数量	是
pageNo	int	页码	是

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
result	List<Result>	列表结果
totalCount	int	数据总量
pageNo	int	当前页码
pageSize	int	每页显示的数量

**Result参数信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | leaderName | String | 主角色的名称 | | leaderShowId | String | 主角色的ID | | leaderRegion | String | 主角色的地域 | | groupId | String | 实例组ID | | groupName | String | 实例组名称 | | groupStatus | String | 实例组状态 | | clusterNum | Integer | 实例组的集群数量 | | groupCreateTime | String | 实例组创建时间 | | forbidWrite | Integer | 禁写标志（0 未禁写， 1 禁写） | | groupType | String | 实例组类型。标准版：standalone；集群版：bdrp |

请求示例

```
POST /v2/group/list
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c63bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de

{
 pageNo: 1,
 pageSize: 10
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "totalCount": 1,
 "pageNo": 1,
 "pageSize": 10,
 "result": [
 {
 "groupId": "scs-group-ajvwhbpgdshx",
 "groupName": "scs_test",
 "groupStatus": "running",
 "clusterNum": 1,
 "groupCreateTime": "2022-10-09T07:20:17Z",
 "forbidWrite": 0,
 "groupType": "standalone",
 "leaderName": "xuepeng_test",
 "leaderShowId": "scs-bj-bdtzbvImghbu",
 "leaderRegion": "bj"
 }
]
}
```

热活实例组详情

请求说明

热活实例组详情。

请求结构

```
GET /v2/group/{groupId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	待查询的热活实例组ID

请求头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
leader	Leader	主角色信息
followers	List<Followers>	从角色信息
groupId	String	实例组ID
groupName	String	实例组名称
groupStatus	String	实例组状态
clusterNum	Integer	实例组的集群数量
groupCreateTime	String	实例组创建时间
forbidWrite	Integer	禁写标志（0 未禁写， 1 禁写）
groupType	String	实例组类型。标准版：standalone；集群版：bdrp

**Leader** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | clusterName | String | 主角色实例名称 | | clusterShowId | String | 主角色实例ID | | region | String | 主角色地域 | | status | String | 主角色实例状态 | | totalCapacityInGB | Float | 总容量 | | usedCapacityInGB | Integer | 已用容量 | | shardNum | Integer | 分片数 | | flavor | Integer | 单分片容量 | | qpsWrite | Long | 写流量阈值 | | qpsRead | Long | 读流量阈值 | | staleReadable | Boolean | 从角色脏读开关状态（true:打开，false:关闭） | | forbidWrite | Integer | 禁写标志（0 未禁写， 1 禁写） | | availabilityZone | String | 可用区 | | expiredTime | Date | 过期时间 |

**Followers** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | clusterName | String | 从角色实例名称 | | clusterShowId | String | 从角色实例ID | | region | String | 从角色地域 | | status | String | 从角色实例状态 | | totalCapacityInGB | Float | 总容量 | | usedCapacityInGB | Integer | 已用容量 | | shardNum | Integer | 分片数 | | flavor | Integer | 单分片容量 | | qpsWrite | Long | 写流量阈值 | | qpsRead | Long | 读流量阈值 | | staleReadable | Boolean | 从角色脏读开关状态（true:打开，false:关闭） | | forbidWrite | Integer | 禁写标志（0 未禁写， 1 禁写） | | availabilityZone | String | 可用区 | | expiredTime | Date | 过期时间 |

请求示例

```
GET /v2/group/scs-group-ajvwhbpgdshx HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "groupId": "scs-group-ajvwhbpgdshx",
 "groupName": "scs_test",
 "groupStatus": "running",
 "clusterNum": 1,
 "groupCreateTime": "2022-10-09T07:20:17Z",
 "forbidWrite": 0,
 "groupType": "standalone",
 "leader": {
 "clusterName": "xuepeng_test",
 "clusterShowId": "scs-bj-bdtzbvImghbu",
 "region": "bj",
 "status": "normal",
 "totalCapacityInGB": 1,
 "usedCapacityInGB": 0,
 "shardNum": 1,
 "flavor": 1,
 "qpsWrite": 0,
 "qpsRead": 0,
 "staleReadable": false,
 "forbidWrite": 0,
 "availabilityZone": "zoneC",
 "expiredTime": null
 },
 "followers": []
}
```

释放热活实例组

请求说明

释放热活实例组。

请求结构

```
DELETE /v2/group/{groupId}/release HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID

**返回头域** 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
DELETE /v2/group/scs-group-qplfjtozixy/release HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

添加从集群

请求说明

添加从角色集群。

请求结构

```
POST /v2/group/{groupId}/join HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
followerId	String	是	RequestBody参数	从角色集群ID
followerRegion	String	是	RequestBody参数	从角色集群所在地域
syncMaster	String	是	RequestBody参数	是否同步主角色的参数（sync: 同步主角色参数，nosync：不同步主角色参数）

**返回头域** 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/group/scs-group-qplfjtozixy/join HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 followerId: "scs-bj-ttegbirwfagl",
 followerRegion: "bj",
 syncMaster: "sync"
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

移除从集群

请求说明

移除从角色集群。

请求结构

```
POST /v2/group/{groupId}/quit/{instanceId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
instanceId	String	是	URL参数	从角色集群ID

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/group/scs-group-qplfjtozixy/quit/scs-bj-ttegbirwfag! HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

修改热活实例组名称

请求说明

修改热活实例组名称。

请求结构

```
PUT /v2/group/{groupId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 groupName: "scs_test"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	待查询的热活实例组ID
groupName	String	是	RequestBody参数	新的名称。规则：支持大小写字母、数字以及-_.等特殊字符，长度6~32

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/group/scs-group-dycgmxbxlbow
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 groupName: "scs_test"
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

变更主角色

请求说明

变更指定从角色为主角色，变更后主角色变更为从角色。

请求结构

```
POST /v2/group/{groupId}/setAsLeader/{instanceId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
instanceId	String	是	URL参数	从角色集群ID

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。



请求示例

```
POST /v2/group/scs-group-qpplfjtozixy/setAsLeader/scs-bj-ttegbirwfagl HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

实例组禁写

请求说明

实例组禁写。

请求结构

```
PUT /v2/group/{groupId}/forbidWrite HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "forbidWriteFlag": forbidWriteFlag
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
forbidWriteFlag	Boolean	是	RequestBody参数	禁写标识（false 未禁写 true 禁写）

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/group/scs-group-qpplfjtozixy/forbidWrite HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "forbidWriteFlag": true
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

## 设置流控规则

### 请求说明

设置流控规则。

### 请求结构

```
PUT /v2/group/{groupId}/qps HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "clusterShowId": "clusterShowId",
 "qpsWrite" : qpsWrite,
 "qpsRead" : qpsRead,
}
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
clusterShowId	String	是	RequestBody参数	集群ID
qpsWrite	Integer	否	RequestBody参数	写流量限制。（读写流量限制至少传一个）
qpsRead	Integer	否	RequestBody参数	读流量限制。（读写流量限制至少传一个）

### 返回头域

除公共头域外，无其它特殊头域。

### 返回参数

无特殊返回参数。

### 请求示例

```
PUT /v2/group/scs-group-qpplfjtozixy/qps HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "clusterShowId": "scs-bj-ttegbirwfagl",
 "qpsWrite" : 100,
 "qpsRead" : 100,
}
```

### 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

## 获取从角色同步状态

### 请求说明

获取从角色同步状态。

### 请求结构

```
GET /v2/group/{groupId}/syncStatus HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID

返回头域 除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
followers	List< <a href="#">Followers</a> >	热活实例组ID

Followers

参数名称	类型	描述
clusterShowId	String	集群ID
syncStatus	String	状态。online：正常
maxOffset	Integer	主从同步offset的最大差值
lag	Integer	主从同步时延

请求示例

```
GET /v2/group/scs-group-vobpzinoqadm/syncStatus HTTP/1.1
HOST : redis.bj.baidubce.com
Content-Type : application/json
Authorization : bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "followers": [{
 "clusterShowId": "scs-bdbl-stsqbjompvol",
 "syncStatus": "offline",
 "maxOffset": 0,
 "lag": 0
 }]
}
```

获取IP白名单列表

请求说明

获取IP白名单列表。

请求结构

```
GET /v2/group/{groupId}/white_list HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID

返回头域 除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
whiteLists	List<String>	列表结果

请求示例

```
GET /v2/group/scs-group-qpplfjtozixy/white_list HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "whiteLists": ["127.0.0.1"]
}
```

添加IP白名单

请求说明

添加IP白名单。

请求结构

```
PUT /v2/group/{groupId}/white_list/add HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "whiteLists" : ["whiteLists0", "whiteLists1"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
whiteLists	List	是	RequestBody参数	IP列表，逗号分隔

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/group/scs-group-qplfjtozixy/white_list/add HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "whiteLists" : ["127.0.0.1", "111.112.113.114/32"]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

删除IP白名单

请求说明

删除IP白名单。

请求结构

```
PUT /v2/group/{groupId}/white_list/delete HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "whiteLists" : ["whiteLists0", "whiteLists1"]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
whiteLists	List<String>	是	RequestBody参数	IP列表，逗号分隔

返回头域 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/group/scs-group-qpplfjtozixy/white_list/delete HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "whiteLists" : ["127.0.0.1", "111.112.113.114/32"]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

设置从角色脏读

请求说明

设置从角色脏读：

- Redis集群版支持；Redis标准版不支持。

请求结构

```
PUT /v2/group/{groupId}/stale_readable HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 followerId: "followerId",
 staleReadable: staleReadable
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
groupId	String	是	URL参数	热活实例组ID
followerId	String	是	RequestBody参数	从角色集群ID
staleReadable	Boolean	是	RequestBody参数	是否开启从角色脏读（true：开启，false：关闭）

**返回头域** 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/group/scs-group-qpplfjtozixy/stale_readable HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 followerId: "scs-bj-ttegbirwfag!",
 staleReadable: true
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

多活实例组

多活实例组删除

🔗 多活实例组删除

接口描述

本接口用于删除多活实例组。请确认当前实例组状态正常时删除。

请求结构

```
DELETE /v{version}/syncGroup/{syncGroupShowId} HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupShowId	String	是	PATH参数	多活实例组ID。

响应头域

除公共头域外，无其它特殊头域。

响应参数

无。

请求示例

```
DELETE http://redis.fsh.baidubce.com/v2/syncGroup/scs-syncgroup-zyroblrgwpl
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

获取同步状态

🔗 获取同步状态

接口描述

本接口用于获取实例组中成员建的同步状态。

请求结构

```
GET /v{version}/syncGroup/{syncGroupShowId}/syncStatus HTTP/1.1
Host: bcc.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupShowId	String	是	PATH参数	实例组ID。

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
syncGroupShowId	String	实例组ID。
syncStatus	List<SyncStatusItem>	状态列表。

**SyncStatusItem** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | memberId | String | 成员集群ID。 | | status | String | 同步状态。normal为正常，其他为异常。 |

请求示例

```
GET http://redis.fsh.baidubce.com/v2/syncGroup/scs-syncgroup-zyroblrgwpl/syncStatus
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "syncGroupShowId": "scs-syncgroup-zyroblrgwpl",
 "syncStatus": [
 {
 "memberId": "scs-bj-wqocnchhrqft",
 "status": "normal"
 },
 {
 "memberId": "scs-bj-spxwdizpqfgq",
 "status": "normal"
 },
 {
 "memberId": "scs-bj-wrphymepyqix",
 "status": "normal"
 }
]
}
```



## 修改多活实例组名称

🔗 修改多活实例组名称

### 接口描述

本接口用于修改多活实例组名称。

### 请求结构

```
PUT /v{version}/syncGroup/{syncGroupShowId} HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "groupName": "groupName"
}
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupShowId	String	是	PATH参数	多活实例组ID。
groupName	String	是	RequestBody参数	修改后的实例组名称。

### 响应头域

除公共头域外，无其它特殊头域。

### 响应参数

无。

### 请求示例

```
PUT http://redis.fsh.baidubce.com/v2/syncGroup/scs-syncgroup-zyroblrgwpl
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "groupName": "cll1100"
}
```

### 响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

## 多活实例组详情

🔗 多活实例组详情

### 接口描述

本接口用于查询指定的多活实例组的详情。

### 请求结构

```
GET /v{version}/syncGroup/{syncGroupShowId} HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupShowId	String	是	PATH参数	多活实例组的ID。

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
syncGroupShowId	String	实例组ID。
syncGroupName	String	实例组名称。
status	String	实例组状态。
clusterNum	Integer	实例组中成员集群的数量。
nodeType	String	实例组中成员集群的规格。
netConn	String	网络联通状态。忽略该字段。
confilctResolution	String	冲突解决办法。忽略该字段。
syncGroupCreateTime	String	多活实例组创建时间。
sameSpec	Boolean	实例组中所有成员的规格是否一致。
sameShardNum	Boolean	实例组中所有成员的分片数量是否一致。
cluster	List<ClusterItem>	多活组列表数据。

**ClusterItem** | 参数名称 | 类型 | 描述 |

| ----- | ----- | ----- | | clusterShowId | String | 成员集群的ID。 | | clusterName | String | 成员集群的名称。 | | region | String | 成员集群所在地域。 | | clusterStatus | String | 集群状态。此状态为集群在实例组中的状态，非集群本身状态。 | | clusterEngine | String | 集群引擎类型。 | | createTime | String | 集群创建时间。 | | totalCapacityInGb | Float | 集群总容量。 | | usedCapacityInGb | Float | 集群已用容量。 | | expiredTime | String | 集群到期时间。 | | shardList | List<String> | 集群分片列表。查询多活组监控数据时使用。 | | syncFlow | List<SyncFlowItem> | 集群BLB信息列表。查询多活组监控数据时使用。 |

**SyncFlowItem** | 参数名称 | 类型 | 描述 |

| ----- | ----- | ----- | | targetBLBIp | String | 同步流目标端BLBIP。 | | targetBLBPort | String | 同步流目标端BLB端口。 | | targetClusterShowId | String | 同步流目标端集群ID。 |

请求示例

```
GET http://redis.fsh.baidubce.com/v2/syncGroup/scs-syncgroup-zyroblrgwpl
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

```
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "syncGroupShowId": "scs-syncgroup-zyroblrgwpl",
 "syncGroupName": "clitest",
 "status": "normal",
 "clusterNum": 3,
 "nodeType": "cache.n1.micro",
 "netConn": "",
 "conflictResolution": "crdt",
 "syncGroupCreateTime": "2023-09-21T06:15:19Z",
 "sameSpec": true,
 "sameShardNum": true,
 "cluster": [
 {
 "clusterId": 43354,
 "clusterName": "redis-lmsrsyv0_4",
 "clusterShowId": "scs-bj-wqocnchhrqft",
 "region": "bj",
 "clusterStatus": "normal",
 "clusterEngine": "redis",
 "createTime": "2023-09-21 06:07:18",
 "totalCapacityInGb": 1,
 "usedCapacityInGb": 0.0,
 "expiredTime": null,
 "shardList": [
 "scs-bj-wqocnchhrqft-0"
],
 "syncFlow": [
 {
 "targetBLBip": "172.17.34.100",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-spxwdizpqfgq"
 },
 {
 "targetBLBip": "172.17.34.106",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-wrphymepyqix"
 }
]
 },
 {
 "clusterId": 43353,
 "clusterName": "redis-lmsrsyv0_3",
 "clusterShowId": "scs-bj-spxwdizpqfgq",
 "region": "bj",
 "clusterStatus": "normal",
 "clusterEngine": "redis",
 "createTime": "2023-09-21 06:07:18",
 "totalCapacityInGb": 1,
 "usedCapacityInGb": 0.0,
 "expiredTime": null,
 "shardList": [
 "scs-bj-spxwdizpqfgq-0"
],
 "syncFlow": [
 {
 "targetBLBip": "172.17.34.108",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-wqocnchhrqft"
 }
]
 }
]
}
```

```
{
 "targetBLBIP": "172.17.34.106",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-wrphymepyqix"
}
],
{
 "clusterId": 43352,
 "clusterName": "redis-lmsrsyv0_2",
 "clusterShowId": "scs-bj-wrphymepyqix",
 "region": "bj",
 "clusterStatus": "normal",
 "clusterEngine": "redis",
 "createTime": "2023-09-21 06:07:18",
 "totalCapacityInGb": 1,
 "usedCapacityInGb": 0.0,
 "expiredTime": null,
 "shardList": [
 "scs-bj-wrphymepyqix-0"
],
 "syncFlow": [
 {
 "targetBLBIP": "172.17.34.108",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-wqocnchhrqft"
 },
 {
 "targetBLBIP": "172.17.34.100",
 "targetBLBPort": 6379,
 "targetClusterShowId": "scs-bj-spxwdizpqfgq"
 }
]
}
]
```

### 多活实例组列表

🔗 多活实例组列表

#### 接口描述

本接口用于查询多活实例组列表。

#### 请求结构

```
GET /v{version}/syncGroup/list?page=1&pageSize=10 HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
```

#### 请求头域

除公共头域外，无其它特殊头域。

#### 请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
page	Integer	是	Query参数	当前页码。
pageSize	Integer	是	Query参数	每页显示的数量。

#### 响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
totalCount	Integer	总数量。
pageNo	Integer	当前页码。
pageSize	Integer	每页显示的数量。
result	List<ListItem>	多活组列表数据。

**ListItem** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | syncGroupId | String | 实例组ID。 | | syncGroupName | String | 实例组名称。 | | status | String | 实例组状态。 | | clusterNum | Integer | 实例组中成员集群的数量。 | | nodeType | String | 实例组中成员集群的规格。 | | netConn | String | 网络联通状态。忽略该字段。 | | conflictResolution | String | 冲突解决办法。忽略该字段。 | | syncGroupCreateTime | String | 多活实例组创建时间。 | | sameSpec | Boolean | 实例组中所有成员的规格是否一致。 | | sameShardNum | Boolean | 实例组中所有成员的分片数量是否一致。 | | cluster | List | 多活组列表数据。 |

**ClusterItem** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | clusterShowId | String | 成员集群的ID。 | | region | String | 成员集群所在地域。 |

请求示例

```
GET http://redis.fsh.baidubce.com/v2/syncGroup/list?page=1&pageSize=10
Host: bcc.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "totalCount": 1,
 "pageNo": 1,
 "pageSize": 10,
 "result": [
 {
 "syncGroupShowId": "scs-syncgroup-zyroblrgwpl",
 "syncGroupName": "clltest",
 "status": "add-member",
 "clusterNum": 3,
 "nodeType": "cache.n1.micro",
 "netConn": "",
 "confilctResolution": "crdt",
 "syncGroupCreateTime": "2023-09-21T06:15:19Z",
 "sameSpec": true,
 "sameShardNum": true,
 "cluster": [
 {
 "clusterShowId": "scs-bj-wqocnchhrqft",
 "region": "bj"
 },
 {
 "clusterShowId": "scs-bj-spxwdizpqfgq",
 "region": "bj"
 },
 {
 "clusterShowId": "scs-bj-wrphymepyqix",
 "region": "bj"
 }
]
 }
]
}
```

## 前置检查

### 前置检查

#### 接口描述

本接口用于创建实例组和实例组中添加集群的前置检查。请确保前置检查全部通过后创建多活实例组。

#### 请求结构

```
POST /v{version}/syncGroup/check HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "syncGroupShowId": "syncGroupShowId",
 "members": [
 {
 "memberId": "scs-fsh-ifjmkpkdzceq",
 "region": "fsh"
 },
 {
 "memberId": "scs-fsh-luoreupenrtn",
 "region": "fsh"
 }
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupShowId	String	否	RequestBody参数	多活实例组ID。已有的多活实例组中添加集群时必须填，新建多活实例组时不需要改参数。
members	List<Member>	是	RequestBody参数	需要检查的多活组成员的集群ID。

**Member** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | memberId | String | 集群ID。 | | region | String | 集群所在地域。 |

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
checkResult	List	多活组成员检查结果。
connectionResults	List	网络连通性检查结果。

CheckItem

参数名称	类型	描述
memberId	String	实例组成员集群ID。
noData	Boolean	实例是否无数据。
version	Boolean	实例引擎类型及版本类型是否支持。
engineVersion	Boolean	引擎版本是否支持。说明：Redis企业集群版支持4.0和6.0版本。
clusterStatus	Boolean	网络连通性检查结果。
shardNum	Boolean	实例分片数是否一致。
replicationNum	Boolean	实例副本数是否一致。
flavor	Boolean	实例单分片容量是否一致。
notJoined	Boolean	实例是否未加入实例组。
noSecurityGroup	Boolean	实例是否未关联VPC侧安全组。
isHitX1	Boolean	集群当前架构是否支持。
isAppendOnlyOn	Boolean	实例是否开启aof。
samePasswd	Boolean	实例密码是否一致。

ConnectionItem

参数名称	类型	描述
sourceId	String	源实例ID。
targetId	String	目标实例ID。
connectable	Boolean	源端到目标端的连通性是否正常。

请求示例

```
POST http://redis.fsh.baidubce.com/v2/syncGroup/check
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "members": [
 {
 "memberId": "scs-fsh-ifjmkpkdzceq",
 "region": "fsh"
 },
 {
 "memberId": "scs-fsh-luoreupenrtn",
 "region": "fsh"
 }
]
}
```

响应示例



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "checkResult": [
 {
 "memberId": "scs-fsh-ifjmkpkdzceq",
 "noData": true,
 "version": true,
 "engineVersion": true,
 "clusterStatus": true,
 "shardNum": false,
 "replicationNum": true,
 "flavor": true,
 "notJoined": true,
 "noSecurityGroup": true,
 "isHitX1": true,
 "isAppendOnlyOn": true,
 "samePasswd": true
 },
 {
 "memberId": "scs-fsh-luoreupenrtn",
 "noData": true,
 "version": true,
 "engineVersion": true,
 "clusterStatus": true,
 "shardNum": false,
 "replicationNum": true,
 "flavor": true,
 "notJoined": true,
 "noSecurityGroup": true,
 "isHitX1": true,
 "isAppendOnlyOn": true,
 "samePasswd": true
 }
],
 "connectionResults": [
 {
 "sourceId": "scs-fsh-ifjmkpkdzceq",
 "targetId": "scs-fsh-luoreupenrtn",
 "connectable": true
 },
 {
 "sourceId": "scs-fsh-luoreupenrtn",
 "targetId": "scs-fsh-ifjmkpkdzceq",
 "connectable": true
 }
]
}
```

## 创建多活实例组

### 创建多活实例组

#### 接口描述

本接口用于创建多活实例组。支持集群跨地域创建实例组。

本接口为v2接口。

#### 注意事项

多活实例组仅支持Redis企业集群版，版本为6.0和4.0的集群。

多活实例组中的集群规格配置必须一致。

多活实例组中最多支持添加5个集群。

创建前请确保前置检查全部通过。

请求结构

```
POST /v{version}/syncGroup/create HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "syncGroupName": "syncname",
 "members": [
 {
 "memberId": "scs-bj-zzkhtsrlnrcf",
 "region": "bj"
 },
 {
 "memberId": "scs-bd-dcyeumpmbars",
 "region": "bd"
 }
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为2。
syncGroupN ame	String	是	RequestBody 参数	多活实例组名称。规则：支持大小写字母、数字以及-_.等特殊字符，以字母开头，长度6~32位
members	List<Memb er>	是	RequestBody 参数	多活实例组成员。最多支持5个成员。

**Member** | 参数名称 | 类型 | 描述 |  
| ----- | ----- | ----- | | memberId | String | 集群ID。 | | region | String | 集群所在地域。 |

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
syncGroupShowId	String	多活实例组ID。

请求示例

```
POST http://redis.bj.baidubce.com/v2/syncGroup/create
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "syncGroupName": "syncname",
 "members": [
 {
 "memberId": "scs-fsh-zzkhtsrlmncf",
 "region": "fsh"
 },
 {
 "memberId": "scs-fsh-dcyeumpmbars",
 "region": "fsh"
 }
]
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "syncGroupShowId": "std1"
}
```

参数模版

创建参数模版

请求说明

创建参数模版，不区分地域。

请求结构

```
POST /v2/template/create HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "engineVersion" : "engineVersion",
 "templateType" : templateType,
 "clusterType" : "clusterType",
 "engine" : "engine",
 "name" : "name",
 "comment" : "测试备注",
 "parameters" : [{
 "confName" : "confName",
 "confModule" : confModule,
 "confValue" : "confValue,confValue1",
 "confType" : confType
 }]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
name	String	是	RequestBody参数	模板名称，格式要求：大小写字母、数字、中文以及-/_/.特殊字符，必须以字母或者中文开头，长度1-65
engine	String	是	RequestBody参数	引擎（当前仅支持redis）
engineVersion	String	是	RequestBody参数	引擎版本
clusterType	String	是	RequestBody参数	集群类型（master_slave\cluster）
templateType	Integer	是	RequestBody参数	模板类型（1、自定义参数模板）
comment	String	否	RequestBody参数	备注
parameters	List<Parameters>	是	RequestBody参数	参数列表，系统模版内的数据，和引擎及引擎版本有关

Parameters参数信息

参数名称	类型	是否必选	描述
confName	String	是	参数名称（可参考系统模版）
confValue	String	是	参数值（可修改对应的值，可参考系统模版在confRange的允许范围内修改）
confType	Integer	是	参数类型（可参考系统模版）
confModule	Integer	是	可参考系统模版

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
templateId	Integer	参数模版数字ID
templateShowId	String	参数模版展示ID

请求示例

```
POST /v2/template/create HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "engineVersion" : "5.0",
 "templateType" : 1,
 "clusterType" : "master_slave",
 "engine" : "redis",
 "name" : "testTemplateName",
 "comment" : "测试备注",
 "parameters" : [{
 "confName" : "disable_commands",
 "confModule" : 1,
 "confValue" : "flushall,flushdb",
 "confType" : 3
 }]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "templateShowId" : "scs-tmpl-xxxxxx",
 "templateId" : 0
}
```

获取参数模版列表

请求说明

查看当前账号有权限的参数模版列表，不区分地域。

请求结构

```
GET /v2/template/list?marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
marker	String	否	Query参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	Integer	否	Query参数	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000

**返回头域** 除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
result	List<Result>	参数模版列表
marker	String	批量获取列表的查询的起始位置，是一个由系统生成的字符串,起始值可传入-1
maxKeys	Integer	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000
isTruncated	Boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现

**Result参数信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | templateId | Integer | 参数模板数字ID | | templateShowId | String | 参数模板ID | | templateName | String | 参数模板名称 | | parametersNum | Integer | 参数模板参数数量 | | clusterType | String | 集群类型 | | engine | String | 引擎类型 | | engineVersion | String | 引擎版本 | | templateType | Integer | 参数模板类型（1为自定义参数模板） | | needReboot | Integer | 是否需要重启(该参数模版是否有需要重启生效的参数)，0：不需要，1：需要 | | comment | String | 备注 | | createTime | String | 创建时间 | | updateTime | String | 更新时间 | | parameters | List<Parameters> | 参数列表 |

**Parameters参数信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | confName | String | 参数名称 | | confDefault | String | 参数默认值 | | confValue | String | 参数值 | | confType | Integer | 参数类型（1:单选类型 2.数字类型 3.多选类型 4.多选类型，单值） | | confRange | String | 参数范围：  
当confType=1时，以逗号（，）为分割符所有选择项  
当confType=2时，以逗号（，）为分割符，第一项为范围最小值，第二项为范围最大值  
当confType=3，以逗号（，）为间隔符所有可选项  
当confType=4时，以逗号（，）为间隔符所有可选项 | | confModule | Integer | 1:redis 2:proxy 3.redis+proxy 4.metaserver | | confDesc | String | 配置对用户展示含义(转码后的一串字符) | | needReboot | Integer | 该参数是否需要重启，0：不需要，1：需要 | | confRedisVersion | String | 参数对应的redis版本。例如：2.6、3.2、4.0、all | | confCacheVersion | Integer | 生效redis version（和VERSION\_TYPE保持一致） |

请求示例

```
GET /v2/template/list?marker=-1&maxKeys=20 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "result": [{
 "engineVersion": "5.0",
 "templateType": 1,
 "clusterType": "master_slave",
 "needReboot": 0,
 "templateShowId": "scs-templ-zuiflgqnmwrx",
 "updateTime": "2022-04-26T16:17:03Z",
 "templateId": 13,
 "parametersNum": 0,
 "templateName": "test3",
 "engine": "redis",
 "createTime": "2022-04-26T16:17:03Z",
 "comment": "test template",
 "parameters": [{
 "confName": "appendonly",
 "confModule": 1,
 "confCacheVersion": 0,
 "confValue": "no",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confDefault": "yes",
 "confType": 1,
 "confRange": "yes|no|partial",
 "confDesc": "5piv5ZCm5ZCv55SoIEFPRiDmjIHkuYXljJY="
 }]
 }],
 "isTruncated": true,
 "nextMarker": "scs-templ-dycgmxblbow",
 "maxKeys": 10,
 "marker": "-1"
}
```

参数模版详情

请求说明

查看当前账号有权限的参数模版详情，不区分地域。

请求结构

```
GET /v2/template/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID

返回头域 除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
templateId	Integer	参数模板数字ID
templateShowId	String	参数模板ID
templateName	String	参数模板名称
parameterNum	Integer	参数模板参数数量
clusterType	String	集群类型 (master_slave、default、cluster)
engine	String	引擎类型
engineVersion	String	引擎版本
templateType	Integer	参数模板类型（1为自定义参数模板）
needReboot	Integer	是否需要重启(该参数模版是否有需要重启生效的参数)，0：不需要，1：需要
comment	String	备注
createTime	String	创建时间
updateTime	String	更新时间
parameters	List<Parameters>	参数列表

**Parameters参数信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | confName | String | 参数名称 | | confDefault | String | 参数默认值 | | confValue | String | 参数值 | | confType | Integer | 参数类型（1:单选类型 2.数字类型 3.多选类型 4.多选类型，单值） | | confRange | String | 参数范围：

当confType=1时，以逗号（，）为分割符所有选择项

当confType=2时，以逗号（，）为分割符，第一项为范围最小值，第二项为范围最大值

当confType=3，以逗号（，）为间隔符所有可选项

当confType=4时，以逗号（，）为间隔符所有可选项 | | confModule | Integer | 1:redis 2:proxy 3.redis+proxy 4.metaserver | | confDesc | String | 配置对用户展示含义(转码后的一串字符) | | needReboot | Integer | 该参数是否需要重启，0：不需要，1：需要 | | confRedisVersion | String | 参数对应的redis版本。例如：2.6、3.2、4.0、all | | confCacheVersion | Integer | 生效redis version（和VERSION\_TYPE保持一致） | confUserVisible | Integer | 用户是否可见（0 可见，1 不可见）

请求示例

```
GET /v2/template/scs-tmpl-jlbohgyzkxgs HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "templateId": 1126,
 "templateShowId": "scs-tmpl-jlbohgyzkxgs",
 "templateName": "lkjklkjin",
 "parameterNum": 16,
 "clusterType": "master_slave",
 "engine": "redis",
 "engineVersion": "6.0",
 "templateType": 1,
 "needReboot": 1,
 "comment": "",
 "createTime": "2023-06-12T09:06:05Z",
 "updateTime": "2023-06-12T09:06:05Z",
 "parameters": [
 {
 "confName": "appendonly",
 "confDefault": "yes",
 "confValue": "yes",
 "confType": 1,
 "confRange": "yes|no|partial",
 "confModule": 1,
 "confDesc": "是否启用 AOF 持久化",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001,
 "confUserVisible": 0
 }
]
}
```

删除参数模版

请求说明

删除参数模版。

请求结构

```
DELETE /v2/template/delete/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
DELETE /v2/template/delete/scs-tmpl-dycgmxblbow
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

修改参数模版名称

请求说明

修改参数模版名称。

请求结构

```
PUT /v2/template/rename/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 name: "name"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID
name	String	是	RequestBody参数	新的实例名称。要求：大小写字母、数字、中文以及-_/特殊字符，必须以字母或者中文开头，长度1-65

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/template/rename/scs-tmpl-dycgmxblbow
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 name: "scs_test"
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

应用参数模版

请求说明

应用参数模版到实例。

请求结构

```
POST /v2/template/apply/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "rebootType" : rebootType,
 "extra" : "extra",
 "cacheClusterList" : [{
 "cacheClusterShowId" : "cacheClusterShowId",
 "region" : "region"
 }],
 "parameters" : [{
 "confName" : "confName",
 "confModule" : confModule,
 "confValue" : "confValue0,confValue1",
 "confType" : confType
 }]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	否	URL参数	模版ID
extra	String	是	RequestBody参数	是否有额外的修改（0：无；1：有），没有修改的参数不需要传parameters
cacheClusterList	List< <a href="#">cacheClusterShowIdItem</a> >	否	RequestBody参数	实例ID集合
rebootType	Integer	否	RequestBody参数	是否需要重启（0 不重启 1 维护时间重启 2 立即重启 如果是热活的主集群的话不可以重启）
parameters	List < <a href="#">Parameters</a> >	否	RequestBody参数	参数模版中需要修改的参数，extra=1时有效

Parameters

参数名称	类型	是否必选	描述
confName	String	是	参数名称（可参考系统模版）
confValue	String	是	参数值（可修改对应的值，可参考系统模版在confRange的允许范围内修改）
confType	Integer	是	参数类型（可参考系统模版）
confModule	Integer	是	可参考系统模版

cacheClusterShowIdItem

参数名称	类型	是否必选	描述
region	String	是	实例所在地域
cacheClusterShowId	String	是	实例ID

返回头域 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/template/apply/scs-tmpl-dycgmxblbow HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "rebootType" : 0,
 "extra" : "0",
 "cacheClusterList" : [{
 "cacheClusterShowId" : "scs-tmpl-jlbohgyzkxgs",
 "region" : "bd"
 }],
 "parameters" : [{
 "confName" : "disable_commands",
 "confModule" : 1,
 "confValue" : "flushall,flushdb",
 "confType" : 3
 }]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

参数模版添加参数

请求说明

参数模版添加参数。

请求结构

```
POST /v2/template/addParams/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 parameters: [{
 "confName" : "confName",
 "confModule" : confModule,
 "confValue" : "confValue0,confValue1",
 "confType" : confType
 }]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID
parameters	List<Parameters>	是	RequestBody参数	要添加的参数集合

Parameters参数信息

参数名称	类型	是否必选	描述
confName	String	是	参数名称（可参考系统模版）
confValue	String	是	参数值（可修改对应的值，可参考系统模版在confRange的允许范围内修改）
confType	Integer	是	参数类型（可参考系统模版）
confModule	Integer	是	可参考系统模版

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/template/addParams/scs-tmpl-dycgmxbxlbw
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 parameters: [{
 "confName" : "disable_commands",
 "confModule" : 1,
 "confValue" : "flushall,flushdb",
 "confType" : 3
 }]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

参数模版修改参数

请求说明

参数模版修改参数。

请求结构

```
PUT /v2/template/modifyParams/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 parameters: [{
 "confName" : "confName",
 "confModule" : confModule,
 "confValue" : "confValue",
 "confType" : confType
 }]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID
parameters	List<Parameters>	是	RequestBody参数	要添加的参数集合

Parameters参数信息

参数名称	类型	是否必选	描述
confName	String	是	参数名称（可参考系统模版）
confValue	String	是	参数值（可修改对应的值，可参考系统模版在confRange的允许范围内修改）
confType	Integer	是	参数类型（可参考系统模版）
confModule	Integer	是	可参考系统模版

返回头域 除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
PUT /v2/template/modifyParams/scs-trmpl-dycgmxbxlbow
HOST: redis.bj.baidubce.com
ContentType: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 parameters: [{
 "confName" : "disable_commands",
 "confModule" : 1,
 "confValue" : "flushall",
 "confType" : 3
 }]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

参数模版删除参数

接口说明

修改参数模版名称。

请求结构

```
POST /v2/template/deleteParams/{templateShowId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 parameters: [
 "parameters0" , "parameters1"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	待查询的参数模版ID
parameters	List<String>	是	RequestBody参数	要删除的参数名称集合

返回头域

除公共头域外，无其它特殊头域。

返回参数

无特殊返回参数。

请求示例

```
POST /v2/template/deleteParams/scs-tmpl-dycgmxblbow
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 parameters: [
 "appendonly"
]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
```

获取系统参数列表

请求说明

系统参数列表，包含全量的系统参数数据；创建参数模版时可从中选取需要的参数，需要注意不同引擎和引擎版本的支持。

请求结构

```
GET /v2/template/system?engine={engine}&engineVersion={engineVersion}&clusterType={clusterType} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
engine	String	否	URL参数	引擎，可选值：redis
engineVersion	String	否	URL参数	引擎版本，示例：5.0
clusterType	String	否	URL参数	集群类型，可选值：master_slave，cluster

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
success	Boolean	是否成功
result	List<Result>	系统参数列表

**Result参数信息** | 参数名称 | 类型 | 描述 | |-----|-----|-----| | confName | String | 参数名称 | | confDefault | String | 参数默认值 | | confValue | String | 参数值 | | confType | Integer | 参数类型（1:单选类型 2.数字类型 3.多选类型 4.多选类型，单值） | | confRange | String | 参数范围：当confType=1时，以逗号（，）为分割符所有选择项；当confType=2时，以逗号（，）为分割符，第一项为范围最小值，第二项为范围最大值；当confType=3，以逗号（，）为间隔符所有可选项；当confType=4时，以逗号（，）为间隔符所有可选项； | | confModule | Integer | 1:redis 2:proxy 3.redis+proxy 4.metaserver | | confDesc | String | 配置对用户展示含义(转码后的一串字符) | | needReboot | Integer | 该参数是否需要重启，0：不需要，1：需要 | | confRedisVersion | String | 参数对应的redis版本。例如：2.6、3.2、 4.0、5.0、 all(所有版本都适配) | | confCacheVersion | Integer | 生效redis version（和VERSION\_TYPE保持一致） |

请求示例

```
GET /v2/template/system?engine=redis&engineVersion=5.0&clusterType=cluster HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "success": true,
 "result": [
 {
 "confName": "qpsquota",
 "confDefault": "100000",
 "confValue": null,
 "confType": 2,
 "confRange": "0-500000",
 "confModule": 2,
 "confDesc": "cmVzdHJpY3Rpb25zIG9uIGVhY2ggcHJveHkgcXBz",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
 }
]
}
```



```

 },
 {
 "confName": "rqpsquota",
 "confDefault": "100000",
 "confValue": null,
 "confType": 2,
 "confRange": "0-500000",
 "confModule": 2,
 "confDesc": "cXVvdGEgb24gZWJjaCBwcm94eSBxcHMgb2YgcmVhZA==",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
 },
 {
 "confName": "wqpsquota",
 "confDefault": "100000",
 "confValue": null,
 "confType": 2,
 "confRange": "0-500000",
 "confModule": 2,
 "confDesc": "cXVvdGEgb24gZWJjaCBwcm94eSBxcHMgb2Ygd3JpdGU=",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
 },
 {
 "confName": "stale_slave_readable",
 "confDefault": "false",
 "confValue": null,
 "confType": 2,
 "confRange": "true|false",
 "confModule": 2,
 "confDesc": "c3RhbmGUgcmVhZGFibGUgb24gcHJveHk=",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
 },
 {
 "confName": "appendonly",
 "confDefault": "yes",
 "confValue": null,
 "confType": 1,
 "confRange": "yes|no|partial",
 "confModule": 1,
 "confDesc": "5piv5ZCm5ZCv55SoIEFPRiDmjIHkuYXljY=",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "maxmemory-policy",
 "confDefault": "volatile-ttl",
 "confValue": null,
 "confType": 1,
 "confRange": "volatile-lru|allkeys-lru|volatile-random|allkeys-random|volatile-ttl|volatile-lfu|allkeys-lfu|noeviction",
 "confModule": 1,
 "confDesc": "5b2T5YaF5a2Y6LaF6YeP5pe255qE6ZSu5reY5rGw562W55Wl44CC5Y+v6YCYJ0jxici8+dm9sYXRpbGUtbHJ177ya5Y+q5LuO5t",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 }
],

```

```

 },
 {
 "confName": "hash-max-ziplist-entries",
 "confDefault": "512",
 "confValue": null,
 "confType": 2,
 "confRange": "0-999999999999999",
 "confModule": 1,
 "confDesc":
"5b2TIGhhc2gg5a+56LGh5ZCM5pe25ruh6Laz5Lul5LiL5Lik5Liq5p2h5Lu25pe277yMIGhhc2gg5a+56LGh5L2/55SolHppcGxp
",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "hash-max-ziplist-value",
 "confDefault": "64",
 "confValue": null,
 "confType": 2,
 "confRange": "0-999999999999999",
 "confModule": 1,
 "confDesc":
"5b2TIGhhc2gg5a+56LGh5ZCM5pe25ruh6Laz5Lul5LiL5Lik5Liq5p2h5Lu25pe277yMIGhhc2gg5a+56LGh5L2/55SolHppcGxp
",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "set-max-intset-entries",
 "confDefault": "512",
 "confValue": null,
 "confType": 2,
 "confRange": "0-999999999999999",
 "confModule": 1,
 "confDesc":
"5b2TIHNldCDlr7nosaHIklzml7bmu6HotrPku6XkulvkuKTkuKrmnaHku7bml7bwlwgc2VOIOWvueixoeS9v+eUqCBpbnRzZXQg57y
",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "zset-max-ziplist-entries",
 "confDefault": "128",
 "confValue": null,
 "confType": 2,
 "confRange": "0-999999999999999",
 "confModule": 1,
 "confDesc":
"5b2TIHpzZXQg5a+56LGh5ZCM5pe25ruh6Laz5Lul5LiL5Lik5Liq5p2h5Lu25pe277yMenNldCDlr7nosaHkvb/nIKggemlwbGlzdC
",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "zset-max-ziplist-value",
 "confDefault": "64",
 "confValue": null,
 "confType": 2,
 "confRange": "0-999999999999999",

```

```

 "confModule": 1,
 "confDesc":
"5b2TIHpzZXQg5a+56LGh5ZCM5pe25ruh6Laz5Lu5LiL5Lik5Liq5p2h5Lu25pe277yMenNldCDlr7nosaHkvb/nlKggemlwbGlzdC
 ,
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "notify-keyspace-events",
 "confDefault": "",
 "confValue": null,
 "confType": 4,
 "confRange": "K|E|g|$||s|h|z|x|e|A",
 "confModule": 1,
 "confDesc":
"5a6i5oi356uv5Y+v5Lu5Yip55So6ZSu56m66Ze06YCa55+I5p2I5o6I5pS26YKj5Lqb5b2x5ZONIFJIZGlzIOaVsOaNrumbhueahC
 ,
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 7001
 },
 {
 "confName": "timeout",
 "confDefault": "0",
 "confValue": null,
 "confType": 2,
 "confRange": "0-100000",
 "confModule": 1,
 "confDesc":
"5pat5byA56m66Zey5a6i5oi356uv5LmL5YmN562J5b6F55qE56eS5pWw44CC6Zu25YC86KGo56S65Lu05LiN5pat5byA56m
 ,
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "hz",
 "confDefault": "10",
 "confValue": null,
 "confType": 2,
 "confRange": "1-500",
 "confModule": 1,
 "confDesc":
"6K6+572ulFJIZGlzIOWQjuWPoS7u+WKoeaJp+ihjOmikee0h++8jOavlOWmgua4hemZpOi/h+acn+mUruS7u+WKoeOAguWPgu
 ,
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 {
 "confName": "disable_commands",
 "confDefault": "flushall,flushdb",
 "confValue": null,
 "confType": 3,
 "confRange": "flushall|flushdb|keys|hgetall",
 "confModule": 1,
 "confDesc": "55So5oi36leq5a6a5LmJ56aB55So5ZG95Luk",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 10001
 },
 },

```

```
{
 "confName": "support_multi_active",
 "confDefault": "no",
 "confValue": null,
 "confType": 1,
 "confRange": "yes|no",
 "confModule": 1,
 "confDesc": "bXVsdGkgYWN0aXZIIG9wdGlrbG==",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
},
{
 "confName": "user_filter_cmds",
 "confDefault": "",
 "confValue": null,
 "confType": 4,
 "confRange": "set|del",
 "confModule": 1,
 "confDesc": "dXNlciBmaWx0ZXIyY21kcw==",
 "needReboot": 0,
 "confRedisVersion": "all",
 "confCacheVersion": 5001
},
{
 "confName": "use-op-header",
 "confDefault": "no",
 "confValue": null,
 "confType": 2,
 "confRange": "yes|no",
 "confModule": 1,
 "confDesc": "dXNlIG9wLWhiYWRIcG==",
 "needReboot": 0,
 "confRedisVersion": "4.0",
 "confCacheVersion": 10001
}
]
```

获取应用参数模版记录

请求说明

获取应用参数模版记录。

请求结构

```
GET /v2/template/record/{templateShowId}?marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
templateShowId	String	是	URL参数	参数模版ID
marker	String	否	URL参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	int	否	URL参数	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	类型	描述
result	List<Result>	参数模版列表
marker	String	批量获取列表的查询的起始位置，是一个由系统生成的字符串,起始值可传入-1
maxKeys	int	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000
isTruncated	Boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现

Result参数信息

参数名称	类型	描述
cacheClusterShowId	String	集群ID
cacheClusterName	String	集群名称
availabilityZone	String	可用区
version	Integer	集群版本
status	String	集群状态
engine	String	引擎类型
engineVersion	String	引擎版本
clusterType	String	集群类型
createTime	String	创建时间
applyTime	String	应用时间

请求示例

```
GET /v2/template/record/scs-tmpl-zuiflgqnmwrx?marker=-1&maxKeys=20 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
{
 "result": [{
 "cacheClusterShowId": "scs-bj-cxisufltkqu1",
 "cacheClusterName": "scs_test",
 "availabilityZone": "zoneA",
 "version": 5001,
 "status": "modifying",
 "engine": "redis",
 "engineVersion": "3.2",
 "clusterType": "cluster",
 "createTime": "2022-09-15T06:48:36Z",
 "applyTime": "2022-09-15T06:48:36Z",
 }],
 "isTruncated": true,
 "nextMarker": "scs-bj-cxisufltkquj",
 "maxKeys": 10,
 "marker": "-1"
}
```

## 日志管理

### 审计日志开关API

#### 请求说明

- 该 API 对应的控制台操作为开启或关闭**审计日志**。
- 审计日志仅**Redis内存型4.0 6.0 7.0**版本支持。

#### 请求结构

```
PUT /v1/instance/{instanceId}/log/audit/switch HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

#### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
action	String	Body	是	开关动作。open（开启）、close（关闭）

#### 返回头域

除公共头部，无其它特殊头部。

#### 返回参数

无特殊返回参数。

#### 请求示例

```
PUT /v1/instance/scs-bj-oqdgnbjfogjw/log/audit/switch HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "action": "close"
}
```

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
```

### 日志明细API

#### 请求说明

- 查询实例的某一个日志的明细信息。
- 日志详情仅Redis内存型支持。

#### 请求结构

```
GET /v1/instance/{instanceId}/log/{logId} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

#### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
logId	String	Path	是	日志ID
validSeconds	String	Query	否	下载有效时间，单位s。默认值1800。

#### 返回头域

除公共头部，无其它特殊头部。

#### 返回参数

参数名称	类型	描述
logId	String	日志ID
downloadUrl	String	日志下载链接
downloadExpire	String	下载链接到期时间

#### 请求示例

```
GET /v1/instance/scs-bj-oqdgmbjfojiw/log/scs-bj-oqdgmbjfojiw_67140_runlog_202410111430
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "logId": "scs-bj-oqdgmbjfojiw_67140_runlog_202410111430",
 "downloadUrl": "https://bj.bcebos.com/scs-log-download-bj/scs-bj-oqdgmbjfojiw_67140_runlog_202410111430?authorization=bce-auth-v1%2F91884b9e79a442f29f8b41106d060e72%2F2024-10-12T10%3A58%3A56Z%2F1800%2Fhost%2F6689fc40ebad827fe774d47e510df6c461ae2bf649598f49de89ea83d398e1ec",
 "downloadExpires": "2024-10-12 19:28:56"
}
```

### 日志列表API

请求说明

- 该 API 对应的控制台操作为日志列表。 您可以查看7天内的运行日志、慢日志、审计日志，寻找解决运维问题的线索。
- 审计日志仅Redis 4.0 6.0 7.0版本支持。
- 日志列表仅Redis内存型支持。

请求结构

```
GET /v1/instance/{instanceId}/log/list?fileType=runlog&startTime={startTime} HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
```

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
fileType	String	Query	是	日志类型，取值范围：    - slowlog（慢日志下载列表）   - runlog（运行日志下载列表）   - auditlog（审计列表）
startTime	String	Query	是	开始时间，格式 "yyyy-MM-dd hh:mm:ss"
endTime	String	Query	否	结束时间，格式 "yyyy-MM-dd hh:mm:ss"     未填写该参数，将默认返回开始时间+24小时内的日志。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	类型	描述
logList	List<ShardLog>	日志列表

ShardLog 分片日志信息

参数名称	类型	描述
shardShowId	String	分片ID
totalNum	Integer	总数
logItem	List<LogItem>	分片日志列表
shardId	Integer	分片数字ID

LogItem 单个日志记录

参数名称	类型	描述
logId	String	日志记录ID
logSizeInBytes	Integer	日志文件大小(单位为字节)
logStartTime	String	日志起始时间点
logEndTime	String	日志结束时间点
downloadUrl	String	日志下载链接
downloadExpires	String	下载链接到期时间



🔗 请求示例

```
GET /v1/instance/scs-bj-oqdgnbjfogjw/log/list?fileType=runlog&startTime=2024-10-11 00:00:00 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

🔗 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "logList": [
 {
 "shardId": 67140,
 "shardShowId": "scs-bj-oqdgnbjfogjw-0",
 "totalNum": 4,
 "logItem": [
 {
 "logId": "scs-bj-oqdgnbjfogjw_67140_runlog_202410111430",
 "logSizeInBytes": 35746,
 "logStartTime": "2024-10-11 14:00:00",
 "logEndTime": "2024-10-11 14:30:00",
 "downloadUrl": "https://bj.bcebos.com/scs-log-download-bj/scs-bj-oqdgnbjfogjw_67140_runlog_202410111430?authorization=bce-auth-v1%2F91884b9e79a442f29f8b41106d060e72%2F2024-10-12T10%3A52%3A25Z%2F1800%2Fhost%2Fad748418009dbce23bb73e7adc8a3f5e09d830eb1a6a69f5047cc14cf26dfcd5",
 "downloadExpires": "2024-10-12 19:22:25"
 }
]
 }
]
}
```

安全管理

TDE加密

🔗 TDE加密

接口描述

本接口用于开启TDE加密。

本接口为v1接口。

注意事项

- 支持Redis标准版6.0集群
- 热活组中的实例不支持此操作，并且开启后的实例不支持加入热活组中
- 开启TDE加密后不支持关闭
- 开启TDE加密后历史备份不支持恢复
- 开启TDE加密后不支持大key分析功能

请求结构

```
POST /v{version}/instance/{instanceId}/tde HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: authorization string
{
 "action": "open"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	类型	是否必选	参数位置	描述
version	String	是	PATH参数	API版本号。此处值为1。
instanceId	String	是	PATH参数	集群ID。
action	String	是	RequestBody参数	开启TDE加密。参数可选值：open

响应头域

除公共头域外，无其它特殊头域。

响应参数

参数名称	类型	描述
operateStatus	String	操作状态。仅成功时返回success
errorMessage	String	操作失败原因。该值为空，无实际意义。

请求示例

```
POST http://redis.bj.baidubce.com/v1/instance/scs-bj-bmaccxilppn/tde
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "action": "open"
}
```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "operateStatus": "success",
 "errorMessage": ""
}
```

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "requestId": "a43bf34f-0c20-4a63-8ec7-d14a369d0f64",
 "code": "ClusterStatusForbidden",
 "message": "The cluster status forbidden operation."
}
```

查看安全组API

请求说明

通过调用API来查看Redis设置的安全组。

请求结构

```
GET /v1/instance/{instanceId}/securityGroup HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID

返回头域

除公共头域，无其它特殊头域。

返回参数

参数名称	参数类型	描述
groups	List<Group>	安全组列表。
activeRules	List<Rule>	安全组规则列表。

Group

参数名称	参数类型	描述
securityGroupRemark	String	安全组备注。
securityGroupName	String	安全组名称。
securityGroupId	String	安全组ID。
securityGroupUuid	String	安全组长ID。
vpclId	String	vpclId。
vpcName	String	vpc名称。
outbound	List<Rule>	安全组规则。

Rule

参数名称	参数类型	描述
id	String	安全组规则ID。
securityGroupRuleId	String	安全组规则ID。
securityGroupId	String	安全组ID。
securityGroupUuid	String	安全组长ID。
direction	String	入站/出站，取值ingress/Ingress或egress/Egress。
ethertype	String	网络类型，取值IPv4或IPv6。值为空时表示默认取值IPv4。
protocol	String	协议类型，tcp、udp或icmp，值为空时默认取值all。
portRange	String	端口范围，可以指定80等单个端口，值为空时默认取值1-65535。
remoteGroupId	String	源安全组ID。
remoteGroupName	String	源安全组名称。
remoteIP	String	源IP地址。
name	String	安全组规则名称。

请求示例

```
GET /v1/instance/scs-bdbl-ynqbwopdesj/securityGroup HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2017-11-21T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "groups": [
 {
 "securityGroupRemark": "default",
 "securityGroupName": "默认安全组",
 "securityGroupId": "g-0qdtc2bzp0gd",
 "securityGroupUuid": "a825fdcd-9533-4bc2-8084-05a01952ceb7",
 "vpclId": "vpc-e20k5jh5afv2",
 "vpcName": "默认私有网络",
 "outbound": [
 {
 "tenantId": "d443fbaf17e044d587701d7f4fdb9bd6",
 "id": "r-lz1iqYEG",
 "securityGroupRuleId": "r-lz1iqYEG",
 "securityGroupId": "g-0qdtc2bzp0gd",
 "securityGroupUuid": "a825fdcd-9533-4bc2-8084-05a01952ceb7",
 "direction": "Egress",
 "ethertype": "IPv4",
 "protocol": "",
 "portRange": "",
 "remoteGroupId": "",
 "remoteIP": "",
 "name": ""
 }
],
 "inbound": [
 {
 "tenantId": "d443fbaf17e044d587701d7f4fdb9bd6",
```

```
 "id": "r-f1y3ruc1rzrb",
 "securityGroupRuleId": "r-f1y3ruc1rzrb",
 "securityGroupId": "g-0qdtc2bzp0gd",
 "securityGroupUuid": "a825fdcd-9533-4bc2-8084-05a01952ceb7",
 "direction": "Ingress",
 "ethertype": "IPv6",
 "protocol": "",
 "portRange": "",
 "remoteGroupId": "",
 "remoteIP": "",
 "name": ""
 }
],
"projectId": "scs-bdbl-ynqvbwopdesj"
}
],
"activeRules": [
{
 "tenantId": "d443fbaf17e044d587701d7f4fdb9bd6",
 "id": "r-r2qdz2sax6w1",
 "securityGroupRuleId": "r-r2qdz2sax6w1",
 "securityGroupId": "g-0qdtc2bzp0gd",
 "securityGroupUuid": "a825fdcd-9533-4bc2-8084-05a01952ceb7",
 "direction": "egress",
 "ethertype": "IPv6",
 "protocol": "",
 "portRange": "",
 "remoteGroupId": "",
 "remoteIP": "",
 "name": ""
}
]
}
```

解绑安全组API

请求说明

通过调用API来解绑Redis当前设置的安全组。

请求结构

```
POST /v1/instance/{instanceId}/securityGroup/unbind HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityGroupIds": [
 "g-0qdtc2bzp0gd"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
securityGroupIds	List<String>	Body	是	解绑的安全组ID列表。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
success	Boolean	是否成功。

请求示例

```
POST /v1/instance/scs-bdbl-ynqvbwopdesj/securityGroup/unbind HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "securityGroupIds": [
 "g-0qdtc2bzp0gd"
]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true
}
```

绑定安全组API

请求说明

通过调用API来绑定Redis的安全组。 不支持绑定已经绑定的安全组。

请求结构

```
POST /v1/instance/{instanceId}/securityGroup/bind HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityGroupIds": [
 "g-0qdtc2bzp0gd"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
securityGroupIds	List<String>	Body	是	绑定的安全组ID列表。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
success	Boolean	是否成功。

请求示例

```
POST /v1/instance/scs-bdbl-ynqvwopdesj/securityGroup/bind HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "securityGroupIds": [
 "g-0qdtc2bzp0gd"
]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true
}
```

## 更新安全组API

请求说明

通过调用API来更新Redis的安全组。更新后会替换之前所有已经绑定的安全组为新的安全组。

请求结构

```
POST /v1/instance/{instanceId}/securityGroup/update HTTP/1.1
HOST: redis.{region}.baidubce.com
Authorization: authorization string
{
 "securityGroupIds": [
 "g-0qdtc2bzp0gd"
]
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
securityGroupIds	List<String>	Body	是	绑定的安全组ID列表。

返回头域

除公共头域外，无其它特殊头域。

返回参数

参数名称	参数类型	描述
success	Boolean	是否成功。

请求示例

```
POST /v1/instance/scs-bdbl-ynqvwopdesj/securityGroup/bind HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "securityGroupIds": [
 "g-0qdtc2bzb0gd"
]
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "success": true
}
```

获取TLS证书API

请求说明

通过调用API来获取Redis的TLS证书。

请求结构

```
GET /v1/instance/{instanceId}/tls
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
status	String	ca证书状态：tcreate（创建中）、inuse（使用中）、todelete（删除中）、deleted（已删除、已失效）
createTime	String	创建时间。
expireTime	String	过期时间。
ca	String	ca证书。

请求示例



```
GET /v1/instance/scs-bj-elahimuhxjia/tls HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "status": "inuse",
 "createTime": "2024-10-25T03:28:16Z",
 "expireTime": "2025-10-25T03:28:16Z",
 "ca": "-----BEGIN CERTIFICATE-----
\nMIIEnTCCAoWgAwIBAgICBVAwDQYJKoZIhvcNAQELBQAwADAEfwOyNDEwMjUwMzI4\nMTZaFw00NDEwMjUwMzI4MTZaMAAw
-----END CERTIFICATE-----\n"
}
```

开启或关闭TLS加密API

请求说明

通过调用API来开启Redis的TLS访问。开启TLS访问，将会重启实例，请在低峰期操作。

请求结构

```
POST /v1/instance/{instanceId}/tls HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "action": "open"
}
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
instanceId	String	Path	是	集群ID
action	String	Body	是	open：开启；close：关闭；update：更新证书有效期。开启加密之后才能进行更新。

返回头域

除公共头域外，无其它特殊头域。

返回参数

无。

请求示例

```
POST /v1/instance/scs-bj-elahimuhxjia/tls HTTP/1.1
HOST: redis.bj.baidubce.com
ContentType: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c63bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "action": "open"
}
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
```

## 附录

Model 对象定义

InstanceModel

参数名称	类型	描述
instanceId	String	实例ID
instanceName	String	实例名
instanceStatus	String	<a href="#">实例状态</a>
isolatedStatus	String	已隔离:Isolated(实例在回收站内)
clusterType	String	引擎类型为Redis，取值：集群：“cluster” 主从：“master_slave”，default: 主从
engine	String	引擎类型，redis,memcache
engineVersion	String	引擎版本
vnetIp	String	私网IP
domain	String	域名
port	String	链接端口
instanceCreateTime	Datetime	实例创建时间
instanceExpireTime	Datetime	实例到期时间
capacity	Float	总容量，单位GB
usedCapacity	double	已用容量，单位GB
paymentTiming	String	付费方式。预付费：Prepaid，后付费：Postpaid
zoneNames	List<String>	可用区list
orderStatus	String	预转后、后转预的标记。    - to_postpay: 预付费转后付费   - to_prepay: 后付费转预付费
deployIdList	List<String>	集群绑定的部署集ID列表。

状态编码定义

实例状态

状态值	描述
Creating	创建中
Running	运行中
Rebooting	实例重启中
Pausing	暂停中
Paused	已欠费
Deleted	已释放
Deleting	释放中
Failed	已失败
Modifying	配置变更中
Modifyfailed	配置变更失败
Backuping	备份中
Aztransforming	转换可用区
Expire	已到期
Flushing	清空实例
Flush failed	清空失败

实例规格

计费项	存储类型	规格	用户可用内存（GB）	支持分片数量
缓存节点	高性能内存型	cache.n1.micro	1	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	高性能内存型	cache.n1.small	2	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	高性能内存型	cache.n1.medium	4	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	高性能内存型	cache.n1.large	8	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	高性能内存型	cache.n1.xlarge	16	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	高性能内存型	cache.n1.2xlarge	32	N/A
缓存节点	高性能内存型	cache.n1.4xlarge	64	N/A
缓存节点	分级存储型	cache.n1hs3.2xlarge	32	2/4/6/8/12/16/24/32/48/64/96/128
缓存节点	分级存储型	cache.n1hs3.4xlarge	64	2/4/6/8/12/16/24/32/48/64/96/128
代理节点	N/A	cache.proxy	N/A	N/A

订单信息定义

Billing

状态	类型	描述
paymentTiming	String	付费方式： 预付费：Prepaid，后付费：Postpaid
reservation	Reservation	保留信息，支付方式为后支付时不需要设置，预支付时必须设置

Reservation

状态	类型	描述
reservationLength	int	时长，[1,2,3,4,5,6,7,8,9,12,24,36]
reservationTimeUnit	String	时间单位，Month，当前仅支持按月

部署集

## 创建部署集API

### 请求说明

通过调用API来创建部署集。

### 请求结构

```
POST /v1/deploySet HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
{
 "name": "clltest11",
 "strategy": "HOST_HA",
 "concurrency": 2
}
```

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	参数类型	参数位置	是否必须	描述
name	String	Body	是	部署集名称。
desc	String	Body	否	备注。
strategy	String	Body	是	策略。默认值：HOST_HA。
concurrency	Integer	Body	是	并发度。取值范围：1-127。默认值：20。

### 返回头域

除公共头部，无其它特殊头部。

### 返回参数

参数名称	参数类型	描述
deploySetId	String	部署集ID。

### 请求示例

```
POST /v1/deploySet HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863fffac5190fdec1ec24b7c03eb5d67d2e1de
{
 "name": "clltest11",
 "desc": "clltest11",
 "strategy": "HOST_HA",
 "concurrency": 2
}
```

### 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Wed, 08 Jul 2015 03:28:11 GMT
x-bce-request-id: d8752367-38e8-45e4-b4c7-e53be3137ce5
Server: BWS
{
 "deploySetId": "dset-72f5618f-f976-4d35-8fe4-1618ab6fec03"
}
```

## 删除部署集API

请求说明

通过调用API删除部署集。 有绑定的实例，不支持删除。

请求结构

```
DELETE v1/deploySet/{deploySetId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
deploySetId	String	Path	是	部署集ID。

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
DELETE /v1/deploySet/dset-cb0ce309-ae1-4309-b041-85532fe36738 HTTP/1.1
Host: redis.bj.baidubce.com
Authorization: bce-auth-v1/f81d3b34e48048fbb2634dc7882d7e21/2015-08-11T04:17:29Z/3600/host/74c506f68c65e26c633bfa104c863ffac5190fdec1ec24b7c03eb5d67d2e1de
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 1214cca7-4ad5-451d-9215-71cb844c0a50
Content-Type: application/json;charset=UTF-8
```

## 修改部署集API

请求说明

通过调用API修改部署集。

请求结构

```
PUT v1/deploySet/{deploySetId} HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
deploySetId	String	Path	是	部署集ID。
desc	String	Body	否	备注。
name	String	Body	是	部署集名称。
concurrency	Integer	Body	是	并发度。取值范围：1-127。默认值：20。

返回头域

除公共头部，无其它特殊头部。

返回参数

无特殊返回参数。

请求示例

```
PUT /v1/deploySet/dset-cb0ce309-aee1-4309-b041-85532fe36738 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
{
 "name": "clltest9",
 "concurrency": 20,
 "desc": "clltest91"
}
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 1214cca7-4ad5-451d-9215-71cb844c0a50
Content-Type: application/json;charset=UTF-8
```

获取部署集列表API

请求说明

通过调用API来获取部署集列表。

请求结构

```
GET /v1/deploySet?maxKeys=1&marker=-1 HTTP/1.1
Host: redis.{region}.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其它特殊头域。

请求参数

参数名称	参数类型	参数位置	是否必须	描述
marker	String	Query	否	批量获取列表的查询的起始位置，是一个由系统生成的字符串。默认-1。
maxKeys	Integer	Query	否	每页包含的最大数量(主实例)，最大数量通常不超过1000，缺省值为1000。

返回头域

除公共头部，无其它特殊头部。

返回参数

参数名称	参数类型	描述
result	List<Deploy>	部署集列表。
marker	String	批量获取列表的查询的起始位置，是一个由系统生成的字符串,起始值可传入-1。
isTruncated	Boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现。
maxKeys	Integer	每页包含的最大数量，最大数量通常不超过1000，缺省值为1000。

Deploy

参数名称	参数类型	描述
scsInstanceCount	Integer	部署集绑定的实例数量。
scsInstancelds	List<String>	部署集绑定的SCS集群ID列表。
name	String	部署集名称。
strategy	String	策略。默认值：HOST_HA。
concurrency	Integer	并发度。
deploySetId	String	部署集ID。
desc	String	备注。

请求示例

```
GET /v1/deploySet?maxKeys=1&marker=-1 HTTP/1.1
HOST: redis.bj.baidubce.com
Content-Type: application/json
Authorization: bce-auth-v1/0255cb4f78f140728045b551d6a317ec/2018-11-13T05:46:05Z/1800/host/70268ec1d0491c86e67f345ad3aee4cf7f8924dade9b7f3e2f1a95814a00c129
```

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
{
 "result": [
 {
 "scsInstanceCount": 0,
 "name": "ceshi",
 "strategy": "HOST_HA",
 "concurrency": 1,
 "deploySetId": "dset-52aaf380-dd37-47a2-b999-70af061a5118",
 "desc": "for_test"
 }
],
 "marker": "-1",
 "isTruncated": true,
 "nextMarker": "dset-3978b624-a9b7-45b8-8a40-3fc107686c35",
 "maxKeys": 1
}
```

# SDK

## GO-SDK

### 概述

本文档主要介绍Redis GO SDK的使用。在使用本文档前，您需要先了解Redis的一些基本知识。若您还不了解Redis，可以参考[产品描述](#)和[入门指南](#)。

### 安装SDK工具包

运行环境

GO SDK可以在go1.3及以上环境下运行。

安装SDK

直接从github下载

使用go get工具从github进行下载：

```
go get github.com/baidubce/bce-sdk-go
```

SDK目录结构

```
bce-sdk-go
|--auth //BCE签名和权限认证
|--bce //BCE公用基础组件
|--http //BCE的http通信模块
|--services //BCE相关服务目录
| |--scs //Redis
|--util //BCE公用的工具实现
```

🔗 卸载SDK

预期卸载SDK时，删除下载的源码即可。

初始化

🔗 确认Endpoint

在确认您使用SDK时配置的Endpoint时，可先阅读开发人员指南中关于[Redis访问域名](#)的部分，理解Endpoint相关的概念。百度云目前开放了多区域支持，请参考[区域选择说明](#)。

🔗 获取密钥

要使用百度云Redis，您需要拥有一个有效的AK(Access Key ID)和SK(Secret Access Key)用来进行签名认证。AK/SK是由系统分配给用户的，均为字符串，用于标识用户，为访问Redis做签名验证。

可以通过如下步骤获得并了解您的AK/SK信息：

[注册百度云账号](#)

[创建AK/SK](#)

🔗 新建Redis Client

Redis Client是Redis服务的客户端，为开发者与Redis服务进行交互提供了一系列的方法。

使用AK/SK新建Redis Client

通过AK/SK方式访问Redis，用户可以参考如下代码新建一个Redis Client：

```
import (
 "github.com/baidubce/bce-sdk-go/services/scs"
)

func main() {
 // 用户的Access Key ID和Secret Access Key
 ACCESS_KEY_ID, SECRET_ACCESS_KEY := <your-access-key-id>, <your-secret-access-key>

 // 用户指定的Endpoint
 ENDPOINT := <domain-name>

 // 初始化一个RedisClient
 scsClient, err := scs.NewClient(AK, SK, ENDPOINT)
}
```

在上面代码中，ACCESS\_KEY\_ID对应控制台中的“Access Key ID”，SECRET\_ACCESS\_KEY对应控制台中的“Access Key Secret”，获取方式请参考《操作指南 [管理ACCESSKEY](#)》。第三个参数ENDPOINT支持用户自己指定域名，如果设置为空字符串，会使用默认域名作为Redis的服务地址。

**注意：**ENDPOINT参数需要用指定区域的域名来进行定义，如服务所在区域为北京，则为redis.bj.baidubce.com。

使用STS创建Redis Client



## 申请STS token

Redis可以通过STS机制实现第三方的临时授权访问。STS (Security Token Service) 是百度云提供的临时授权服务。通过STS, 您可以为第三方用户颁发一个自定义时效和权限的访问凭证。第三方用户可以使用该访问凭证直接调用百度的API或SDK访问百度云资源。

通过STS方式访问Redis, 用户需要先通过STS的client申请一个认证字符串, 申请方式可参见[百度云STS使用介绍](#)。

## 用STS token新建Redis Client

申请好STS后, 可将STS Token配置到Redis Client中, 从而实现通过STS Token创建Redis Client。

## 代码示例

GO SDK实现了STS服务的接口, 用户可以参考如下完整代码, 实现申请STS Token和创建Redis Client对象:

```
import (
 "fmt"

 "github.com/baidubce/bce-sdk-go/auth" //导入认证模块
 "github.com/baidubce/bce-sdk-go/services/scs" //导入Redis服务模块
 "github.com/baidubce/bce-sdk-go/services/sts" //导入STS服务模块
)

func main() {
 // 创建STS服务的Client对象, Endpoint使用默认值
 AK, SK := <your-access-key-id>, <your-secret-access-key>
 stsClient, err := sts.NewClient(AK, SK)
 if err != nil {
 fmt.Println("create sts client object :", err)
 return
 }

 // 获取临时认证token, 有效期为60秒, ACL为空
 stsObj, err := stsClient.GetSessionToken(60, "")
 if err != nil {
 fmt.Println("get session token failed:", err)
 return
 }
 fmt.Println("GetSessionToken result:")
 fmt.Println(" accessKeyId:", stsObj.AccessKeyId)
 fmt.Println(" secretAccessKey:", stsObj.SecretAccessKey)
 fmt.Println(" sessionToken:", stsObj.SessionToken)
 fmt.Println(" createTime:", stsObj.CreateTime)
 fmt.Println(" expiration:", stsObj.Expiration)
 fmt.Println(" userId:", stsObj.UserId)

 // 使用申请的临时STS创建Redis服务的Client对象, Endpoint使用默认值
 scsClient, err := scs.NewClient(stsObj.AccessKeyId, stsObj.SecretAccessKey, "")
 if err != nil {
 fmt.Println("create scs client failed:", err)
 return
 }
 stsCredential, err := auth.NewSessionBceCredentials(
 stsObj.AccessKeyId,
 stsObj.SecretAccessKey,
 stsObj.SessionToken)
 if err != nil {
 fmt.Println("create sts credential object failed:", err)
 return
 }
 scsClient.Config.Credentials = stsCredential
}
```

注意：目前使用STS配置Redis Client时，无论对应Redis服务的Endpoint在哪里，STS的Endpoint都需配置为<http://sts.bj.baidubce.com>。上述代码中创建STS对象时使用此默认值。

## 配置HTTPS协议访问Redis

Redis支持HTTPS传输协议，您可以通过在创建Redis Client对象时指定的Endpoint中指明HTTPS的方式，在Redis GO SDK中使用HTTPS访问Redis服务：

```
// import "github.com/baidubce/bce-sdk-go/services/scs"

ENDPOINT := "https://redis.bj.baidubce.com" //指明使用HTTPS协议
AK, SK := <your-access-key-id>, <your-secret-access-key>
scsClient, _ := scs.NewClient(AK, SK, ENDPOINT)
```

## 配置Redis Client

如果用户需要配置Redis Client的一些细节的参数，可以在创建Redis Client对象之后，使用该对象的导出字段Config进行自定义配置，可以为客户端配置代理，最大连接数等参数。

### 使用代理

下面一段代码可以让客户端使用代理访问Redis服务：

```
// import "github.com/baidubce/bce-sdk-go/services/scs"

//创建Redis Client对象
AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "redis.bj.baidubce.com"
client, _ := scs.NewClient(AK, SK, ENDPOINT)

//代理使用本地的8080端口
client.Config.ProxyUrl = "127.0.0.1:8080"
```

### 设置网络参数

用户可以通过如下的示例代码进行网络参数的设置：

```
// import "github.com/baidubce/bce-sdk-go/services/scs"

AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "redis.bj.baidubce.com"
client, _ := scs.NewClient(AK, SK, ENDPOINT)

// 配置不进行重试，默认为Back Off重试
client.Config.Retry = bce.NewNoRetryPolicy()

// 配置连接超时时间为30秒
client.Config.ConnectionTimeoutInMillis = 30 * 1000
```

### 配置生成签名字符串选项

```
// import "github.com/baidubce/bce-sdk-go/services/scs"

AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "redis.bj.baidubce.com"
client, _ := scs.NewClient(AK, SK, ENDPOINT)

// 配置签名使用的HTTP请求头为`Host`
headersToSign := map[string]struct{"Host": struct{}}
client.Config.SignOption.HeadersToSign = HeadersToSign

// 配置签名的有效期为30秒
client.Config.SignOption.ExpireSeconds = 30
```

参数说明

用户使用GO SDK访问Redis时，创建的Redis Client对象的Config字段支持的所有参数如下表所示：

配置项名称	类型	含义
Endpoint	string	请求服务的域名
ProxyUrl	string	客户端请求的代理地址
Region	string	请求资源的区域
UserAgent	string	用户名称，HTTP请求的User-Agent头
Credentials	*auth.BceCredentials	请求的鉴权对象，分为普通AK/SK与STS两种
SignOption	*auth.SignOptions	认证字符串签名选项
Retry	RetryPolicy	连接重试策略
ConnectionTimeoutInMillis	int	连接超时时间，单位毫秒，默认20分钟

说明：

1. Credentials字段使用auth.NewBceCredentials与auth.NewSessionBceCredentials函数创建，默认使用前者，后者为使用STS鉴权时使用，详见“使用STS创建Redis Client”小节。
2. SignOption字段为生成签名字符串时的选项，详见下表说明：

名称	类型	含义
HeadersToSign	map[string]struct{}	生成签名字符串时使用的HTTP头
Timestamp	int64	生成的签名字符串中使用的时间戳，默认使用请求发送时的值
ExpireSeconds	int	签名字符串的有效期

其中，HeadersToSign默认为`Host`，`Content-Type`，`Content-Length`，`Content-MD5`；TimeStamp一般为零值，表示使用调用生成认证字符串时的时间戳，用户一般不应该明确指定该字段的值；ExpireSeconds默认为1800秒即30分钟。

3. Retry字段指定重试策略，目前支持两种：NoRetryPolicy和BackOffRetryPolicy。默认使用后者，该重试策略是指定最大重试次数、最长重试时间和重试基数，按照重试基数乘以2的指数级增长的方式进行重试，直到达到最大重试测试或者最长重试时间为止。

主要接口

云数据库 Redis（Simple Cache Service）提供稳定、高效以及高可扩展性的分布式缓存服务。云数据库 Redis 兼容 Redis/Memcached 协议，基于 Redis 提供标准版和集群版的架构模式，并支持自定义副本数量，为您提供多样化的数据结构支持。

☞ 订单服务

创建实例

使用以下代码可以创建Redis实例，用于创建一个或多个redis实例

```
args := &scs.CreateInstanceArgs{
 // 选择付款方式，可以选择预付费或后付费
 Billing: api.Billing{
 PaymentTiming: api.PaymentTimingPostPaid,
 },
 // 购买个数，最大不超过10，默认1
 PurchaseCount: 1,
 // 实例名
 // 要求：1) 支持大小写字母、数字以及_/.等特殊字符，必须以字母开头；2) 长度限制为1-64；
 InstanceName: "sdk-scs",
 // 端口号 1025<port<22222, 22222<port<65535
 Port: 6379,
 // 引擎版本，集群：3.2、4.0、5.0 主从：2.8、3.2、4.0、5.0
 EngineVersion: "3.2",
 // 节点规格
 NodeType: "cache.n1.micro",
 // 集群类型，集群版："cluster"，主从版："master_slave"
 ClusterType: "master_slave",
 // 副本个数，单副本为1，双副本为2，多副本依此类推
 ReplicationNum: 1,
 // 分片个数
 ShardNum: 1,
 // 代理节点数，主从版：0，集群版：代理节点数=分片个数
 ProxyNum: 0,
}

// 若要生成预付费实例，可以按以下配置生成一个月的预付费实例
args.Billing = api.Billing{
 PaymentTiming: api.PaymentTimingPrePaid,
 Reservation: &api.Reservation{
 ReservationLength: 1,
 ReservationTimeUnit: "month",
 }
}

// 若要设置自动续费，可以按以下参数设置一年内自动续费
// 按月付费或者按年付费 月是"month"，年是"year"
// 自动续费的时间 按月是1-9 按年是1-3
args.AutoRenewTimeUnit = "year"
args.AutoRenewTime = 1

result, err := client.CreateInstance(args)
if err != nil {
 fmt.Println("create instance failed:", err)
} else {
 fmt.Println("create instance success: ", result)
}
```

#### 提示：

1. 创建Redis请求是一个异步请求，返回200表明订单生成，后续可以通过实例id查询实例创建进度。
2. 本接口用于创建一个或多个同配置Redis实例。
3. 创建实例需要实名认证，没有通过实名认证的可以前往百度开放云官网控制台中的安全认证下的实名认证中进行认证。
4. 创建计费方式为后付费的实例需要账户现金余额+通用代金券大于100；预付费方式的实例则需要账户现金余额大于等于实例费用。

5. 支持批量创建，且如果创建过程中有一个实例创建失败，所有实例将全部回滚。
6. 创建请求详细使用请参考Redis API 文档[创建实例](#)
7. 创建Redis需要使用指定规格，详细请参考Redis API 文档[实例规格](#)

## 变更配置

```
args := &scs.ResizeInstanceArgs{
 NodeType: "cache.n1.small",
 ShardNum: 2,
}
err := client.ResizeInstance(instanceId, args)
if err != nil {
 fmt.Println("resize instance failed:", err)
} else {
 fmt.Println("resize instance success")
}
```

### 提示：

- 实例计费方式为预付费时，不能进行缩容操作
- 实例计费方式为后付费时，可弹性扩缩容
- 只有实例正常运行状态时才可以进行扩缩容操作，变更接口为异步变更，可通过查询指定实例详情接口查询实例状态

## 释放实例

如下代码可以释放实例,实例将自动进入回收站，保留7天后删除

```
err := client.DeleteInstance(instanceId)
if err != nil {
 fmt.Println("delete instance failed:", err)
} else {
 fmt.Println("delete instance success")
}
```

### 提示：

- 释放单个Redis实例，释放后实例将进入回收站，保留7天后自动删除所有物理资源。
- 可以从回收站内恢复实例或者彻底删除实例,预付费实例需要通过续费实例接口进行恢复。

## 实例服务

### 查询实例列表

以下代码可以查询Redis实例列表

```
args := &scs.ListInstancesArgs{}

result, err := client.ListInstances(args)
if err != nil {
 fmt.Println("list instance failed:", err)
} else {
 fmt.Println("list instance success: ", result)
}
```

### 查询指定实例详情

使用以下代码可以查询指定Redis虚机的详细信息

```
result, err := client.GetInstanceDetail(instanceId)
if err != nil {
 fmt.Println("get instance detail failed:", err)
} else {
 fmt.Println("get instance detail success ", result)
}
```

### 修改实例名称

如下代码可以修改实例名称

```
args := &scs.UpdateInstanceNameArgs{
 InstanceName: "newInstanceName",
}
err := client.UpdateInstanceName(instanceId, args)
if err != nil {
 fmt.Println("update instance name failed:", err)
} else {
 fmt.Println("update instance name success")
}
```

提示：

- 只有实例Running状态时可以修改实例名称

### 查询实例套餐规格

如下代码可以查询当前可以创建的实例的套餐的规格

```
result, err := client.GetNodeTypeList()
if err != nil {
 fmt.Println("list node type failed: ", err)
} else {
 fmt.Println("list node type success: ", result)
}
```

### 获取子网列表

如下代码可以获取子网列表

```
args := &scs.ListSubnetsArgs{}
_, err := client.ListSubnets(args)
if err != nil {
 fmt.Println("get subnet list failed:", err)
} else {
 fmt.Println("get subnet list success")
}
```

提示：

- 请求参数 vpId 和 zoneName 不是必须的

### 修改实例域名

如下代码可以修改实例域名

```
args := &scs.UpdateInstanceDomainNameArgs{
 Domain: "newDomainName",
}
err := client.UpdateInstanceDomainName(instanceId, args)
if err != nil {
 fmt.Println("update instance domain name failed:", err)
} else {
 fmt.Println("update instance domain name success")
}
```

提示：

- 只有实例Running状态时可以修改实例域名

### 获取可用区列表

使用以下代码可以获取可用区列表

```
result, err := client.GetZoneList()
if err != nil {
 fmt.Println("get zone list failed:", err)
} else {
 fmt.Println("get zone list success ", result)
}
```

### 清空实例

如下代码可以清空实例

```
args := &scs.FlushInstanceArgs{
 Password: "Password",
}
result, err := client.FlushInstance(instanceId, args)
if err != nil {
 fmt.Println("flush instance failed:", err)
} else {
 fmt.Println("flush instance success ", result)
}
```

提示：

- 对redis实例，清空后表现为数据占用内存下降，数据被清空；对memcache实例，清空后表现为占用内存不会下降，但是数据被清空
- 如果没有设置密码，传递空字符串

### 绑定标签

如下代码可以绑定标签

```
args := &scs.BindingTagArgs{
 ChangeTags: []model.TagModel{
 {
 TagKey: "tag1",
 TagValue: "var1",
 },
 },
}
result, err := client.BindingTag(instanceId, args)
if err != nil {
 fmt.Println("bind tags failed:", err)
} else {
 fmt.Println("bind tags success ", result)
}
```

### 解绑标签

如下代码可以解绑标签

```
args := &scs.BindingTagArgs{
 ChangeTags: []model.TagModel{
 {
 TagKey: "tag1",
 TagValue: "var1",
 },
 },
}
result, err := client.UnBindingTag(instanceId, args)
if err != nil {
 fmt.Println("unbind tags failed:", err)
} else {
 fmt.Println("unbind tags success ", result)
}
```

提示：

- 解绑实例上定义的标签
- 可以同时解绑多个标签

## 🔒 网络安全

### 修改访问密码

如下代码可以修改访问密码



```
args := &scs.ModifyPasswordArgs{
 Password: "newPassword",
}
result, err := client.ModifyPassword(instanceId, args)
if err != nil {
 fmt.Println("modify password failed:", err)
} else {
 fmt.Println("modify password success ", result)
}
```

提示：

- 密码长度8~16位，至少包含字母、数字和特殊字符中两种。允许的特殊字符包括 \$^\*( )\_+ =

### 查询IP白名单

如下代码可以查询允许访问实例的IP白名单

```
result, err := client.GetSecurityIp(instanceId)
if err != nil {
 fmt.Println("get security IP failed:", err)
} else {
 fmt.Println("get security IP success ", result)
}
```

提示：

- 返回参数 IP白名单列表, 包括常规地址: 如192.168.0.1 , CIDR地址: 如192.168.1.0/24 , 0.0.0.0/0代表允许所有地址

### 增加IP白名单

如下代码可以增加访问实例的IP白名单

```
args := &scs.SecurityIpArgs{
 SecurityIps: []string{
 "192.0.0.1",
 },
}
result, err := client.AddSecurityIp(instanceId, args)
if err != nil {
 fmt.Println("add security IP failed:", err)
} else {
 fmt.Println("add security IP success ", result)
}
```

### 删除IP白名单

如下代码可以删除访问实例的IP白名单

```
args := &scs.SecurityIpArgs{
 SecurityIps: []string{
 "192.0.0.1",
 },
}
result, err := client.DeleteSecurityIp(instanceId, args)
if err != nil {
 fmt.Println("delete security IP failed:", err)
} else {
 fmt.Println("delete security IP success ", result)
}
```

## 参数管理

### 获取参数列表

使用以下代码可以获取Redis实例的配置参数和运行参数

```
result, err := client.GetParameters(instanceId)
if err != nil {
 fmt.Println("get parameter list failed:", err)
} else {
 fmt.Println("get parameter list success ", result)
}
```

### 修改参数

如下代码可以修改redis实例参数值

```
args := &scs.ModifyParametersArgs{
 Parameter: InstanceParam{
 Name: "parameter name",
 Value: "new value",
 },
}
result, err := client.ModifyParameters(instanceId, args)
if err != nil {
 fmt.Println("modify parameters failed:", err)
} else {
 fmt.Println("modify parameters success ", result)
}
```

## 备份恢复

### 查看备份列表

使用以下代码可以查询某个实例备份列表

```
result, err := client.GetBackupList(instanceId)
if err != nil {
 fmt.Println("get backup list failed:", err)
} else {
 fmt.Println("get backup list success ", result)
}
```

### 修改备份策略

如下代码可以修改redis实例自动备份策略

```
args := &scs.ModifyBackupPolicyArgs{
 BackupDays: "Sun,Mon,Tue,Wed,Thu,Fri,Sta",
 BackupTime: "01:05:00",
 ExpireDay: 7,
}
result, err := client.ModifyBackupPolicy(instanceId, args)
if err != nil {
 fmt.Println("modify backup policy failed:", err)
} else {
 fmt.Println("modify backup policy success ", result)
}
```

#### 提示：

- BackupDays: 标识一周中哪几天进行备份备份周期：Mon（周一）Tue（周二）Wed（周三）Thu（周四）Fri（周五）Sat（周六）Sun（周日）逗号分隔，取值如：Sun,Wed,Thu,Fri,Sta
- BackupTime: 标识一天中何时进行备份，UTC时间（+8为北京时间）取值如：01:05:00
- ExpireDay: 备份文件过期时间，取值如：3

## 异常处理

GO语言以error类型标识错误，Redis支持两种错误见下表：

错误类型	说明
BceClientError	用户操作产生的错误
BceServiceError	Redis服务返回的错误

用户使用SDK调用Redis相关接口，除了返回所需的结果之外还会返回错误，用户可以获取相关错误进行处理。实例如下：

```
// scsClient 为已创建的Redis Client对象
instanceDetail, err := scsClient.GetInstanceDetail(instanceId)
if err != nil {
 switch realErr := err.(type) {
 case *bce.BceClientError:
 fmt.Println("client occurs error:", realErr.Error())
 case *bce.BceServiceError:
 fmt.Println("service occurs error:", realErr.Error())
 default:
 fmt.Println("unknown error:", err)
 }
} else {
 fmt.Println("get instance detail success: ", instanceDetail)
}
```

### 🔗 客户端异常

客户端异常表示客户端尝试向Redis发送请求以及数据传输时遇到的异常。例如，当发送请求时网络连接不可用时，则会返回BceClientError；当上传文件时发生IO异常时，也会抛出BceClientError。

### 🔗 服务端异常

当Redis服务端出现异常时，Redis服务端会返回给用户相应的错误信息，以便定位问题。常见服务端异常可参见[Redis错误返回](

### 🔗 SDK日志

GO SDK自行实现了支持六个级别、三种输出（标准输出、标准错误、文件）、基本格式设置的日志模块，导入路径为github.com/baidubce/bce-sdk-go/util/log。输出为文件时支持设置五种日志滚动方式（不滚动、按天、按小时、按分钟、按大

小)，此时还需设置输出日志文件的目录。

该日志模块无任何外部依赖，开发者使用GO SDK开发项目，可以直接引用该日志模块自行在项目中使用。可使用GO SDK使用的包级别的日志对象，也可创建新的日志对象，详见如下示例：

```
// 直接使用包级别全局日志对象（会和GO SDK自身日志一并输出）
log.SetLogHandler(log.STDERR)
log.Debugf("%s", "logging message using the log package in the sdk")

// 创建新的日志对象（依据自定义设置输出日志，与GO SDK日志输出分离）
myLogger := log.NewLogger()
myLogger.SetLogHandler(log.FILE)
myLogger.SetLogDir("/home/log")
myLogger.SetRotateType(log.ROTATE_SIZE)
myLogger.Info("this is my own logger from the sdk")
```

## Redis使用限制

### 背景信息

云数据库 Redis 服务兼容 Redis 2.6.16 版本的原生 API，详情请参见如下：

Redis：<http://redis.io/clients>。

Redis 是一个使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 的可开源的数据库，并提供多种语言的API。

自2017年8月9日起，Redis 版本升级为 3.2.8，所有新建的主从节点实例均使用新的 Redis 3.2.8版本，之前已经创建的 Redis 实例仍保持老版本，包括升降配操作也不受影响；集群实例不在此次升级之列，仍使用 Redis 2.6.16版本。

支持版本：Redis 2.6.16。

支持协议：redis protocol，兼容codis。

### 其他说明

- 对于多 Key 操作，使用 mget、mset、del 由于需要计算相关节点信息，延迟将会加大。以 Value 为 10 字节为例，set 操作大概是在 2ms 左右，10 个 key 的 mget 大概是在 5ms，100 个 Key 的 mget 大概是 20ms 左右。
- Key 最大长度为 1KB。
- Redis 限制要求 Value 的最大长度为 512MB。
- Redis 目前的淘汰策略是 volatile-ttl。

### Redis命令使用限制

#### 概述

本文档以 Redis 7.2 版本的[全部命令](#)为基准。

#### 主从节点实例与集群实例不支持的命令

- BGSAVE
- FLUSHDB
- FLUSHALL
- MIGRATE
- DEBUG
- BGREWRITEAOF

- LASTSAVE
- SAVE
- SYNC
- SLAVEOF
- SHUTDOWN
- ACL

🔗 PROXY命令支持情况概览

类别	支持情况
数据读写类命令	string、list、zset、hash、set、pubsub、geo、hyperloglog、stream、transaction 均支持
acl	访问控制列表通过控制台配置，仅支持 acl whoami 查看当前用户
config	不支持，可通过控制台变更配置
memory	仅支持 memory usage，集群内存使用情况，通过控制台查看
client	支持 clientid、info、kill、list、getname、setname、reply，其他不支持
cluster	因为 SCSS 是基于代理的架构和 redis cluster 在架构上有较大差异。     • cluster addslots、countkeysinslot、delslots、failover、flushslots 等命令不支持。    • 支持 cluster info、nodes、slots、slaves、myid 用以将 Proxy 模拟为单节点的 redis cluster 架构的集群。
其他	• 支持 keys、dbsize、monitor 等用于数据访问的命令。    • 不支持 wait、migrate、sync、psync、swapdb、save、shutdown，且通常业务不会直接用到。    • 支持 info keyspace 用于查看集群中各个 db key 的分布情况，支持 info rocksdb 查看 PegaDB 集群信息。

🔗 Proxy 命令支持情况

下面按类别列出了 百度云数据库Redis 6.2 支持的所有命令，并标明了当前 Redis 平台是否支持。说明：

- 表格中 Y 表示支持，N 表示不支持，Y # 表示需要依赖于 hash tag 来支持（保证多个 key 在同一分片）。
- Redis 版本，表示从哪个版本 Redis 中才出现这个命令。

🔗 server

命令	描述	Redis 版本	是否支持
acl load	Reload the ACLs from the configured ACL file. <a href="#">doc</a>	6.0.0	N
acl save	Save the current ACL rules in the configured ACL file. <a href="#">doc</a>	6.0.0	N
acl list	List the current ACL rules in ACL config file format. <a href="#">doc</a>	6.0.0	N
acl users	List the username of all the configured ACL rules. <a href="#">doc</a>	6.0.0	N
acl getuser	Get the rules for a specific ACL user. <a href="#">doc</a>	6.0.0	N
acl setuser	Modify or create the rules for a specific ACL user. <a href="#">doc</a>	6.0.0	N
acl deluser	Remove the specified ACL users and the associated rules. <a href="#">doc</a>	6.0.0	N
acl cat	List the ACL categories or the commands inside a category. <a href="#">doc</a>	6.0.0	N

acl genpass	Generate a pseudorandom secure password to use for ACL users. <a href="#">doc</a>	6.0.0	N
acl whoami	Return the name of the user associated to the current connection. <a href="#">doc</a>	6.0.0	Y
acl log	List latest events denied because of ACLs in place. <a href="#">doc</a>	6.0.0	N
acl help	Show helpful text about the different subcommands. <a href="#">doc</a>	6.0.0	N
bgrewriteaof	Asynchronously rewrite the append-only file. <a href="#">doc</a>	1.0.0	N
bgsave	Asynchronously save the dataset to disk. <a href="#">doc</a>	1.0.0	N
command	Get array of Redis command details. <a href="#">doc</a>	2.8.1	Y
command count	Get total number of Redis commands. <a href="#">doc</a>	2.8.1	Y
command getkeys	Extract keys given a full Redis command. <a href="#">doc</a>	2.8.1	Y
command info	Get array of specific Redis command details. <a href="#">doc</a>	2.8.1	Y
config get	Get the value of a configuration parameter. <a href="#">doc</a>	2.0.0	Y
config rewrite	Rewrite the configuration file with the in memory configuration. <a href="#">doc</a>	2.8.0	N
config set	Set a configuration parameter to the given value. <a href="#">doc</a>	2.0.0	Y
config resetstat	Reset the stats returned by INFO. <a href="#">doc</a>	2.0.0	N
dbsize	Return the number of keys in the selected database. <a href="#">doc</a>	1.0.0	Y
debug object	Get debugging information about a key. <a href="#">doc</a>	1.0.0	N
debug segfault	Make the server crash. <a href="#">doc</a>	1.0.0	N
failover	Start a coordinated failover between this server and one of its replicas.. <a href="#">doc</a>	6.2.0	N
flushall	Remove all keys from all databases. <a href="#">doc</a>	1.0.0	Y
flushdb	Remove all keys from the current database. <a href="#">doc</a>	1.0.0	Y
info	Get information and statistics about the server. <a href="#">doc</a>	1.0.0	Y
lolwut	Display some computer art and the Redis version. <a href="#">doc</a>	5.0.0	N
lastsave	Get the UNIX time stamp of the last successful save to disk. <a href="#">doc</a>	1.0.0	N
memory doctor	Outputs memory problems report. <a href="#">doc</a>	4.0.0	N
memory help	Show helpful text about the different subcommands. <a href="#">doc</a>	4.0.0	N
memory malloc-stats	Show allocator internal stats. <a href="#">doc</a>	4.0.0	N
memory purge	Ask the allocator to release memory. <a href="#">doc</a>	4.0.0	N
memory stats	Show memory usage details. <a href="#">doc</a>	4.0.0	N
memory usage	Estimate the memory usage of a key. <a href="#">doc</a>	4.0.0	Y
module list	List all modules loaded by the server. <a href="#">doc</a>	4.0.0	N
module load	Load a module. <a href="#">doc</a>	4.0.0	N
module unload	Unload a module. <a href="#">doc</a>	4.0.0	N
monitor	Listen for all requests received by the server in real time. <a href="#">doc</a>	1.0.0	Y

role	Return the role of the instance in the context of replication. <a href="#">doc</a>	2.8.1	Y
save	Synchronously save the dataset to disk. <a href="#">doc</a>	1.0.0	N
shutdown	Synchronously save the dataset to disk and then shut down the server. <a href="#">doc</a>	1.0.0	N
slaveof	Make the server a replica of another instance, or promote it as master. Deprecated starting with Redis 5. Use REPLICAOF instead.. <a href="#">doc</a>	1.0.0	N
replicaof	Make the server a replica of another instance, or promote it as master.. <a href="#">doc</a>	5.0.0	N
slowlog	Manages the Redis slow queries log. <a href="#">doc</a>	2.2.1	N
swapdb	Swaps two Redis databases. <a href="#">doc</a>	4.0.0	N
sync	Internal command used for replication. <a href="#">doc</a>	1.0.0	N
psync	Internal command used for replication. <a href="#">doc</a>	2.8.0	N
time	Return the current server time. <a href="#">doc</a>	2.6.0	Y
latency doctor	Return a human readable latency analysis report.. <a href="#">doc</a>	2.8.1	Y
latency graph	Return a latency graph for the event.. <a href="#">doc</a>	2.8.1	Y
latency history	Return timestamp-latency samples for the event.. <a href="#">doc</a>	2.8.1	Y
latency latest	Return the latest latency samples for all events.. <a href="#">doc</a>	2.8.1	Y
latency reset	Reset latency data for one or more events.. <a href="#">doc</a>	2.8.1	Y
latency help	Show helpful text about the different subcommands.. <a href="#">doc</a>	2.8.1	Y

 string

命令	描述	Redis 版本	是否支持
append	Append a value to a key. <a href="#">doc</a>	2.0.0	Y
bitcount	Count set bits in a string. <a href="#">doc</a>	2.6.0	Y
bitfield	Perform arbitrary bitfield integer operations on strings. <a href="#">doc</a>	3.2.0	Y
bitfield_ro	Performs arbitrary read-only bitfield integer operations on strings. <a href="#">doc</a>	6.0.0	Y
bitop	Perform bitwise operations between strings. <a href="#">doc</a>	2.6.0	Y
bitpos	Find first bit set or clear in a string. <a href="#">doc</a>	2.8.7	Y
decr	Decrement the integer value of a key by one. <a href="#">doc</a>	1.0.0	Y
decrby	Decrement the integer value of a key by the given number. <a href="#">doc</a>	1.0.0	Y
get	Get the value of a key. <a href="#">doc</a>	1.0.0	Y
getbit	Returns the bit value at offset in the string value stored at key. <a href="#">doc</a>	2.2.0	Y
getdel	Get the value of a key and delete the key. <a href="#">doc</a>	6.2.0	Y
getex	Get the value of a key and optionally set its expiration. <a href="#">doc</a>	6.2.0	Y
getrange	Get a substring of the string stored at a key. <a href="#">doc</a>	2.4.0	Y
getset	Set the string value of a key and return its old value. <a href="#">doc</a>	1.0.0	Y
incr	Increment the integer value of a key by one. <a href="#">doc</a>	1.0.0	Y
incrby	Increment the integer value of a key by the given amount. <a href="#">doc</a>	1.0.0	Y
incrbyfloat	Increment the float value of a key by the given amount. <a href="#">doc</a>	2.6.0	Y
mget	Get the values of all the given keys. <a href="#">doc</a>	1.0.0	Y
mset	Set multiple keys to multiple values. <a href="#">doc</a>	1.0.1	Y
msetnx	Set multiple keys to multiple values, only if none of the keys exist. <a href="#">doc</a>	1.0.1	Y #
psetex	Set the value and expiration in milliseconds of a key. <a href="#">doc</a>	2.6.0	Y
set	Set the string value of a key. <a href="#">doc</a>	1.0.0	Y
setbit	Sets or clears the bit at offset in the string value stored at key. <a href="#">doc</a>	2.2.0	Y
setex	Set the value and expiration of a key. <a href="#">doc</a>	2.0.0	Y
setnx	Set the value of a key, only if the key does not exist. <a href="#">doc</a>	1.0.0	Y
setrange	Overwrite part of a string at key starting at the specified offset. <a href="#">doc</a>	2.2.0	Y
stralgo	Run algorithms (currently LCS) against strings. <a href="#">doc</a>	6.0.0	N
strlen	Get the length of the value stored in a key. <a href="#">doc</a>	2.2.0	Y
lcs	Finds the longest common substring. <a href="#">doc</a>	7.0.0	Y

 connection



命令	描述	Redis 版本	是否支持
auth	Authenticate to the server. <a href="#">doc</a>	1.0.0	Y
client caching	Instruct the server about tracking or not keys in the next request. <a href="#">doc</a>	6.0.0	N
client id	Returns the client ID for the current connection. <a href="#">doc</a>	5.0.0	Y
client info	Returns information about the current client connection.. <a href="#">doc</a>	6.2.0	Y
client kill	Kill the connection of a client. <a href="#">doc</a>	2.4.0	Y
client list	Get the list of client connections. <a href="#">doc</a>	2.4.0	Y
client getname	Get the current connection name. <a href="#">doc</a>	2.6.9	Y
client getredir	Get tracking notifications redirection client ID if any. <a href="#">doc</a>	6.0.0	N
client unpause	Resume processing of clients that were paused. <a href="#">doc</a>	6.2.0	N
client pause	Stop processing commands from clients for some time. <a href="#">doc</a>	2.9.5	N
client reply	Instruct the server whether to reply to commands. <a href="#">doc</a>	3.2.0	Y
client setname	Set the current connection name. <a href="#">doc</a>	2.6.9	Y
client tracking	Enable or disable server assisted client side caching support. <a href="#">doc</a>	6.0.0	N
client trackinginfo	Return information about server assisted client side caching for the current connection. <a href="#">doc</a>	6.2.0	N
client unblock	Unblock a client blocked in a blocking command from a different connection. <a href="#">doc</a>	5.0.0	N
echo	Echo the given string. <a href="#">doc</a>	1.0.0	Y
hello	Handshake with Redis. <a href="#">doc</a>	6.0.0	Y
ping	Ping the server. <a href="#">doc</a>	1.0.0	Y
quit	Close the connection. <a href="#">doc</a>	1.0.0	Y
reset	Reset the connection. <a href="#">doc</a>	6.2.0	N
select	Change the selected database for the current connection. <a href="#">doc</a>	1.0.0	Y

 list

命令	描述	Redis 版本	是否支持
blpop	Remove and get the first element in a list, or block until one is available. <a href="#">doc</a>	2.0.0	Y #
brpop	Remove and get the last element in a list, or block until one is available. <a href="#">doc</a>	2.0.0	Y #
brpoplpush	Pop an element from a list, push it to another list and return it; or block until one is available. <a href="#">doc</a>	2.2.0	Y #
blmove	Pop an element from a list, push it to another list and return it; or block until one is available. <a href="#">doc</a>	6.2.0	Y #
lindex	Get an element from a list by its index. <a href="#">doc</a>	1.0.0	Y
linsert	Insert an element before or after another element in a list. <a href="#">doc</a>	2.2.0	Y
llen	Get the length of a list. <a href="#">doc</a>	1.0.0	Y
lpop	Remove and get the first elements in a list. <a href="#">doc</a>	1.0.0	Y
lpos	Return the index of matching elements on a list. <a href="#">doc</a>	6.0.6	Y
lpush	Prepend one or multiple elements to a list. <a href="#">doc</a>	1.0.0	Y
lpushx	Prepend an element to a list, only if the list exists. <a href="#">doc</a>	2.2.0	Y
lrange	Get a range of elements from a list. <a href="#">doc</a>	1.0.0	Y
lrem	Remove elements from a list. <a href="#">doc</a>	1.0.0	Y
lset	Set the value of an element in a list by its index. <a href="#">doc</a>	1.0.0	Y
ltrim	Trim a list to the specified range. <a href="#">doc</a>	1.0.0	Y
rpop	Remove and get the last elements in a list. <a href="#">doc</a>	1.0.0	Y
rpoplpush	Remove the last element in a list, prepend it to another list and return it. <a href="#">doc</a>	1.2.0	Y
lmove	Pop an element from a list, push it to another list and return it. <a href="#">doc</a>	6.2.0	Y
rpush	Append one or multiple elements to a list. <a href="#">doc</a>	1.0.0	Y
rpushx	Append an element to a list, only if the list exists. <a href="#">doc</a>	2.2.0	Y
lmpop	Returns multiple elements from a list after removing them. Deletes the list if the last element was popped. <a href="#">doc</a>	7.0.0	Y
blmpop	Pops the first element from one of multiple lists. Blocks until an element is available otherwise. Deletes the list if the last element was popped. <a href="#">doc</a>	7.0.0	Y

## sorted\_set

命令	描述	Redis 版本	是否支持
bzpopmin	Remove and return the member with the lowest score from one or more sorted sets, or block until one is available. <a href="#">doc</a>	5.0.0	Y
bzpopmax	Remove and return the member with the highest score from one or more sorted sets, or block until one is available. <a href="#">doc</a>	5.0.0	Y
zadd	Add one or more members to a sorted set, or update its score if it already exists. <a href="#">doc</a>	1.2.0	Y
zcard	Get the number of members in a sorted set. <a href="#">doc</a>	1.2.0	Y
zcount	Count the members in a sorted set with scores within the given values. <a href="#">doc</a>	2.0.0	Y
zdiff	Subtract multiple sorted sets. <a href="#">doc</a>	6.2.0	Y
zdiffstore	Subtract multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	6.2.0	Y #
zincrby	Increment the score of a member in a sorted set. <a href="#">doc</a>	1.2.0	Y

zinter	Intersect multiple sorted sets. <a href="#">doc</a>	6.2.0	Y
zinterstore	Intersect multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	2.0.0	Y
zlexcount	Count the number of members in a sorted set between a given lexicographical range. <a href="#">doc</a>	2.8.9	Y
zpopmax	Remove and return members with the highest scores in a sorted set. <a href="#">doc</a>	5.0.0	Y
zpopmin	Remove and return members with the lowest scores in a sorted set. <a href="#">doc</a>	5.0.0	Y
zmpop	Returns the highest- or lowest-scoring members from one or more sorted sets after removing them. Deletes the sorted set if the last member was popped. <a href="#">doc</a>	7.0.0	Y
zrandmember	Get one or multiple random elements from a sorted set. <a href="#">doc</a>	6.2.0	Y
zrangestore	Store a range of members from sorted set into another key. <a href="#">doc</a>	6.2.0	Y
zrange	Return a range of members in a sorted set. <a href="#">doc</a>	1.2.0	Y
zrangebylex	Return a range of members in a sorted set, by lexicographical range. <a href="#">doc</a>	2.8.9	Y
zrevrangebylex	Return a range of members in a sorted set, by lexicographical range, ordered from higher to lower strings.. <a href="#">doc</a>	2.8.9	Y
zrangebyscore	Return a range of members in a sorted set, by score. <a href="#">doc</a>	1.0.5	Y
zrank	Determine the index of a member in a sorted set. <a href="#">doc</a>	2.0.0	Y
zrem	Remove one or more members from a sorted set. <a href="#">doc</a>	1.2.0	Y
zremrangebylex	Remove all members in a sorted set between the given lexicographical range. <a href="#">doc</a>	2.8.9	Y
zremrangebyrank	Remove all members in a sorted set within the given indexes. <a href="#">doc</a>	2.0.0	Y
zremrangebyscore	Remove all members in a sorted set within the given scores. <a href="#">doc</a>	1.2.0	Y
zrevrange	Return a range of members in a sorted set, by index, with scores ordered from high to low. <a href="#">doc</a>	1.2.0	Y
zrevrangebyscore	Return a range of members in a sorted set, by score, with scores ordered from high to low. <a href="#">doc</a>	2.2.0	Y
zrevrank	Determine the index of a member in a sorted set, with scores ordered from high to low. <a href="#">doc</a>	2.0.0	Y
zscore	Get the score associated with the given member in a sorted set. <a href="#">doc</a>	1.2.0	Y
zunion	Add multiple sorted sets. <a href="#">doc</a>	6.2.0	Y
zmscore	Get the score associated with the given members in a sorted set. <a href="#">doc</a>	6.2.0	Y
zunionstore	Add multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	2.0.0	Y
zscan	Incrementally iterate sorted sets elements and associated scores. <a href="#">doc</a>	2.8.0	Y
bzmpop	Removes and returns a member by score from one or more sorted sets. Blocks until a member is available otherwise. Deletes the sorted set if the last element was popped. <a href="#">doc</a>	7.0.0	Y

 hash

命令	描述	Redis 版本	是否支持
hdel	Delete one or more hash fields. <a href="#">doc</a>	2.0.0	Y
hexists	Determine if a hash field exists. <a href="#">doc</a>	2.0.0	Y
hget	Get the value of a hash field. <a href="#">doc</a>	2.0.0	Y
hgetall	Get all the fields and values in a hash. <a href="#">doc</a>	2.0.0	Y
hincrby	Increment the integer value of a hash field by the given number. <a href="#">doc</a>	2.0.0	Y
hincrbyfloat	Increment the float value of a hash field by the given amount. <a href="#">doc</a>	2.6.0	Y
hkeys	Get all the fields in a hash. <a href="#">doc</a>	2.0.0	Y
hlen	Get the number of fields in a hash. <a href="#">doc</a>	2.0.0	Y
hmget	Get the values of all the given hash fields. <a href="#">doc</a>	2.0.0	Y
hmset	Set multiple hash fields to multiple values. <a href="#">doc</a>	2.0.0	Y
hset	Set the string value of a hash field. <a href="#">doc</a>	2.0.0	Y
hsetnx	Set the value of a hash field, only if the field does not exist. <a href="#">doc</a>	2.0.0	Y
hrandfield	Get one or multiple random fields from a hash. <a href="#">doc</a>	6.2.0	Y
hstrlen	Get the length of the value of a hash field. <a href="#">doc</a>	3.2.0	Y
hvals	Get all the values in a hash. <a href="#">doc</a>	2.0.0	Y
hscan	Incrementally iterate hash fields and associated values. <a href="#">doc</a>	2.8.0	Y

## set

命令	描述	Redis 版本	是否支持
sadd	Add one or more members to a set. <a href="#">doc</a>	1.0.0	Y
scard	Get the number of members in a set. <a href="#">doc</a>	1.0.0	Y
sdiff	Subtract multiple sets. <a href="#">doc</a>	1.0.0	Y
sdiffstore	Subtract multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sinter	Intersect multiple sets. <a href="#">doc</a>	1.0.0	Y
sinterstore	Intersect multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sismember	Determine if a given value is a member of a set. <a href="#">doc</a>	1.0.0	Y
smismember	Returns the membership associated with the given elements for a set. <a href="#">doc</a>	6.2.0	Y
smembers	Get all the members in a set. <a href="#">doc</a>	1.0.0	Y
smove	Move a member from one set to another. <a href="#">doc</a>	1.0.0	Y
spop	Remove and return one or multiple random members from a set. <a href="#">doc</a>	1.0.0	Y
srandmember	Get one or multiple random members from a set. <a href="#">doc</a>	1.0.0	Y
srem	Remove one or more members from a set. <a href="#">doc</a>	1.0.0	Y
sunion	Add multiple sets. <a href="#">doc</a>	1.0.0	Y
sunionstore	Add multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sscan	Incrementally iterate Set elements. <a href="#">doc</a>	2.8.0	Y
sintercard	Returns the number of members of the intersect of multiple sets. <a href="#">doc</a>	7.0.0	Y
zintercard	Returns the number of members of the intersect of multiple sorted sets. <a href="#">doc</a>	7.0.0	Y

## generic

命令	描述	Redis 版本	是否支持
copy	Copy a key. <a href="#">doc</a>	6.2.0	Y
del	Delete a key. <a href="#">doc</a>	1.0.0	Y
dump	Return a serialized version of the value stored at the specified key.. <a href="#">doc</a>	2.6.0	Y
exists	Determine if a key exists. <a href="#">doc</a>	1.0.0	Y
expire	Set a key's time to live in seconds. <a href="#">doc</a>	1.0.0	Y
expireat	Set the expiration for a key as a UNIX timestamp. <a href="#">doc</a>	1.2.0	Y
keys	Find all keys matching the given pattern. <a href="#">doc</a>	1.0.0	Y
migrate	Atomically transfer a key from a Redis instance to another one.. <a href="#">doc</a>	2.6.0	N
move	Move a key to another database. <a href="#">doc</a>	1.0.0	Y
object	Inspect the internals of Redis objects. <a href="#">doc</a>	2.2.3	Y
persist	Remove the expiration from a key. <a href="#">doc</a>	2.2.0	Y
pexpire	Set a key's time to live in milliseconds. <a href="#">doc</a>	2.6.0	Y
pexpireat	Set the expiration for a key as a UNIX timestamp specified in milliseconds. <a href="#">doc</a>	2.6.0	Y
expiretime	Returns the expiration time of a key as a Unix timestamp. <a href="#">doc</a>	7.0.0	Y
pexpiretime	Returns the expiration time of a key as a Unix milliseconds timestamp. <a href="#">doc</a>	7.0.0	Y
pttl	Get the time to live for a key in milliseconds. <a href="#">doc</a>	2.6.0	Y
randomkey	Return a random key from the keyspace. <a href="#">doc</a>	1.0.0	Y
rename	Rename a key. <a href="#">doc</a>	1.0.0	Y #
renamenx	Rename a key, only if the new key does not exist. <a href="#">doc</a>	1.0.0	Y #
restore	Create a key using the provided serialized value, previously obtained using DUMP.. <a href="#">doc</a>	2.6.0	Y
sort	Sort the elements in a list, set or sorted set. <a href="#">doc</a>	1.0.0	Y
sort_ro	Returns the sorted elements of a list, a set, or a sorted set. <a href="#">doc</a>	7.0.0	Y
touch	Alters the last access time of a key(s). Returns the number of existing keys specified.. <a href="#">doc</a>	3.2.1	Y
ttl	Get the time to live for a key. <a href="#">doc</a>	1.0.0	Y
type	Determine the type stored at key. <a href="#">doc</a>	1.0.0	Y
unlink	Delete a key asynchronously in another thread. Otherwise it is just as DEL, but non blocking.. <a href="#">doc</a>	4.0.0	Y
wait	Wait for the synchronous replication of all the write commands sent in the context of the current connection. <a href="#">doc</a>	3.0.0	N
scan	Incrementally iterate the keys space. <a href="#">doc</a>	2.8.0	Y
readonly	Enables read-only queries for a connection to a Redis Cluster replica node. <a href="#">doc</a>	3.0.0	Y
readwrite	Enables read-write queries for a connection to a Redis Cluster replica node. <a href="#">doc</a>	3.0.0	Y

命令	描述	Redis 版本	是否支持
psubscribe	Listen for messages published to channels matching the given patterns. <a href="#">doc</a>	2.0.0	Y
pubsub	Inspect the state of the Pub/Sub subsystem. <a href="#">doc</a>	2.8.0	Y
publish	Post a message to a channel. <a href="#">doc</a>	2.0.0	Y
punsubscribe	Stop listening for messages posted to channels matching the given patterns. <a href="#">doc</a>	2.0.0	Y
subscribe	Listen for messages published to the given channels. <a href="#">doc</a>	2.0.0	Y
unsubscribe	Stop listening for messages posted to the given channels. <a href="#">doc</a>	2.0.0	Y
ssubscribe	Listens for messages published to shard channels. <a href="#">doc</a>	7.0.0	Y
spublish	Post a message to a shard channel. <a href="#">doc</a>	7.0.0	Y

## scripting

命令	描述	Redis 版本	是否支持
eval	Execute a Lua script server side. <a href="#">doc</a>	2.6.0	Y
evalsha	Execute a Lua script server side. <a href="#">doc</a>	2.6.0	Y
script debug	Set the debug mode for executed scripts.. <a href="#">doc</a>	3.2.0	N
script exists	Check existence of scripts in the script cache.. <a href="#">doc</a>	2.6.0	Y
script flush	Remove all the scripts from the script cache.. <a href="#">doc</a>	2.6.0	Y
script kill	Kill the script currently in execution.. <a href="#">doc</a>	2.6.0	Y
script load	Load the specified Lua script into the script cache.. <a href="#">doc</a>	2.6.0	Y
eval_ro	Executes a read-only server-side Lua script. <a href="#">doc</a>	7.0.0	Y
evalsha_ro	Executes a read-only server-side Lua script by SHA1 digest. <a href="#">doc</a>	7.0.0	Y
function delete	Deletes a library and its functions. <a href="#">doc</a>	7.0.0	Y
function dump	Dumps all libraries into a serialized binary payload. <a href="#">doc</a>	7.0.0	Y
function flush	Deletes all libraries and functions. <a href="#">doc</a>	7.0.0	Y
function kill	Terminates a function during execution. <a href="#">doc</a>	7.0.0	Y
function list	Returns information about all libraries. <a href="#">doc</a>	7.0.0	Y
function load	Creates a library. <a href="#">doc</a>	7.0.0	Y
function restore	Restores all libraries from a payload. <a href="#">doc</a>	7.0.0	Y
function stats	Returns information about a function during execution. <a href="#">doc</a>	7.0.0	Y
fcall	Invokes a function. <a href="#">doc</a>	7.0.0	Y
fcall_ro	Invokes a read-only function. <a href="#">doc</a>	7.0.0	Y

## geo

命令	描述	Redis 版本	是否支持
geoadd	Add one or more geospatial items in the geospatial index represented using a sorted set. <a href="#">doc</a>	3.2.0	Y
geohash	Returns members of a geospatial index as standard geohash strings. <a href="#">doc</a>	3.2.0	Y
geopos	Returns longitude and latitude of members of a geospatial index. <a href="#">doc</a>	3.2.0	Y
geodist	Returns the distance between two members of a geospatial index. <a href="#">doc</a>	3.2.0	Y
georadius	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a point. <a href="#">doc</a>	3.2.0	Y
georadiusbymember	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a member. <a href="#">doc</a>	3.2.0	Y
georadiusbymember_ro	Returns members from a geospatial index that are within a distance from a member. <a href="#">doc</a>	7.0.0	Y
geosearch	Query a sorted set representing a geospatial index to fetch members inside an area of a box or a circle.. <a href="#">doc</a>	6.2.0	Y
geosearchstore	Query a sorted set representing a geospatial index to fetch members inside an area of a box or a circle, and store the result in another key.. <a href="#">doc</a>	6.2.0	Y

#### hyperloglog

命令	描述	Redis 版本	是否支持
pfadd	Adds the specified elements to the specified HyperLogLog.. <a href="#">doc</a>	2.8.9	Y
pfcount	Return the approximated cardinality of the set(s) observed by the HyperLogLog at key(s).. <a href="#">doc</a>	2.8.9	Y
pfmerge	Merge N different HyperLogLogs into a single one.. <a href="#">doc</a>	2.8.9	Y

#### cluster

命令	描述	Redis 版本	是否支持
cluster addslots	Assign new hash slots to receiving node. <a href="#">doc</a>	3.0.0	N
cluster bumpepoch	Advance the cluster config epoch. <a href="#">doc</a>	3.0.0	N
cluster count-failure-reports	Return the number of failure reports active for a given node. <a href="#">doc</a>	3.0.0	N
cluster countkeysinslot	Return the number of local keys in the specified hash slot. <a href="#">doc</a>	3.0.0	N
cluster delslots	Set hash slots as unbound in receiving node. <a href="#">doc</a>	3.0.0	N
cluster failover	Forces a replica to perform a manual failover of its master.. <a href="#">doc</a>	3.0.0	N
cluster flushslots	Delete a node's own slots information. <a href="#">doc</a>	3.0.0	N
cluster forget	Remove a node from the nodes table. <a href="#">doc</a>	3.0.0	N
cluster getkeysinslot	Return local key names in the specified hash slot. <a href="#">doc</a>	3.0.0	N
cluster info	Provides info about Redis Cluster node state. <a href="#">doc</a>	3.0.0	Y
cluster keyslot	Returns the hash slot of the specified key. <a href="#">doc</a>	3.0.0	Y
cluster meet	Force a node cluster to handshake with another node. <a href="#">doc</a>	3.0.0	N
cluster myid	Return the node id. <a href="#">doc</a>	3.0.0	Y
cluster nodes	Get Cluster config for the node. <a href="#">doc</a>	3.0.0	Y
cluster replicate	Reconfigure a node as a replica of the specified master node. <a href="#">doc</a>	3.0.0	N
cluster reset	Reset a Redis Cluster node. <a href="#">doc</a>	3.0.0	N
cluster saveconfig	Forces the node to save cluster state on disk. <a href="#">doc</a>	3.0.0	N
cluster set-config-epoch	Set the configuration epoch in a new node. <a href="#">doc</a>	3.0.0	N
cluster setslot	Bind a hash slot to a specific node. <a href="#">doc</a>	3.0.0	N
cluster slaves	List replica nodes of the specified master node. <a href="#">doc</a>	3.0.0	Y
cluster replicas	List replica nodes of the specified master node. <a href="#">doc</a>	5.0.0	Y
cluster slots	Get array of Cluster slot to node mappings. <a href="#">doc</a>	3.0.0	Y
readonly	Enables read queries for a connection to a cluster replica node. <a href="#">doc</a>	3.0.0	Y
readwrite	Disables read queries for a connection to a cluster replica node. <a href="#">doc</a>	3.0.0	Y

 stream



命令	描述	Redis 版本	是否支持
xinfo	Get information on streams and consumer groups. <a href="#">doc</a>	5.0.0	Y
xadd	Appends a new entry to a stream. <a href="#">doc</a>	5.0.0	Y
xtrim	Trims the stream to (approximately if '~' is passed) a certain size. <a href="#">doc</a>	5.0.0	Y
xdel	Removes the specified entries from the stream. Returns the number of items actually deleted, that may be different from the number of IDs passed in case certain IDs do not exist.. <a href="#">doc</a>	5.0.0	Y
xrange	Return a range of elements in a stream, with IDs matching the specified IDs interval. <a href="#">doc</a>	5.0.0	Y
xrevrange	Return a range of elements in a stream, with IDs matching the specified IDs interval, in reverse order (from greater to smaller IDs) compared to XRANGE. <a href="#">doc</a>	5.0.0	Y
xlen	Return the number of entries in a stream. <a href="#">doc</a>	5.0.0	Y
xread	Return never seen elements in multiple streams, with IDs greater than the ones reported by the caller for each stream. Can block.. <a href="#">doc</a>	5.0.0	Y
xgroup	Create, destroy, and manage consumer groups.. <a href="#">doc</a>	5.0.0	Y
xreadgroup	Return new entries from a stream using a consumer group, or access the history of the pending entries for a given consumer. Can block.. <a href="#">doc</a>	5.0.0	Y
xack	Marks a pending message as correctly processed, effectively removing it from the pending entries list of the consumer group. Return value of the command is the number of messages successfully acknowledged, that is, the IDs we were actually able to resolve in the PEL.. <a href="#">doc</a>	5.0.0	Y
xclaim	Changes (or acquires) ownership of a message in a consumer group, as if the message was delivered to the specified consumer.. <a href="#">doc</a>	5.0.0	Y
xautoclaim	Changes (or acquires) ownership of messages in a consumer group, as if the messages were delivered to the specified consumer.. <a href="#">doc</a>	6.2.0	Y
xpending	Return information and entries from a stream consumer group pending entries list, that are messages fetched but never acknowledged.. <a href="#">doc</a>	5.0.0	Y
xsetid	An internal command for replicating stream values. <a href="#">doc</a>	5.0.0	Y

🔗 transactions

支持率 : 5 / 5

命令	描述	Redis 版本	是否支持
discard	Discard all commands issued after MULTI. <a href="#">doc</a>	2.0.0	Y #
exec	Execute all commands issued after MULTI. <a href="#">doc</a>	1.2.0	Y #
multi	Mark the start of a transaction block. <a href="#">doc</a>	1.2.0	Y #
unwatch	Forget about all watched keys. <a href="#">doc</a>	2.2.0	Y #
watch	Watch the given keys to determine execution of the MULTI/EXEC block. <a href="#">doc</a>	2.2.0	Y #

Redis容量型（PegaDB）使用命令限制

概述

本文档以 Redis 7.2 版本的[全部命令](#)为基准。

🔗 主从节点实例与集群实例不支持的命令

- BGSAVE

- FLUSHDB
- FLUSHALL
- MIGRATE
- DEBUG
- BGREWRITEAOF
- LASTSAVE
- SAVE
- SYNC
- SLAVEOF
- SHUTDOWN
- ACL

🔗 PROXY命令支持情况概览

类别	支持情况
数据读写类命令	string、list、zset、hash、set、pubsub、geo、hyperloglog、stream、transaction 均支持
acl	访问控制列表通过控制台配置，仅支持 acl whoami 查看当前用户
config	不支持，可通过控制台变更配置
memory	仅支持 memory usage，集群内存使用情况，通过控制台查看
client	支持 clientid、info、kill、list、getname、setname、reply，其他不支持
cluster	因为 SCS 是基于代理的架构和 redis cluster 在架构上有较大差异。      • cluster addslots、countkeysinslot、delslots、failover、flushslots 等命令不支持。    • 支持 cluster info、nodes、slots、slaves、myid 用以将 Proxy 模拟为单节点的 redis cluster 架构的集群。
其他	• 支持 keys、dbsize、monitor 等用于数据访问的命令。    • 不支持 wait、migrate、sync、psync、swapdb、save、shutdown，且通常业务不会直接用到。    • 支持 info keypace 用于查看集群中各个 db key 的分布情况，支持 info rocksdb 查看 PegaDB 集群信息。

🔗 Proxy 命令支持情况

下面按类别列出了 百度云数据库Redis 6.2 支持的所有命令，并标明了当前 Redis 平台是否支持。说明：

- 表格中 Y 表示支持，N 表示不支持，Y # 表示需要依赖于 hash tag 来支持（保证多个 key 在同一分片）。
- Redis 版本，表示从哪个版本 Redis 中才出现这个命令。

🔗 server

命令	描述	Redis 版本	是否支持
acl load	Reload the ACLs from the configured ACL file. <a href="#">doc</a>	6.0.0	N
acl save	Save the current ACL rules in the configured ACL file. <a href="#">doc</a>	6.0.0	N

acl list	List the current ACL rules in ACL config file format. <a href="#">doc</a>	6.0.0	N
acl users	List the username of all the configured ACL rules. <a href="#">doc</a>	6.0.0	N
acl getuser	Get the rules for a specific ACL user. <a href="#">doc</a>	6.0.0	N
acl setuser	Modify or create the rules for a specific ACL user. <a href="#">doc</a>	6.0.0	N
acl deluser	Remove the specified ACL users and the associated rules. <a href="#">doc</a>	6.0.0	N
acl cat	List the ACL categories or the commands inside a category. <a href="#">doc</a>	6.0.0	N
acl genpass	Generate a pseudorandom secure password to use for ACL users. <a href="#">doc</a>	6.0.0	N
acl whoami	Return the name of the user associated to the current connection. <a href="#">doc</a>	6.0.0	Y
acl log	List latest events denied because of ACLs in place. <a href="#">doc</a>	6.0.0	N
acl help	Show helpful text about the different subcommands. <a href="#">doc</a>	6.0.0	N
bgrewriteaof	Asynchronously rewrite the append-only file. <a href="#">doc</a>	1.0.0	N
bgsave	Asynchronously save the dataset to disk. <a href="#">doc</a>	1.0.0	N
command	Get array of Redis command details. <a href="#">doc</a>	2.8.1	Y
command count	Get total number of Redis commands. <a href="#">doc</a>	2.8.1	Y
command getkeys	Extract keys given a full Redis command. <a href="#">doc</a>	2.8.1	Y
command info	Get array of specific Redis command details. <a href="#">doc</a>	2.8.1	Y
config get	Get the value of a configuration parameter. <a href="#">doc</a>	2.0.0	Y
config rewrite	Rewrite the configuration file with the in memory configuration. <a href="#">doc</a>	2.8.0	N
config set	Set a configuration parameter to the given value. <a href="#">doc</a>	2.0.0	N
config resetstat	Reset the stats returned by INFO. <a href="#">doc</a>	2.0.0	N
dbsize	Return the number of keys in the selected database. <a href="#">doc</a>	1.0.0	Y
debug object	Get debugging information about a key. <a href="#">doc</a>	1.0.0	N
debug segfault	Make the server crash. <a href="#">doc</a>	1.0.0	N
failover	Start a coordinated failover between this server and one of its replicas.. <a href="#">doc</a>	6.2.0	N
flushall	Remove all keys from all databases. <a href="#">doc</a>	1.0.0	N
flushdb	Remove all keys from the current database. <a href="#">doc</a>	1.0.0	N
info	Get information and statistics about the server. <a href="#">doc</a>	1.0.0	Y
lolwut	Display some computer art and the Redis version. <a href="#">doc</a>	5.0.0	N
lastsave	Get the UNIX time stamp of the last successful save to disk. <a href="#">doc</a>	1.0.0	N
memory doctor	Outputs memory problems report. <a href="#">doc</a>	4.0.0	N
memory help	Show helpful text about the different subcommands. <a href="#">doc</a>	4.0.0	N
memory malloc-stats	Show allocator internal stats. <a href="#">doc</a>	4.0.0	N
memory purge	Ask the allocator to release memory. <a href="#">doc</a>	4.0.0	N
memory stats	Show memory usage details. <a href="#">doc</a>	4.0.0	N

memory usage	Estimate the memory usage of a key. <a href="#">doc</a>	4.0.0	N
module list	List all modules loaded by the server. <a href="#">doc</a>	4.0.0	N
module load	Load a module. <a href="#">doc</a>	4.0.0	N
module unload	Unload a module. <a href="#">doc</a>	4.0.0	N
monitor	Listen for all requests received by the server in real time. <a href="#">doc</a>	1.0.0	Y
role	Return the role of the instance in the context of replication. <a href="#">doc</a>	2.8.1	Y
save	Synchronously save the dataset to disk. <a href="#">doc</a>	1.0.0	N
shutdown	Synchronously save the dataset to disk and then shut down the server. <a href="#">doc</a>	1.0.0	N
slaveof	Make the server a replica of another instance, or promote it as master. Deprecated starting with Redis 5. Use REPLICAOF instead.. <a href="#">doc</a>	1.0.0	N
replicaof	Make the server a replica of another instance, or promote it as master.. <a href="#">doc</a>	5.0.0	N
slowlog	Manages the Redis slow queries log. <a href="#">doc</a>	2.2.1	N
swapdb	Swaps two Redis databases. <a href="#">doc</a>	4.0.0	N
sync	Internal command used for replication. <a href="#">doc</a>	1.0.0	N
psync	Internal command used for replication. <a href="#">doc</a>	2.8.0	N
time	Return the current server time. <a href="#">doc</a>	2.6.0	Y
latency doctor	Return a human readable latency analysis report.. <a href="#">doc</a>	2.8.1	N
latency graph	Return a latency graph for the event.. <a href="#">doc</a>	2.8.1	N
latency history	Return timestamp-latency samples for the event.. <a href="#">doc</a>	2.8.1	N
latency latest	Return the latest latency samples for all events.. <a href="#">doc</a>	2.8.1	N
latency reset	Reset latency data for one or more events.. <a href="#">doc</a>	2.8.1	N
latency help	Show helpful text about the different subcommands.. <a href="#">doc</a>	2.8.1	N

 string

命令	描述	Redis 版本	是否支持
append	Append a value to a key. <a href="#">doc</a>	2.0.0	Y
bitcount	Count set bits in a string. <a href="#">doc</a>	2.6.0	Y
bitfield	Perform arbitrary bitfield integer operations on strings. <a href="#">doc</a>	3.2.0	Y
bitfield_ro	Performs arbitrary read-only bitfield integer operations on strings. <a href="#">doc</a>	6.0.0	N
bitop	Perform bitwise operations between strings. <a href="#">doc</a>	2.6.0	N
bitpos	Find first bit set or clear in a string. <a href="#">doc</a>	2.8.7	Y
decr	Decrement the integer value of a key by one. <a href="#">doc</a>	1.0.0	Y
decrby	Decrement the integer value of a key by the given number. <a href="#">doc</a>	1.0.0	Y
get	Get the value of a key. <a href="#">doc</a>	1.0.0	Y
getbit	Returns the bit value at offset in the string value stored at key. <a href="#">doc</a>	2.2.0	Y
getdel	Get the value of a key and delete the key. <a href="#">doc</a>	6.2.0	Y
getex	Get the value of a key and optionally set its expiration. <a href="#">doc</a>	6.2.0	Y
getrange	Get a substring of the string stored at a key. <a href="#">doc</a>	2.4.0	Y
getset	Set the string value of a key and return its old value. <a href="#">doc</a>	1.0.0	Y
incr	Increment the integer value of a key by one. <a href="#">doc</a>	1.0.0	Y
incrby	Increment the integer value of a key by the given amount. <a href="#">doc</a>	1.0.0	Y
incrbyfloat	Increment the float value of a key by the given amount. <a href="#">doc</a>	2.6.0	Y
mget	Get the values of all the given keys. <a href="#">doc</a>	1.0.0	Y
mset	Set multiple keys to multiple values. <a href="#">doc</a>	1.0.1	Y
msetnx	Set multiple keys to multiple values, only if none of the keys exist. <a href="#">doc</a>	1.0.1	Y
psetex	Set the value and expiration in milliseconds of a key. <a href="#">doc</a>	2.6.0	Y
set	Set the string value of a key. <a href="#">doc</a>	1.0.0	Y
setbit	Sets or clears the bit at offset in the string value stored at key. <a href="#">doc</a>	2.2.0	Y
setex	Set the value and expiration of a key. <a href="#">doc</a>	2.0.0	Y
setnx	Set the value of a key, only if the key does not exist. <a href="#">doc</a>	1.0.0	Y
setrange	Overwrite part of a string at key starting at the specified offset. <a href="#">doc</a>	2.2.0	Y
stralgo	Run algorithms (currently LCS) against strings. <a href="#">doc</a>	6.0.0	N
strlen	Get the length of the value stored in a key. <a href="#">doc</a>	2.2.0	Y
lcs	Finds the longest common substring. <a href="#">doc</a>	7.0.0	N

 connection

命令	描述	Redis 版本	是否支持
auth	Authenticate to the server. <a href="#">doc</a>	1.0.0	Y
client caching	Instruct the server about tracking or not keys in the next request. <a href="#">doc</a>	6.0.0	N
client id	Returns the client ID for the current connection. <a href="#">doc</a>	5.0.0	Y
client info	Returns information about the current client connection.. <a href="#">doc</a>	6.2.0	Y
client kill	Kill the connection of a client. <a href="#">doc</a>	2.4.0	Y
client list	Get the list of client connections. <a href="#">doc</a>	2.4.0	Y
client getname	Get the current connection name. <a href="#">doc</a>	2.6.9	Y
client getredir	Get tracking notifications redirection client ID if any. <a href="#">doc</a>	6.0.0	N
client unpause	Resume processing of clients that were paused. <a href="#">doc</a>	6.2.0	N
client pause	Stop processing commands from clients for some time. <a href="#">doc</a>	2.9.5	N
client reply	Instruct the server whether to reply to commands. <a href="#">doc</a>	3.2.0	Y
client setname	Set the current connection name. <a href="#">doc</a>	2.6.9	Y
client tracking	Enable or disable server assisted client side caching support. <a href="#">doc</a>	6.0.0	N
client trackinginfo	Return information about server assisted client side caching for the current connection. <a href="#">doc</a>	6.2.0	N
client unblock	Unblock a client blocked in a blocking command from a different connection. <a href="#">doc</a>	5.0.0	N
echo	Echo the given string. <a href="#">doc</a>	1.0.0	Y
hello	Handshake with Redis. <a href="#">doc</a>	6.0.0	Y
ping	Ping the server. <a href="#">doc</a>	1.0.0	Y
quit	Close the connection. <a href="#">doc</a>	1.0.0	Y
reset	Reset the connection. <a href="#">doc</a>	6.2.0	N
select	Change the selected database for the current connection. <a href="#">doc</a>	1.0.0	Y

 list

命令	描述	Redis 版本	是否支持
blpop	Remove and get the first element in a list, or block until one is available. <a href="#">doc</a>	2.0.0	Y #
brpop	Remove and get the last element in a list, or block until one is available. <a href="#">doc</a>	2.0.0	Y #
brpoplpush	Pop an element from a list, push it to another list and return it; or block until one is available. <a href="#">doc</a>	2.2.0	N
blmove	Pop an element from a list, push it to another list and return it; or block until one is available. <a href="#">doc</a>	6.2.0	N
lindex	Get an element from a list by its index. <a href="#">doc</a>	1.0.0	Y
linsert	Insert an element before or after another element in a list. <a href="#">doc</a>	2.2.0	Y
llen	Get the length of a list. <a href="#">doc</a>	1.0.0	Y
lpop	Remove and get the first elements in a list. <a href="#">doc</a>	1.0.0	Y
lpos	Return the index of matching elements on a list. <a href="#">doc</a>	6.0.6	N
lpush	Prepend one or multiple elements to a list. <a href="#">doc</a>	1.0.0	Y
lpushx	Prepend an element to a list, only if the list exists. <a href="#">doc</a>	2.2.0	Y
lrange	Get a range of elements from a list. <a href="#">doc</a>	1.0.0	Y
lrem	Remove elements from a list. <a href="#">doc</a>	1.0.0	Y
lset	Set the value of an element in a list by its index. <a href="#">doc</a>	1.0.0	Y
ltrim	Trim a list to the specified range. <a href="#">doc</a>	1.0.0	Y
rpop	Remove and get the last elements in a list. <a href="#">doc</a>	1.0.0	Y
rpoplpush	Remove the last element in a list, prepend it to another list and return it. <a href="#">doc</a>	1.2.0	Y
lmove	Pop an element from a list, push it to another list and return it. <a href="#">doc</a>	6.2.0	N
rpush	Append one or multiple elements to a list. <a href="#">doc</a>	1.0.0	Y
rpushx	Append an element to a list, only if the list exists. <a href="#">doc</a>	2.2.0	Y
lmpop	Returns multiple elements from a list after removing them. Deletes the list if the last element was popped. <a href="#">doc</a>	7.0.0	N
blmpop	Pops the first element from one of multiple lists. Blocks until an element is available otherwise. Deletes the list if the last element was popped. <a href="#">doc</a>	7.0.0	N

## 🔗 sorted\_set

命令	描述	Redis 版本	是否支持
bzpopmin	Remove and return the member with the lowest score from one or more sorted sets, or block until one is available. <a href="#">doc</a>	5.0.0	N
bzpopmax	Remove and return the member with the highest score from one or more sorted sets, or block until one is available. <a href="#">doc</a>	5.0.0	N
zadd	Add one or more members to a sorted set, or update its score if it already exists. <a href="#">doc</a>	1.2.0	Y
zcard	Get the number of members in a sorted set. <a href="#">doc</a>	1.2.0	Y
zcount	Count the members in a sorted set with scores within the given values. <a href="#">doc</a>	2.0.0	Y
zdiff	Subtract multiple sorted sets. <a href="#">doc</a>	6.2.0	N
zdiffstore	Subtract multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	6.2.0	N
zincrby	Increment the score of a member in a sorted set. <a href="#">doc</a>	1.2.0	Y

zinter	Intersect multiple sorted sets. <a href="#">doc</a>	6.2.0	N
zinterstore	Intersect multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	2.0.0	Y
zlexcount	Count the number of members in a sorted set between a given lexicographical range. <a href="#">doc</a>	2.8.9	Y
zpopmax	Remove and return members with the highest scores in a sorted set. <a href="#">doc</a>	5.0.0	Y
zpopmin	Remove and return members with the lowest scores in a sorted set. <a href="#">doc</a>	5.0.0	Y
zmpop	Returns the highest- or lowest-scoring members from one or more sorted sets after removing them. Deletes the sorted set if the last member was popped. <a href="#">doc</a>	7.0.0	N
zrandmember	Get one or multiple random elements from a sorted set. <a href="#">doc</a>	6.2.0	N
zrangestore	Store a range of members from sorted set into another key. <a href="#">doc</a>	6.2.0	N
zrange	Return a range of members in a sorted set. <a href="#">doc</a>	1.2.0	Y
zrangebylex	Return a range of members in a sorted set, by lexicographical range. <a href="#">doc</a>	2.8.9	N
zrevrangebylex	Return a range of members in a sorted set, by lexicographical range, ordered from higher to lower strings.. <a href="#">doc</a>	2.8.9	Y
zrangebyscore	Return a range of members in a sorted set, by score. <a href="#">doc</a>	1.0.5	Y
zrank	Determine the index of a member in a sorted set. <a href="#">doc</a>	2.0.0	Y
zrem	Remove one or more members from a sorted set. <a href="#">doc</a>	1.2.0	Y
zremrangebylex	Remove all members in a sorted set between the given lexicographical range. <a href="#">doc</a>	2.8.9	Y
zremrangebyrank	Remove all members in a sorted set within the given indexes. <a href="#">doc</a>	2.0.0	Y
zremrangebyscore	Remove all members in a sorted set within the given scores. <a href="#">doc</a>	1.2.0	Y
zrevrange	Return a range of members in a sorted set, by index, with scores ordered from high to low. <a href="#">doc</a>	1.2.0	Y
zrevrangebyscore	Return a range of members in a sorted set, by score, with scores ordered from high to low. <a href="#">doc</a>	2.2.0	Y
zrevrank	Determine the index of a member in a sorted set, with scores ordered from high to low. <a href="#">doc</a>	2.0.0	Y
zscore	Get the score associated with the given member in a sorted set. <a href="#">doc</a>	1.2.0	Y
zunion	Add multiple sorted sets. <a href="#">doc</a>	6.2.0	N
zmscore	Get the score associated with the given members in a sorted set. <a href="#">doc</a>	6.2.0	Y
zunionstore	Add multiple sorted sets and store the resulting sorted set in a new key. <a href="#">doc</a>	2.0.0	Y
zscan	Incrementally iterate sorted sets elements and associated scores. <a href="#">doc</a>	2.8.0	Y
bzmpop	Removes and returns a member by score from one or more sorted sets. Blocks until a member is available otherwise. Deletes the sorted set if the last element was popped. <a href="#">doc</a>	7.0.0	N

 hash



命令	描述	Redis 版本	是否支持
hdel	Delete one or more hash fields. <a href="#">doc</a>	2.0.0	Y
hexists	Determine if a hash field exists. <a href="#">doc</a>	2.0.0	Y
hget	Get the value of a hash field. <a href="#">doc</a>	2.0.0	Y
hgetall	Get all the fields and values in a hash. <a href="#">doc</a>	2.0.0	Y
hincrby	Increment the integer value of a hash field by the given number. <a href="#">doc</a>	2.0.0	Y
hincrbyfloat	Increment the float value of a hash field by the given amount. <a href="#">doc</a>	2.6.0	Y
hkeys	Get all the fields in a hash. <a href="#">doc</a>	2.0.0	Y
hlen	Get the number of fields in a hash. <a href="#">doc</a>	2.0.0	Y
hmget	Get the values of all the given hash fields. <a href="#">doc</a>	2.0.0	Y
hmset	Set multiple hash fields to multiple values. <a href="#">doc</a>	2.0.0	Y
hset	Set the string value of a hash field. <a href="#">doc</a>	2.0.0	Y
hsetnx	Set the value of a hash field, only if the field does not exist. <a href="#">doc</a>	2.0.0	Y
hrandfield	Get one or multiple random fields from a hash. <a href="#">doc</a>	6.2.0	N
hstrlen	Get the length of the value of a hash field. <a href="#">doc</a>	3.2.0	Y
hvals	Get all the values in a hash. <a href="#">doc</a>	2.0.0	Y
hscan	Incrementally iterate hash fields and associated values. <a href="#">doc</a>	2.8.0	Y

## set

命令	描述	Redis 版本	是否支持
sadd	Add one or more members to a set. <a href="#">doc</a>	1.0.0	Y
scard	Get the number of members in a set. <a href="#">doc</a>	1.0.0	Y
sdiff	Subtract multiple sets. <a href="#">doc</a>	1.0.0	Y
sdiffstore	Subtract multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sinter	Intersect multiple sets. <a href="#">doc</a>	1.0.0	Y
sinterstore	Intersect multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sismember	Determine if a given value is a member of a set. <a href="#">doc</a>	1.0.0	Y
smismember	Returns the membership associated with the given elements for a set. <a href="#">doc</a>	6.2.0	N
smembers	Get all the members in a set. <a href="#">doc</a>	1.0.0	Y
smove	Move a member from one set to another. <a href="#">doc</a>	1.0.0	Y
spop	Remove and return one or multiple random members from a set. <a href="#">doc</a>	1.0.0	Y
srandmember	Get one or multiple random members from a set. <a href="#">doc</a>	1.0.0	Y
srem	Remove one or more members from a set. <a href="#">doc</a>	1.0.0	Y
sunion	Add multiple sets. <a href="#">doc</a>	1.0.0	Y
sunionstore	Add multiple sets and store the resulting set in a key. <a href="#">doc</a>	1.0.0	Y
sscan	Incrementally iterate Set elements. <a href="#">doc</a>	2.8.0	Y
sintercard	Returns the number of members of the intersect of multiple sets. <a href="#">doc</a>	7.0.0	N
zintercard	Returns the number of members of the intersect of multiple sorted sets. <a href="#">doc</a>	7.0.0	N

## generic

命令	描述	Redis 版本	是否支持
copy	Copy a key. <a href="#">doc</a>	6.2.0	N
del	Delete a key. <a href="#">doc</a>	1.0.0	Y
dump	Return a serialized version of the value stored at the specified key.. <a href="#">doc</a>	2.6.0	N
exists	Determine if a key exists. <a href="#">doc</a>	1.0.0	Y
expire	Set a key's time to live in seconds. <a href="#">doc</a>	1.0.0	Y
expireat	Set the expiration for a key as a UNIX timestamp. <a href="#">doc</a>	1.2.0	Y
keys	Find all keys matching the given pattern. <a href="#">doc</a>	1.0.0	Y
migrate	Atomically transfer a key from a Redis instance to another one.. <a href="#">doc</a>	2.6.0	N
move	Move a key to another database. <a href="#">doc</a>	1.0.0	N
object	Inspect the internals of Redis objects. <a href="#">doc</a>	2.2.3	Y
persist	Remove the expiration from a key. <a href="#">doc</a>	2.2.0	Y
pexpire	Set a key's time to live in milliseconds. <a href="#">doc</a>	2.6.0	Y
pexpireat	Set the expiration for a key as a UNIX timestamp specified in milliseconds. <a href="#">doc</a>	2.6.0	Y
expiretime	Returns the expiration time of a key as a Unix timestamp. <a href="#">doc</a>	7.0.0	N
pexpiretime	Returns the expiration time of a key as a Unix milliseconds timestamp. <a href="#">doc</a>	7.0.0	N
pttl	Get the time to live for a key in milliseconds. <a href="#">doc</a>	2.6.0	Y
randomkey	Return a random key from the keyspace. <a href="#">doc</a>	1.0.0	Y
rename	Rename a key. <a href="#">doc</a>	1.0.0	N
renamenx	Rename a key, only if the new key does not exist. <a href="#">doc</a>	1.0.0	N
restore	Create a key using the provided serialized value, previously obtained using DUMP.. <a href="#">doc</a>	2.6.0	N
sort	Sort the elements in a list, set or sorted set. <a href="#">doc</a>	1.0.0	N
sort_ro	Returns the sorted elements of a list, a set, or a sorted set. <a href="#">doc</a>	7.0.0	N
touch	Alters the last access time of a key(s). Returns the number of existing keys specified.. <a href="#">doc</a>	3.2.1	N
ttd	Get the time to live for a key. <a href="#">doc</a>	1.0.0	Y
type	Determine the type stored at key. <a href="#">doc</a>	1.0.0	Y
unlink	Delete a key asynchronously in another thread. Otherwise it is just as DEL, but non blocking.. <a href="#">doc</a>	4.0.0	Y
wait	Wait for the synchronous replication of all the write commands sent in the context of the current connection. <a href="#">doc</a>	3.0.0	N
scan	Incrementally iterate the keys space. <a href="#">doc</a>	2.8.0	Y
readonly	Enables read-only queries for a connection to a Redis Cluster replica node. <a href="#">doc</a>	3.0.0	Y
readwrite	Enables read-write queries for a connection to a Reids Cluster replica node. <a href="#">doc</a>	3.0.0	Y

命令	描述	Redis 版本	是否支持
psubscribe	Listen for messages published to channels matching the given patterns. <a href="#">doc</a>	2.0.0	Y
pubsub	Inspect the state of the Pub/Sub subsystem. <a href="#">doc</a>	2.8.0	Y
publish	Post a message to a channel. <a href="#">doc</a>	2.0.0	Y
punsubscribe	Stop listening for messages posted to channels matching the given patterns. <a href="#">doc</a>	2.0.0	Y
subscribe	Listen for messages published to the given channels. <a href="#">doc</a>	2.0.0	Y
unsubscribe	Stop listening for messages posted to the given channels. <a href="#">doc</a>	2.0.0	Y
ssubscribe	Listens for messages published to shard channels. <a href="#">doc</a>	7.0.0	N
spublish	Post a message to a shard channel. <a href="#">doc</a>	7.0.0	N

## scripting

命令	描述	Redis 版本	是否支持
eval	Execute a Lua script server side. <a href="#">doc</a>	2.6.0	Y
evalsha	Execute a Lua script server side. <a href="#">doc</a>	2.6.0	Y
script debug	Set the debug mode for executed scripts.. <a href="#">doc</a>	3.2.0	N
script exists	Check existence of scripts in the script cache.. <a href="#">doc</a>	2.6.0	Y
script flush	Remove all the scripts from the script cache.. <a href="#">doc</a>	2.6.0	Y
script kill	Kill the script currently in execution.. <a href="#">doc</a>	2.6.0	Y
script load	Load the specified Lua script into the script cache.. <a href="#">doc</a>	2.6.0	Y
eval_ro	Executes a read-only server-side Lua script. <a href="#">doc</a>	7.0.0	N
evalsha_ro	Executes a read-only server-side Lua script by SHA1 digest. <a href="#">doc</a>	7.0.0	N
function delete	Deletes a library and its functions. <a href="#">doc</a>	7.0.0	Y
function dump	Dumps all libraries into a serialized binary payload. <a href="#">doc</a>	7.0.0	N
function flush	Deletes all libraries and functions. <a href="#">doc</a>	7.0.0	N
function kill	Terminates a function during execution. <a href="#">doc</a>	7.0.0	N
function list	Returns information about all libraries. <a href="#">doc</a>	7.0.0	Y
function load	Creates a library. <a href="#">doc</a>	7.0.0	Y
function restore	Restores all libraries from a payload. <a href="#">doc</a>	7.0.0	N
function stats	Returns information about a function during execution. <a href="#">doc</a>	7.0.0	N
fcall	Invokes a function. <a href="#">doc</a>	7.0.0	Y
fcall_ro	Invokes a read-only function. <a href="#">doc</a>	7.0.0	N

## geo

命令	描述	Redis 版本	是否支持
geoadd	Add one or more geospatial items in the geospatial index represented using a sorted set. <a href="#">doc</a>	3.2.0	Y
geohash	Returns members of a geospatial index as standard geohash strings. <a href="#">doc</a>	3.2.0	Y
geopos	Returns longitude and latitude of members of a geospatial index. <a href="#">doc</a>	3.2.0	Y
geodist	Returns the distance between two members of a geospatial index. <a href="#">doc</a>	3.2.0	Y
georadius	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a point. <a href="#">doc</a>	3.2.0	Y
georadiusbymember	Query a sorted set representing a geospatial index to fetch members matching a given maximum distance from a member. <a href="#">doc</a>	3.2.0	Y
georadiusbymember_ro	Returns members from a geospatial index that are within a distance from a member. <a href="#">doc</a>	7.0.0	Y
geosearch	Query a sorted set representing a geospatial index to fetch members inside an area of a box or a circle.. <a href="#">doc</a>	6.2.0	N
geosearchstore	Query a sorted set representing a geospatial index to fetch members inside an area of a box or a circle, and store the result in another key.. <a href="#">doc</a>	6.2.0	N

#### hyperloglog

命令	描述	Redis 版本	是否支持
pfadd	Adds the specified elements to the specified HyperLogLog.. <a href="#">doc</a>	2.8.9	Y
pfcount	Return the approximated cardinality of the set(s) observed by the HyperLogLog at key(s).. <a href="#">doc</a>	2.8.9	Y
pfmerge	Merge N different HyperLogLogs into a single one.. <a href="#">doc</a>	2.8.9	Y

#### cluster

命令	描述	Redis 版本	是否支持
cluster addslots	Assign new hash slots to receiving node. <a href="#">doc</a>	3.0.0	N
cluster bumpepoch	Advance the cluster config epoch. <a href="#">doc</a>	3.0.0	N
cluster count-failure-reports	Return the number of failure reports active for a given node. <a href="#">doc</a>	3.0.0	N
cluster countkeysinslot	Return the number of local keys in the specified hash slot. <a href="#">doc</a>	3.0.0	N
cluster delslots	Set hash slots as unbound in receiving node. <a href="#">doc</a>	3.0.0	N
cluster failover	Forces a replica to perform a manual failover of its master.. <a href="#">doc</a>	3.0.0	N
cluster flushslots	Delete a node's own slots information. <a href="#">doc</a>	3.0.0	N
cluster forget	Remove a node from the nodes table. <a href="#">doc</a>	3.0.0	N
cluster getkeysinslot	Return local key names in the specified hash slot. <a href="#">doc</a>	3.0.0	N
cluster info	Provides info about Redis Cluster node state. <a href="#">doc</a>	3.0.0	Y
cluster keyslot	Returns the hash slot of the specified key. <a href="#">doc</a>	3.0.0	Y
cluster meet	Force a node cluster to handshake with another node. <a href="#">doc</a>	3.0.0	N
cluster myid	Return the node id. <a href="#">doc</a>	3.0.0	Y
cluster nodes	Get Cluster config for the node. <a href="#">doc</a>	3.0.0	Y
cluster replicate	Reconfigure a node as a replica of the specified master node. <a href="#">doc</a>	3.0.0	N
cluster reset	Reset a Redis Cluster node. <a href="#">doc</a>	3.0.0	N
cluster saveconfig	Forces the node to save cluster state on disk. <a href="#">doc</a>	3.0.0	N
cluster set-config-epoch	Set the configuration epoch in a new node. <a href="#">doc</a>	3.0.0	N
cluster setslot	Bind a hash slot to a specific node. <a href="#">doc</a>	3.0.0	N
cluster slaves	List replica nodes of the specified master node. <a href="#">doc</a>	3.0.0	Y
cluster replicas	List replica nodes of the specified master node. <a href="#">doc</a>	5.0.0	Y
cluster slots	Get array of Cluster slot to node mappings. <a href="#">doc</a>	3.0.0	Y
readonly	Enables read queries for a connection to a cluster replica node. <a href="#">doc</a>	3.0.0	Y
readwrite	Disables read queries for a connection to a cluster replica node. <a href="#">doc</a>	3.0.0	Y

 stream

命令	描述	Redis 版本	是否支持
xinfo	Get information on streams and consumer groups. <a href="#">doc</a>	5.0.0	Y
xadd	Appends a new entry to a stream. <a href="#">doc</a>	5.0.0	Y
xtrim	Trims the stream to (approximately if '~' is passed) a certain size. <a href="#">doc</a>	5.0.0	Y
xdel	Removes the specified entries from the stream. Returns the number of items actually deleted, that may be different from the number of IDs passed in case certain IDs do not exist.. <a href="#">doc</a>	5.0.0	Y
xrange	Return a range of elements in a stream, with IDs matching the specified IDs interval. <a href="#">doc</a>	5.0.0	Y
xrevrange	Return a range of elements in a stream, with IDs matching the specified IDs interval, in reverse order (from greater to smaller IDs) compared to XRANGE. <a href="#">doc</a>	5.0.0	Y
xlen	Return the number of entries in a stream. <a href="#">doc</a>	5.0.0	Y
xread	Return never seen elements in multiple streams, with IDs greater than the ones reported by the caller for each stream. Can block.. <a href="#">doc</a>	5.0.0	Y
xgroup	Create, destroy, and manage consumer groups.. <a href="#">doc</a>	5.0.0	N
xreadgroup	Return new entries from a stream using a consumer group, or access the history of the pending entries for a given consumer. Can block.. <a href="#">doc</a>	5.0.0	N
xack	Marks a pending message as correctly processed, effectively removing it from the pending entries list of the consumer group. Return value of the command is the number of messages successfully acknowledged, that is, the IDs we were actually able to resolve in the PEL.. <a href="#">doc</a>	5.0.0	N
xclaim	Changes (or acquires) ownership of a message in a consumer group, as if the message was delivered to the specified consumer.. <a href="#">doc</a>	5.0.0	N
xautoclaim	Changes (or acquires) ownership of messages in a consumer group, as if the messages were delivered to the specified consumer.. <a href="#">doc</a>	6.2.0	N
xpending	Return information and entries from a stream consumer group pending entries list, that are messages fetched but never acknowledged.. <a href="#">doc</a>	5.0.0	N
xsetid	An internal command for replicating stream values. <a href="#">doc</a>	5.0.0	Y

🔗 transactions

支持率：5 / 5

命令	描述	Redis 版本	是否支持
discard	Discard all commands issued after MULTI. <a href="#">doc</a>	2.0.0	Y #
exec	Execute all commands issued after MULTI. <a href="#">doc</a>	1.2.0	Y #
multi	Mark the start of a transaction block. <a href="#">doc</a>	1.2.0	Y #
unwatch	Forget about all watched keys. <a href="#">doc</a>	2.2.0	N
watch	Watch the given keys to determine execution of the MULTI/EXEC block. <a href="#">doc</a>	2.2.0	N

# Memcached使用限制

## Memcached-使用限制

### 背景信息

云数据库 Redis 服务兼容 Memcached 1.4.19 版本的原生 API，详情请参见。

Memcached 1.4.19 版本原生API：<http://memcached.org/>。

支持版本：Memcached 1.4.19。

支持协议：text protocol。

### 支持的操作类型

- GET、GETS
- SET
- ADD
- REPLACE
- DELETE
- INCREMENT
- DECREMENT
- QUIT
- APPEND
- PREPEND
- TOUCH

### 操作说明

key 最大长度为1KB，val 最大长度为1MB。

## 最佳实践

### Redis设计规范与最佳实践

#### 🔗 Key名字如何设计？

#### 🔗 Key的大小

- 建议控制在**30个字节内**，尽可能**不超过128字节**，如果是MGET超过100个数据的场景，此时Key名字最好不超过20字节。
- 在能完整描述业务的前提下，尽量简化Key名的长度，比如“economic”简写为“ec”，“file path”简写为“fp”。

#### 🔗 Key的命名

- **不使用特殊字符**，比如空格、单引号、转义字符。
- **业务标识可以当做Key的前缀**，标识和含义内容用英文冒号（:）分隔，比如业务标识是“AID:15”，Key就可以写成：“AID:15:player:ranking”，方便遇到问题时根据业务标识尽快定位；**如果Key本身是拆分Key，可以使用“#”分隔**，比如“video\_info#olympic#2020-08-29#1”，“video\_info#olympic#2020-08-29#2”，提高可读性。
- 正确使用Redis的**hashtag**，默认符号的配置是“{}”，集群版是按照hashtag（第一个“{”和第一个“}”之间的内容）来进行路由的，比如“a{aa{xxx}bb}b”是按照“{aa{xxx}”转发到同一个数据分片。
- 由于大括号（{}）为Redis的**hash tag**语义，如果使用的是集群架构的实例，Key名称需要正确地使用大括号避免 引发数据倾斜，更多信息，请参见[keys-hash-tags](#)。

#### 🔗 Key的过期时间

- 建议Key都是需要设置**过期时间**的，尤其是秒杀、热门活动等**集中写入的Key**的场景，可以在设置过期时间的时候要**加上随机值**，让过期时间分散一些。

#### 🔗 Value如何设计？

### ☞ Value的大小

- **String类型的数据建议百字节以内，控制在1K以下**，最好不超过10K，如果是MGET或者使用Pipeline的场景，建议小于4K，Redis限制值是每个Value最大不超过512M。
- 过大的Value会引发数据倾斜、热Key、CPU性能被占满等问题，业务可以从设计源头上避免此类问题带来的影响。

### ☞ 使用的命令请求遵守哪些规范？

#### ☞ 控制数量：

- 对于**单个请求批量访问数据**的场景，建议**限制在100个Key以内**（如：MGET一次不能超过100个Key、HMGET一次不能超过100个FIELD）。
- 每个Pipeline批次下Key数量，建议**限制在200以内，最好控制在50以内**。
- 对于O(n)，O(log(n))以及更高复杂度的，控制元素个数以及使用频率，比如LRANGE 0 -1，HGETALL

#### ☞ 数据删除：

- 对于复合类型的DEL，如果是已知元素个数很多的大Key，4.0以上版本建议使用UNLINK方式删除，如果是低版本的Redis，可以使用HSCAN、SSCAN、ZSCAN。
- 对于4系以上版本Redis建议开启Redis的lazyfree配置，启用异步删除的功能

#### ☞ LUA 使用：

- **不建议**将复杂的EVAL脚本放在Redis执行，会导致占用CPU以及内存资源，带来性能下降，我们6系的版本支持基础命令以外的MODULE来满足业务场景。
- **不建议**LUA中包含事务以及redis.replication相关的操作。
- LUA脚本中变量需要保证是执行的时候给定脚本和参数，建议执行EVAL将LUA的脚本缓存在Redis中，通过EVALSHA的命令执行脚本内容。
- 执行EVALSHA收到返回是脚本不存在的时候，需要重新执行EVAL。

#### ☞ 消息队列或者通信：

- 发布订阅的业务场景，**要注意发布和订阅保持平衡**，包括订阅或者取消订阅的是已有的频道，避免出现热点。
- 一次订阅建议**控制在50个频道以内**，以免输出缓冲区过高触发客户端连接的断连。
- 用List类型做消息队列，**要注意生产和消费保持平衡**，并且通过LEN检查长度，通过LTRIM对队列的长度进行控制，及时截断。
- 当出现队列异常堆积以后，控制再次消费的速度，**避免大量堆积任务**同一时间消费导致输出缓冲区过高触发客户端连接的断连。

#### ☞ 慎用的命令：

- 如果不确认Key是Big-key，不确认当前Redis的内存使用量，**慎用DEBUG相关的命令**（如 DEBUG DIGEST/OBJECT等），避免造成阻塞，如果需要检查Key的大小，使用4系及以上高版本MEMORY命令检查，低版本建议使用复杂度是O(1)的命令检查，比如：STRLEN，HLEN（更多详见：<https://redis.io/commands/>）。
- **非必要不使用MONITOR命令**，如果需要，预先检查Redis的网络流量，在确保网络流量在10M/s以下，安全使用MONITOR命令。
- **非必要不使用KEYS命令**，如果需要，在确保自己的数据量控制在网络流量10M/s以下，安全使用KEYS命令。

## 避免大 Key 热 Key 实践指南

### ☞ 该如何设计数据结构以避免大Key和热Key？



老生常谈的大Key和热Key，需要拆、需要打散，如上，我们也给出各种类型Value建议大小，怎么拆呢？又如何打散呢？除了拆和打散，还有没有别的办法呢？下面举一些常见的场景案例。

业务使用场景

举例

用户任务记录（读文章，签到，看视频...），比如：

Key的设计最好以时间段划分

按月记录 uid:task:202209 而非将所有任务记录到一个Key：uid:task:all

消息队列任务记录

hash	uid:task:all（任务记录）	
Row（行数）	Field(任务名字)	Value（任务完成情况）
1	20220510:activity	.....
2	20220117:sign	.....
3	20210901:activity	.....
4	20201102:activity	.....
.....	.....	.....
1050	20190908:activity	.....
1051	20190415:activity	.....

hash	uid:task:202209（9月份任务记录）	
Row（行数）	Field(任务名字)	Value（任务完成情况）
1	reading:article	.....
2	sign	.....
3	01:activity	.....
4	02:activity	.....
.....	.....	.....
50	comment:3:times	.....
51	45:activity	.....

注意事项：

- Key名字要设置合理过期时间，避免一个Key中堆积很长时间历史数据。
- 如果预期Key的成员会随着时间增加而变多，建议“HSCAN，SSCAN.....”的方式定期清理。
- 消息队列要监控上下游的生产消费速度，或者监控队列长度，避免任务堆积。

业务使用场景

举例

根据考生分数查询可报考学校，比如：

将多个地域存一个Key拆开，改为多个地域对应多个Key，比如：

“全国考生分数学校对应信息（key）—— 省份\_分数段\_政史地（field）—— 可以报考学校的信息（value）”，

“可分类的数据”地域信息配置信息活动种类业务种类

可以优化成：“云南考生（key）—— 政史地:分数段（field）—— 可以报考学校的信息（value）”

hash	field	value
score:edu:info	liaoning_100_150_物化生	edu0_info
.....	.....	edu1_info
.....	liaoning_700_750_政史地	edu2_info
.....	.....	edu3_info
.....	yunnan_100_150_物化生	edu4_info
.....	.....	edu5_info
.....	yunnan_700_750_政史地	edu6_info
.....	.....	edu7_info
.....	qinghai_100_150_物化生	edu8_info
.....	.....	edu9_info
.....	qinghai_700_750_政史地	edu10_info

hash	field	value
liaoning:score:edu	100_150_物化生	edu0_info
.....	.....	edu1_info
.....	700_750_政史地	edu2_info

hash	field	value
yunnan:score:edu	100_150_物化生	edu3_info
.....	.....	edu4_info
.....	700_750_政史地	edu5_info

hash	field	value
qinghai:score:edu	100_150_物化生	edu6_info
.....	.....	edu7_info
.....	700_750_政史地	edu8_info

注意事项：

- 对于地域，时间，业务种类这种数据，设计Key的粒度要尽可能小，避免出现地域级别热点Key，利用分类将Key打散。

- 有时候设计Key的时候习惯用Key名字对应MySQL表名字，那这个Key的范围就容易很大，我们建议将Key拆开，用一些常用数据库字段对应到Key上。

业务使用场景

举例

“属性相关信息”

记录不同分数段对应的学校信息，比如：

上例中“edu0\_info”对应“100-150分数段物化生考生”可报考所有学校的信息，那么所有学校的信息本身是超过4KB的JSON，此时不建议Hash类型Value存储JSON，可以存JSON的Key名字“edu0\_info\_key”

用户信息

设备规格

聊天记录

商品规格

hash	field	value
liaojiang:score:edu	100_150_物化生	edu0_info_key
	.....	.....
	700_750_政史地	edu6_info_key

避免value过大，可以拆开

hash	field	value
edu0_info_key	edu_a	a_info
	.....	.....
	edu_k	k_info

hash	member	value
edu60_info_key	edu_p	p_info
	.....	.....
	edu_k	k_info

注意事项：

- 复合类型元素或者成员的Value建议小于4k，避免HGETALL，LRANGE 0 -1，SMEMBERS 取全集导致Redis压力过大。
- 建议将HGETALL换成HMGET，将LRANGE 0 -1换成LRANGE 0 50获取，SMEMBERS换成SISMEMBER，ZRange 0 -1换成ZRangeByScore加LIMIT。
- 元素或者成员Value建议是字符串，而不建议使用大JSON或者数组，可以将JSON存到单独的Hash或者String类型的Key中。

业务使用场景

举例

某次活动中奖用户记录，比如：

用户ID用1000取模，就可以将用户ID全集的1个Key拆成1000个Key，如果ID很多，可以用ID和ID前2个字符取模两次，按需分段uid#0#0，uid#1#1，.....，uid#999#99

“各种ID”  
粉丝列表  
视频收藏列表  
歌单歌曲列表  
文章信息库  
关注用户列表

list	uid
member	
00001	
...	
00999	
01000	
01001	
...	
11999	
12000	

uid % 1000

list	uid#0
member	
01000	
.....	
12000	

list	uid#1
member	
00001	
.....	
11001	

.....

list	uid#999
member	
00999	
.....	
11999	

注意事项：

- 利用Value在Key的名字上取模，常用在各种ID，即Value是整型，比如用户ID、书籍ID、文章ID。
- Key名字使用的分隔符号不建议使用空格和可能引起运算的符号（“-”，“.”，“+”，“=”，.....），建议用“#”，“:”，“\_”。

433

业务使用场景

举例

音乐播放历史，比如：

使用ZREMRANGEBYRANK 定期控制成员个数在1000左右，减轻ZREVRANK的O(log(N))开销

“排序场景”

音乐历史记录排序

文章取热门评论排序

歌手热门歌曲排序

zset Uid:music:history ( 音乐播放历史 )		
Row (行数)	Members(歌曲名)	Scores (听歌次数)
1	a1:FarAwayFromHome	2
2	5f:Apologize	34
3	4e:Apologize	4
4	0j:InANoteBook	90
.....	.....	.....
1510	9i:TheWayWeWere	43
1511	8u:Journey	21

```
127.0.0.1:6379> ZCARD Uid:music:history
(integer) 1511
删除低排名成员
127.0.0.1:6379> ZREMRANGEBYRANK Uid:music:history 0 510
(integer) 511
控制成员个数在1000以内
127.0.0.1:6379> ZCARD Uid:music:history
(integer) 1000
```

注意事项：

- 预期会经常排序的数据，建议控制元素（成员）个数在1000以内。
- ZREMRANGEBYRANK本身复杂度为O(log(N)+M)，注意使用频率，天级别或者整点或者业务低峰期执行为好。

业务使用场景

举例

屏蔽黑名单用户，比如：

大Set拆成单个String类型的Key，JSON格式的大String换Hash类型的Key

“一对多的关系”

作弊规则

反攻击规则限制

配置应用

权限管理

SET转换成STRING类型

set	member
屏蔽策略A	uid:0000001
	.....
	uid:2997788 uid:2997789

# 查询屏蔽策略A中有没有用户uid1，避免热点策略查询  
127.0.0.1:6379> SISMEMBER 屏蔽策略A uid1

→

string	value
uid:0000001	{[屏蔽策略A]}

# 查询用户uid1的屏蔽策略  
127.0.0.1:6379> GET uid1 屏蔽策略

string	value
uid:0000001	{[屏蔽策略A]}

# 查询用户uid1的屏蔽策略  
127.0.0.1:6379> GET uid1 屏蔽策略

注意事项：

- 使用合理的数据类型。

业务使用场景

举例

查询直播间的商品，比如：

同样的数据存多份备份，随机读一个，读不到再从DB回写

“热门场景”

热门直播间商品信息

热门赛事赛程信息

redis 中复制多个相同的key

string	value
直播间67909#0	{[商品A,商品B]}
直播间67909#1	{[商品A,商品B]}
.....	
直播间67909#199	{[商品A,商品B]}

redis miss可以从DB回写

DB层

直播间ID	商品
67909	{[商品A,商品B]}

注意事项：

- 适合读多写少的数据。
- 最佳实践是，分布式Redis中，备份数据份数可以用Redis集群分片数评估QPS分布，份数最好是集群分片数的一倍及以上。

业务使用场景	举例
页面“默认”展示	查询高考分数线的功能，比如： 需要显示类如【北京】【211】【北京邮电大学】，那么对于这种前端页面打开默认显示的数据不建议从Redis每次动态读取，可以业务侧本地缓存

注意事项：

- 如果是常用枚举，或者默认参数的数据,建议代码里保存。

业务使用场景	举例
“大文本” 大图文 图片内容 小说 长篇文章 博客 评论内容 客服对话	将小说文章压缩后存储

注意事项：

- JSON和XML换成轻量级的Protobuf。
- 使用Snappy或GZIP压缩。
- 预期Key可能过大，灌入数据的时候，使用EXPIREAT设置过期时间，不要集中到某个时间戳，最好加上随机时间。

业务使用场景	举例
所有场景	比如O(n)，O(log(n))以及更高复杂度的，控制元素个数，以及命令使用频率，时间复杂度详见： <a href="https://redis.io/commands/">https://redis.io/commands/</a>
注意命令时间复杂度及其使用频率	比如INCR的Key需要设置一天的过期时间，那么可以根据返回结果判断，如果已经大于1，就不需要重复 PEXPIREAT 比如Redis4系及以上可以使用MEMORY USAGE查询Key的内存占用，避免未知Key大小，执行DEBUG OBJECT 造成阻塞 比如 O(N) 的N是成员（元素）总数的慎用或者降低使用频率

注意事项：

列举常见又常用的高复杂度命令如下：

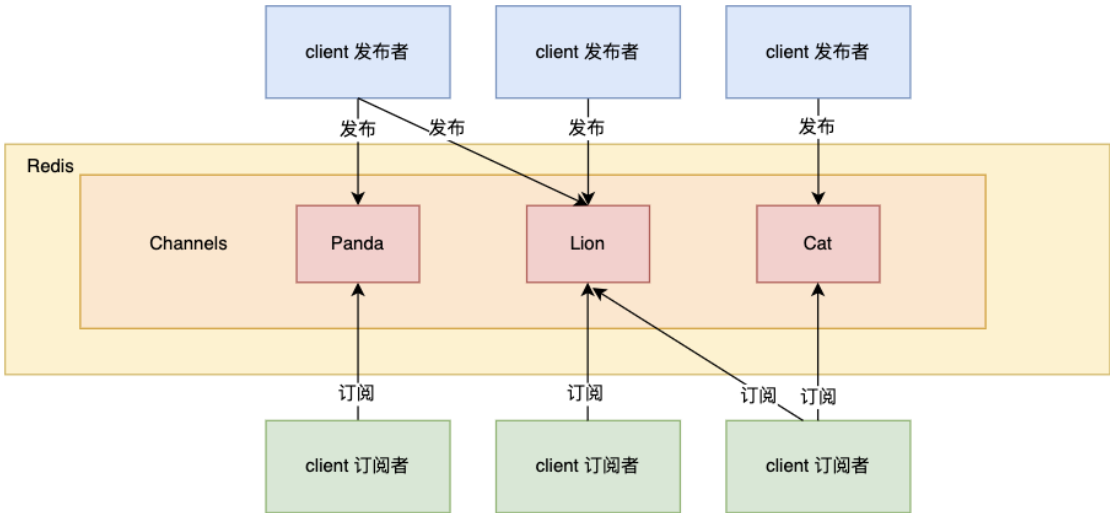
- List类型：LRANGE 范围取0 -1时慎用，LREM。
- Hash类型：HKEYS，HVALS，HGETALL。
- Set类型：SINTER / SINTERSTORE，SUNION / SUNIONSTORE，SMEMBERS。
- Sorted Set类型：ZRANGE(...) 范围取0 -1时慎用，ZREVRANGE(...)。ZREMRANGEBY(...)，ZUNION(...)。
- Geo 类型：GEOHASH，GEORADIUS(...)，GEOSEARCH(...)。
- 其他：KEYS，MONITOR，DEBUG。

## 消息发布订阅

### 🔗 功能介绍

Redis 发布订阅是一种消息传递模式，它允许用户将消息发送到频道，并且允许其他用户订阅该频道以接收这些消息。Channel的消息会放在订阅的客户端输出缓冲区中发送给订阅者，和传统的MQ不同，发布订阅的消息不会持久化，这个消息一旦发出，没订阅到的客户端也无法重试获取过去的消息。

发布者在一个连接内可以发布多个消息，也可以向多个Channels发布消息，但是订阅者一旦处于订阅状态，只能执行SUBSCRIBE，SSUBSCRIBE, PSUBSCRIBE, UNSUBSCRIBE, SUNSUBSCRIBE, PUNSUBSCRIBE。



### 🔗 适用场景：

可以设定对某一个Channel进行消息发布及消息订阅，当一个Channel上进行了消息发布后，所有订阅它的客户端都会接受到相应的消息。这一功能最明显的用途就是用作实时消息系统，比如：

1. **实时通知**：Redis发布订阅可以用于实时通知应用程序，例如在社交媒体上关注某人或某个主题，当有新消息时，可以向订阅者推送实时通知。
2. **聊天应用**：Redis发布订阅可以用于实现聊天应用中的实时聊天功能，发布者将消息发送到频道，订阅者可以实时接收这些消息。
3. **事件驱动系统**：Redis发布-订阅还可用于构建事件驱动系统。当某些事件发生时，可以将事件消息发布到相应的频道，各个订阅者可以根据自己的需求对这些事件做出响应，比如数据库高可用的功能也可以依赖这个实现。

#### 🔗 使用注意事项

1. 避免发布者数量过多：如果发布者数量过多，可能会导致订阅者无法处理所有消息，建议每个Channel的发布者<10。
2. 避免订阅过多频道导致输出缓冲区打满，减少使用通配符匹配，比如：“Channe\*”，如果需要订阅多个频道，建议控制在50个以内，控制Redis的输出流量在20MB/s。
3. 对消息顺序敏感的业务，如果订阅了多个频道，或者一个频道有多个发布者，订阅者需要处理消息的顺序。
4. 要注意发布和订阅保持平衡，不要出现只订阅不发布，或者只订阅没有发布，此外，不要持续订阅和取消不存在的Channel。

#### 🔗 命令的支持与使用规范

1. 命令支持情况：详情参考[Proxy命令支持情况](#)
2. 开源框架Redisson的分布式锁redis-cluster模式的实现依赖在LUA中的使用发布订阅功能，但是由于Redis集群版对LUA脚本有严格使用要求，故不支持框架中分布式锁的功能。

#### 🔗 建议的连接池配置

因为订阅场景的客户端连接占用时间过长，可能因为连接池不够用影响整体吞吐，建议连接配置如下：

1. **最大连接数**：在数据库集群能承载的连接数安全限度内，建议将最大连接数设置略高。（集群基本信息可查看连接数规格）
2. **最大空闲连接数**：建议设置最大空闲连接数，以保证客户端其他非订阅请求有可用连接。
3. **最小空闲连接数**：建议设置最小空闲连接数，以保证订阅客户端连接可用。
4. **空闲连接回收时长**：根据业务的订阅消息时效以及超时时间，合理设置空闲连接回收时长，确保整体连接池有可用连接。
5. **重连次数**：建议将重连次数设置为1-3次，以确保连接池中的连接在Redis服务端故障时能够快速恢复连接。

## 视频专区

### 操作指南

#### 🔗 Redis 创建实例教程

- 本视频介绍如何创建一个 Redis 实例。





#### 🔗 Redis 连接实例教程

- 本视频介绍如何通过云服务器 BCC 和 Redis 创建实例连接。



## 常见问题

### 性能类问题

#### 🔗 1.性能调优参考

- [如何处理Redis集群数据倾斜](#)
- [排查Redis实例流量使用率高的问题](#)
- [排查Redis实例内存使用率高的问题](#)
- [排查Redis实例CPU使用率高的问题](#)
- [常见Latency（时延）事件的处理建议](#)

#### 🔗 2.已经使用了存储服务，为什么还要使用缓存服务？

- **加速数据访问（高性能读写）：**数据存储在内存（RAM）中，读写速度远超磁盘数据库（如 MySQL），适合高并发场景。例如：秒杀、实时排行榜、用户会话（Session）等高频访问数据。
- **减轻后端数据库压力：**将频繁查询的数据库结果缓存在 Redis 中，减少直接访问数据库的次数。避免“雪崩效应”：当数据库压力激增时，缓存层可充当缓冲，保护后端数据库不被压垮。

#### 🔗 3.云数据库 Redis实例是否支持选择CPU？

云数据库 Redis 实例规格支持**2核**、**4核**两种CPU。

#### 🔗 4.云数据库 Redis 是否应该开启持久化？对性能的影响？

- 开启aof参数持久化对性能存在一定影响。
- 云数据库 Redis 提供“内存和磁盘”方式存储数据，持久化到磁盘默认是使用aof的方式，如果有增量数据每一秒中会写磁盘一次，当数据增长达到100%时还会触发一次aof-rewrite，这些都会导致处理用户请求增加一定的时延。
- 对某些纯缓存用户，持久化到磁盘不是必须的，此时可以选择关闭磁盘持久化来提升性能。

### 安全管控保障

百度云数据库Redis是兼容开源Redis协议标准、提供内存加硬盘混合存储的数据库服务，支持主从、集群和读写分离架构，



从网络、备份、容灾等方面，提供全面的安全加固功能来保障您的数据安全。

#### 🔗 网络攻击防护

当您使用公网访问实例时，可能会遭受DDoS攻击。百度云为Redis实例提供免费的DDoS基础防护，实时监测并清除大流量攻击。

#### 🔗 客户端访问限制

**白名单** 您可以配置白名单，将客户端的IP地址或IP地址段添加到Redis实例的白名单中后才能通过该客户端访问Redis实例，实现从访问源进行风险控制。更多信息请参见[设置白名单](#)。 **安全组**

- 安全组是在VPC网络内为BCC实例和DCC专属实例、负载均衡实例、云数据库实例中创建的安全防火墙，可定义IP + 端口的入站和出站访问策略。增加安全组时需要确保安全组规则包含白名单中所有IP，否则对应IP无法访问。
- 为了方便客户端管理、提高连接的安全性，云数据库Redis支持配置安全组。具体操作可参考[配置安全组](#)
- 更多安全组信息资料参考：[私有网络安全组](#)、[安全组配置实践](#)。

**数据库账号管理** 数据库账号是访问Redis实例的凭证，您可以通过Redis控制台或API创建、管理数据库账号的读、写权限，确保访问安全可靠，更多信息请参见[创建账号](#)、[修改账号权限](#)。

#### 🔗 网络隔离

云数据库Redis支持通过专有网络和公网访问，建议您使用专有网络，专有网络的安全性更高。

**专有网络（VPC）** VPC（Virtual Private Cloud）是私有网络环境，通过底层网络协议，在网络二层完成网络隔离，具备安全可靠、灵活可控、简单易用的特性和较强的可扩展性。更多信息请参考[专有网络](#)。 **公网（Internet）** 您也可以为Redis实例申请公网地址，并通过公网访问Redis实例，更多信息请参见[绑定公网IP](#)。

#### 🔗 备份恢复

**数据备份** 云数据库Redis支持以下持久化策略（备份方式）：

- RDB（Redis Database）持久化**：周期性地为引擎中的数据创建快照，生成RDB文件，保存到磁盘中，实现数据持久化。默认RDB持久化策略为：每天自动生成RDB快照，默认保存时间为7天。如果需要更长时间的备份存档（例如监管或信息安全需要），您可以将备份文件下载到本地进行存储。
- AOF（Append-only-file）持久化**：以日志的形式记录所有的写入类操作，例如SET。服务重启时，可以重新执行AOF文件中的操作来恢复数据。 [数据恢复](#)
- 从备份集恢复至新实例、原实例**：支持根据指定的RDB备份文件创建新实例、覆盖原实例，可用于数据恢复、快速部署业务或数据验证等场景。
- 通过数据闪回按时间点恢复数据**：Redis除支持通过RDB快照执行数据备份和恢复以外，优化了基于AOF（Append-only-file）的持久化机制，实现AOF增量归档，可以将实例整体Key的数据恢复至某个秒级的时间点。开启该功能后，在备份文件的保存期内，您可以恢复指定时间点（精确到秒级）的Redis数据，可最大限度地避免误操作带来的数据损失，或者在频繁回档的业务场景快速完成数据切换。

#### 🔗 实例容灾

**双可用区** 云数据库Redis标准版和集群版提供跨双机房的同城容灾架构。如果业务为单一地域部署，且对容灾要求较高，可在创建云数据库Redis实例时，选择支持同城容灾的双可用区。

**\*\*异地多活\*\*** 云数据库Redis支持多地域数据灾备，可轻松支持异地多个站点同时对外提供服务的业务场景。更多信息请参考[多活实例组](#)。

## 费用相关问题

#### 🔗 1. 产品选型参考

- 大规模数据、高QPS要求，推荐Redis容量型（pegaDB）和Redis内存型都测试下，看下效果对比。
- 大规模数据、低QPS要求，推荐Redis容量型（pegaDB）。
- 小规模数据，推荐使用Redis内存型。

## ☞ 2.云数据库 Redis 如何计费，有哪些套餐？

目前支持预付费、后付费两种模式。

- 预付费即包年、包月付费，在创建实例时需要支付费用。适合业务稳定的长期需求，费用比后付费更低，且购买时间越长，折扣越多（一年6折、两年5折、三年4折）。
- 后付费即小时计费，在北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后1小时内。
- 适合业务瞬间有较大变化的场景，资源使用完即可立即释放，降低成本。

## ☞ 3.资源配置发生更改后，资费是如何变更的？

### 预付费转后付费

- 预付费转后付费后，计费方式变更将在预付费合同到期后立即生效。
- 预付费转后付费后，合同未到期前，实例的支付方式状态为「预付费（待变更）」，此阶段可取消「计费变更」。
- 预付费转后付费前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于**100元**，避免计费方式变更失败。
- 对于已启用自动续费的预付费实例，不能进行计费变更。

### 后付费转预付费

- 实例的计费方式可以从后付费变更为预付费，且计费方式变更成功后，将立即按照预付费方式计费。

## Lua脚本使用规范与注意事项

云数据库Redis 支持 Lua 相关命令，通过 Lua 可轻松实现更复杂的业务逻辑，并原子性地执行。本文介绍通过 Redis 使用 Lua 脚本的使用规范和注意事项。

## ☞ Lua 相关命令使用方法

eval eval 命令执行给定的脚本，其语法规则如下：

```
1.EVAL script numkeys [key [key ...]] [arg [arg ...]]
```

- script 为需要执行的脚本
- numkeys 为传入脚本的 key 的数量，在 lua 脚本中可以使用 KEYS这个预定义的全局变量获取到传入的 keys
- [key [key ...]] 为传入的 key，这里 key 的数量需要和 numkeys 保持一致
- [arg [arg ...]] 为传入 lua 脚本的参数

下面是在 Python 中使用 eval 命令执行 lua 脚本的例子：

```
1. import redis
2. lua = """
3. if redis.call("type", KEYS[1]) == 'string' and redis.call("GET", KEYS[1]) == ARGV[1]
4. then
5. redis.call("SET", KEYS[1], ARGV[2])
6. return 1
7. end
8. return 0
9. """
10. conn = redis.Redis("127.0.0.1", 6379)
11. conn.eval(lua, 1, "foo", "a", "b")
```

上面的例子中，实现 CAS（compare and set）功能，当 KEYS[1]的值等于 ARGV[1]时，将其此 key 的值设置为 ARGV[2],如果成功执行了设置，则返回 1，否则返回 0。

## ☞ eval\_ro

eval\_ro 命令的语法和 eval 一样，区别在于 eval\_ro 表明将要执行的 lua 脚本不会修改 redis 中的数据，其中只会执行读操作

(read only) 。

```
1.EVAL_RO script numkeys [key [key ...]] [arg [arg ...]]
```

如果确定 lua 脚本中只会执行读操作，那么可以使用 eval\_ro 来执行它，这样使用云数据库Redis 集群时，可能将其发送至从节点执行，以降低主节点的压力。

#### 🔗 script load

当 redis 执行过一个 lua 脚本后，会将此脚本缓存下来，当用户需要再次执行此 lua 脚本时，可以不必再次提供此脚本。可以使用 script load 命令将 lua 脚本载入到 redis 节点中，后续可以使用一个表示该脚本的 id 来执行此脚本。

在云数据库Redis 集群版中，执行 script load 后，会将此 lua 脚本载入到所有的分片中，只要有一个分片载入失败，script load 命令就会失败。

下面是在 Python 中使用 script load 的例子：

```
1. import redis
2. lua = "..."
3. conn = redis.Redis("127.0.0.1", 6379)
4. sha = conn.script_load(lua)
5. assert sha == "2fcf32663b5faac34faebe59525c14b035c12705"
```

载入 lua 脚本时，redis 会使用 sha1 算法对该 lua 脚本执行 hash 操作，得到一个字符串，后续可以使用此字符串来指代此 lua 脚本。

#### 🔗 evalsha

使用 script load 载入脚本后，可以得到一个表示该 lua 脚本的 sha 值，而后可以使用 evalsha 来执行此 lua 脚本。evalsha 的语法如下：

```
EVALSHA sha1 numkeys [key [key ...]] [arg [arg ...]]
```

和 eval 唯一的区别是，这里使用 sha1 值指代此前载入的 lua 脚本。

#### 🔗 evalsha\_ro

和 eval\_ro 类似，evalsha 也有其只读版本 evalsha\_ro。

#### 🔗 script exists

给定一个（或多个）脚本的 SHA1，script exists 返回每个 SHA1 对应的脚本是否已缓存在当前 Redis 节点中。若脚本已存在，则返回 1，否则返回 0。在云数据库Redis 集群版中，script exists 会被转发给所有分片，一旦有任何一个分片中不存在提供的脚本，就会返回不存在。

#### 🔗 script flush

清空所有分片中缓存的 lua 脚本。

#### 🔗 script kill

停止正在运行的 Lua 脚本。

#### 🔗 lua脚本使用限制

##### 🔗 至少需要提供一个 key

集群版中，Proxy 需要有 key 的信息才能知道将该 lua 脚本转发至哪一个分片中，因此在执行 EVAL 和 EVALSHA 命令时，必须提供至少一个 key，即 numkeys 不能为 0。

```
1.EVAL script numkeys [key [key ...]] [arg [arg ...]]
```

比如像下面这样的命令是不支持的：

```
1.eval 'return redis.call("GET", "ABC")' 0
```

##### 🔗 若传入多个 key，需保证其落在同一个 slot 上

eval 支持传递多个 key，lua 脚本中涉及对这多个 key 的操作。如果传入了多个 key，则要保证这些 key 落在同一个 slot 中。因为 lua 脚本最终会在某个分片的 redis 节点中执行，如果其中涉及的多个 key 不再同一个 slot 上，那么多个 key 对应的数据可能在不同的节点中，这样自然是不符合预期的。在云数据库Redis 集群版中，Proxy 会检查 numkeys 的值是否为 0，若为 0，则 eval/evalsha/eval\_ro/evalsha\_ro 命令会失败。如果 LUA 脚本中涉及到对多个 key 的操作，可以考虑对业务做适当改造，使用 hashtag 保证这多个 key 落在同一个 slot 上：

```
eval "return {KEYS[1], KEYS[2]}" 2 user_1234 user_2345 # NOT OK
eval "return {KEYS[1], KEYS[2]}" 2 {user}_1234 {user}_2345 # OK
```

🔗 不要在 lua 脚本中对 key 硬编码

所有的 key 均应该通过参数传递，不应该在 lua 中硬编码 key，虽然这样不会报错，但该 key 会被写入本不属于它的分片中。

```
eval 'return redis.call("GET", "ABC")' 0 # NOT OK
eval 'return redis.call("GET", KEYS[1])' 1 ABC # OK
```

🔗 常见报错

错误信息	原因
wrong number of arguments for 'eval' command	eval 命令没有正确地提供一个以上的 key
No matching script. Please use EVAL.	当前执行的 lua 脚本不存在，需要下您执行载入

常见报错与解决方案

当使用 Redis 时，可能会遇到各种报错，这些报错可能由多种原因引起，例如网络问题、配置错误、资源限制等。以下是一些常见的 Redis 报错及其可能的解决方案。目前内容还有欠缺，后续会继续补充整理。

错误类型	报错信息	报错原因	解决方法
连接相关错误	NOAUTH Authentication required	Redis 实例设置了密码鉴权，但客户端没有提供密码或提供了错误的密码。	使用正确的账号密码进行访问。
	WRONGPASS invalid username-password pair , ERR invalid password	密码错误。	确保提供正确的密码。
	ERR client ip is not in whitelist , ERR no permission to connect	客户端 IP 未添加至白名单。	查看白名单配置，确保本机 IP 已经加入白名单。
	permission denied for 'xxx'	该客户端无权限执行 xxx 命令。	检查账号管理配置中当前账号的读写权限。
命令相关错误	exceed qps quota , exceed read qps quota , exceed write qps quota	当前客户端达到 QPS 限制。	确认集群中流量是否均匀分布在不同的 Proxy 上，如果负载不均衡，可配置连接池将流量分散在多个 Proxy 上。如果确认流量已经均匀分散在所有的 Proxy 上，则说明流量太高，客户端需要做流量控制，或者发起扩容。
	WRONGTYPE Operation against a key holding the wrong kind of value	命令使用错误，例如对 String 数据类型执行 HASH 命令。	修改错误代码或命令。
	ERR wrong number of arguments for 'xxx' command	命令参数数量错误。	确保按照命令的语法正确使用命令。
	ERR unknown command 'xxx'	执行了不支持的命令。	检查命令是否存在或是否被禁用。
	ERR DB index is out of range	select 命令超出了 databases 设置的范围。	使用多 db 时，不要超出当前配置的 db 范围。

企业上云评估参考

1.根据QPS、平均KV长度、数据量评估

1.1 根据QPS、平均KV长度计算需要分片数量

根据如下表格, 大致估算每个分片可以承担的QPS。

- 其中KV长度是key\_size+value\_size。
- 分片数=总QPS/{单分片最大QPS}。

平均KV长度	命令复杂程度	单分片QPS参考
16-64 Bytes	低(以简单的kv存储为主)	12w
16-64 Bytes	中(使用部分O(n)命令, 如hgetall)	5w
16-64 Bytes	高(使用大量O(n)复杂度命令, 使用复杂lua脚本等)	2w
64-256 Bytes	低(以简单的kv存储为主)	10w
64-256 Bytes	中(使用部分O(n)命令, 如hgetall)	5w
64-256 Bytes	高(使用大量O(n)复杂度命令, 使用复杂lua脚本等)	2w
256-1024 Bytes	低(以简单的kv存储为主)	4w
256-1024 Bytes	中(使用部分O(n)命令, 如hgetall)	2w
256-1024 Bytes	高(使用大量O(n)复杂度命令, 使用复杂lua脚本等)	1w
1024-4096 Bytes	-	5k-1w

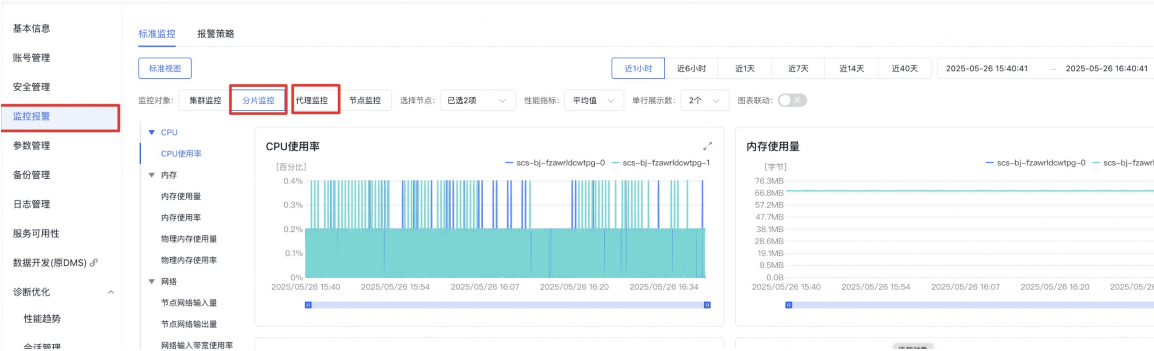
1.2 根据数据量、分片数计算每个分片规格

单分片规格容量 = {数据量} / 0.7 / {分片数}。

- 注意：由于Redis内存中数据结构本身也会占用一定空间, 实际使用中内存用量一般会大于估算量。

2.通过业务压测观察性能指标调整实例规格、分片数量

2.1可以通过实例详情页面中的“监控报警”页面, 实时观察压测时实例性能表现。



2.2重点需要关注的是“分片监控”、“代理监控”两种监控对象, 分别代表Redis节点以及Proxy节点的性能表现。

监控对象	指标名称	指标值判断
分片监控	CPU使用率	• 均值 &gt; 70%: 需要增加分片     • 50% &lt; 均值 &lt; 70% : 考虑增加分片     • 均值 &lt; 20%: 考虑减少分片
分片监控	网络输出带宽使用率	均值 > 25%(约对应50MB/s): 考虑增加分片(即使CPU用量低, 也需要考虑增加分片)
分片监控	网络输入带宽使用率	均值 > 10%(约对应20MB/s): 考虑增加分片(即使CPU用量低, 也需要考虑增加分片)
分片监控	内存使用率	均值 > 70%: 考虑增加规格(增加内存)
代理监控	CPU使用率	• 均值 &gt; 70%: 需要增加分片     • 50% &lt; 均值 &lt; 70% : 考虑增加分片     • 均值 &lt; 20%: 考虑减少分片
代理监控	采样周期内慢查询次数	均值 > 0: 在代理、分片CPU使用率较低的场景, 不应出现慢查询, 可通过 <a href="#">热key分析</a> , <a href="#">大key分析</a> 等查看是否有热key、大key。
代理监控	平均执行时延	具体阈值应当由业务确定, 当延时不满足业务需求时, 可从如下几方面进行排查:       • CPU、网络带宽等指标是否较高, 考虑增加分片     • 如果压测所在节点与实例不在一个可用区, 时延会增加。     • 通过<a href="#">热key分析</a>, <a href="#">大key分析</a>等查看是否有热key、大key。
代理监控	P99执行时延	具体阈值应当由业务确定, 当延时不满足业务需求时, 可从如下几方面进行排查:       • CPU、网络带宽等指标是否较高, 考虑增加分片     • 如果压测所在节点与实例不在一个可用区, 时延会增加。     • 通过<a href="#">热key分析</a>, <a href="#">大key分析</a>等查看是否有热key、大key。
代理监控	网络输出带宽使用率	默认proxy数量=分片数量, 未开启副本只读场景, 平均一个proxy上流量=主节点的流量, 当开启副本只读后, proxy上读流量会对应增加, 一般不应超过40%(80MB/s)。
代理监控	网络输入带宽使用率	默认proxy数量=分片数量, 未开启副本只读场景, 平均一个proxy上流量=主节点的流量。当开启副本只读后, proxy上读流量会对应增加, 一般不应超过40%(80MB/s)。
代理监控	代理上流量均衡:  • CPU使用率  • ProxyQPS  • 实例连接数	CPU使用率、ProxyQPS、实例连接数三个指标, 不同代理上应当尽量一致; 如果差距较大说明不同代理上流量不均衡, 可能会因为单点的高负载导致整体的可用性下降如果出现代理上流量不均衡。

- **注意事项** : 增删分片时需要同时考虑分片监控和代理监控的CPU使用率。



- 代理CPU使用率 >> 分片CPU使用率: 业务流量对代理压力大(一般为kv小, 客户端多的场景), 可联系OP增加代理数量, 无需增加分片数量。
- 代理CPU使用率 << 分片CPU使用率: 业务使用高复杂度命令, 不能因为代理CPU使用率低而减少分片。
- 调整实例规格、分片数量可参考: [变更配置](#)

# 服务等级协议SLA

## Redis服务等级协议SLA

协议生效时间: 2019 年 1 月 15 日

本服务等级协议 (Service Level Agreement, 简称“SLA”) 规定了百度智能云向用户提供的云数据库 Redis (Simple Cache Service for Redis, 以下简称“Redis”) 的服务可用性等级指标及赔偿方案。

### 1. 定义

**服务周期:** 一个服务周期为一个自然月。

**服务周期总分钟数:** 按照服务周期内的总天数 \* 24 (小时) \* 60 (分钟) 计算。

**服务不可用分钟数:** 在某一分钟内, 用户所有试图与指定的云数据库Redis实例建立连接的连续尝试均失败, 则视为该分钟内该云数据库Redis实例服务不可用。 在一个服务周期内云数据库Redis实例不可用分钟数之和即服务不可用分钟数。

**月度服务费用:** 用户在一个自然月中就单个Redis实例所支付的服务费用总额, 如果用户一次性支付了多个月份的服务费用, 则将按照所购买的月数分摊计算月度服务费用。

### 2. 服务可用性

#### 2.1 服务可用性计算公式

服务可用性以单个实例为维度, 按照如下方式计算:

$$\text{服务可用性} = \left( \frac{\text{服务周期总分钟数} - \text{服务不可用分钟数}}{\text{服务周期总分钟数}} \right) \times 100\%$$

#### 2.2 服务可用性承诺

Redis承诺服务可用性不低于 99.99%, 即用户每月业务可用时间应为 $30 \times 24 \times 60 \times 99.99\% = 43195.68$  (分钟), 即存在 $43200 - 43195.68 = 4.32$  (分钟) 的不可用时间。 如Redis未达到上述服务可用性承诺, 用户可以根据本协议第三条约定获得赔偿。赔偿范围不包括以下原因所导致的服务不可用时间:

- (1) 百度智能云预先通知用户后进行系统维护所引起的, 包括割接、维修、升级和模拟故障演练。
- (2) 任何百度智能云所属设备以外的网络、设备故障或配置调整引起的。
- (3) 用户的应用程序受到黑客攻击而引起的。
- (4) 用户维护不当或保密不当致使数据、口令、密码等丢失或泄漏所引起的。
- (5) 用户不开启监控告警致使系统故障无法及时暴露引起的。
- (6) 用户执行清空数据等操作造成数据丢失引起的。
- (7) 用户的疏忽或由用户授权的操作所引起的。
- (8) 用户未遵循百度智能云产品使用文档或使用建议引起的。
- (9) 不可抗力或者其他意外事件引起的。
- (10) 其他非百度智能云原因所造成的不可用。
- (11) 购买的实例为单副本形态。

### 3. 赔偿方案

#### 3.1 赔偿标准

每个Redis实例按单实例月度服务可用性, 按照下表中的标准计算赔偿金额, 赔偿方式仅限于用于购买Redis产品的代金券, 且赔偿总额不超过未达到服务可用性承诺当月用户就该Redis实例支付的月度服务费用。



服务可用性	赔偿代金券金额
低于99.99%但等于或高于99.00%	月度服务费用的15%
低于99.00%但等于或高于95.00%	月度服务费用的30%
低于95.00%	月度服务费用的100%

3.2 赔偿申请时限

用户可以在每月第五（5）个工作日后对上个月没有达到可用性的实例提出赔偿申请。赔偿申请必须限于在Redis实例没有达到可用性的相关月份结束后两（2）个月内提出。超出申请时限的赔偿申请将不被受理。

4. 其他

- (1) 在法律法规允许的范围内，百度智能云负责对本协议进行解释说明。
- (2) 百度智能云有权对本SLA条款作出修改。如本SLA条款有任何修改，百度智能云以网站公示或发送邮件的方式通知您。如您不同意百度智能云对SLA所做的修改，您有权停止使用Redis服务，如您继续使用Redis服务，则视为您接受修改后的SLA。
- (3) 本协议项下百度智能云Redis服务对于用户所有的通知均可通过网页公告、站内信、电子邮件、手机短信或其他形式等方式进行；该类通知于发送之日视为已送达收件人。百度智能云不对用户承担因此产生的任何损失。
- (4) 本协议的订立、执行和解释及争议的解决均应适用中国法律并受中国法院管辖。如双方就本协议内容或其执行发生任何争议，双方应尽量友好协商解决；协商不成时，任何一方均可向北京市海淀区人民法院提起诉讼。
- (5) 本协议构成双方对本协议之约定事项及其他有关事宜的完整协议，除本协议规定的之外，未赋予本协议各方其他权利。
- (6) 如本协议中的任何协议无论因何种原因完全或部分无效或不具有执行力，本协议的其余协议仍应有效并且有约束力。
- (7) 关于用户约束条款，详见《[用户服务协议](#)》中的"用户的权利与义务"相关条款内容。
- (8) 关于服务商免责条款，详见《[用户服务协议](#)》中的"免责声明"相关条款内容。

云数据库Redis服务条款

更新时间：2024-10-18

本服务条款是百度智能云网站（cloud.baidu.com，以下简称为“百度智能云”）的经营者北京百度网讯科技有限公司（以下简称为“本公司”），与用户（以下称为“您”）共同缔结的针对百度智能云的云数据库 Redis（Simple Cache Service for Redis，以下简称“本产品”）服务适用的、对双方具有约束力的有效契约。

1. 总则

1.1 您确认

您在使用本服务之前，已经充分阅读、理解并接受本服务条款的全部内容，一旦您选择使用云数据库 Redis 服务，即表示您同意遵循本服务条款之所有约定。本服务条款内容同时包括百度智能云可能不断发布的关于所提供服务的协议、业务规则、保密协议等内容。上述内容一经正式发布，即为本服务条款不可分割的组成部分，请您一并遵守。

1.2 本服务条款项下

百度智能云对于您所有的通知均可通过网页公告、站内信、电子邮件、手机短信或其他形式等方式进行；该等通知于发送之日视为已送达收件人。

1.3 您同意

百度智能云有权随时对本服务条款及相应的服务规则内容进行单方面的修改，并有权以消息推送、网页公告等方式予以公布，而无需另行单独通知您；若您在收到本服务条款及其相应的服务规则变更的消息或公告内容后继续使用本产品的，表示您已充分阅读、理解并接受修改后的协议内容，也将遵循修改后的协议内容使用本产品；若您不同意对本服务条款及相应规则所做的修改，您应立即停止使用本服务。

2. 服务内容

2.1 本服务条款中的“服务”：指百度智能云向您提供cloud.baidu.com网站上所展示的云数据库 Redis服务及相关的技术网络支持服务。

2.2 服务费用：根据您选择的服务类型，在购买页面列明公示服务价格并支付。

2.3 您同意：若您使用本产品不符合本服务条款，百度智能云在经由通知、举报等途径发现时有权做出独立判断，且有权在无需事先通知您的情况下，立即终止向您提供部分或全部云数据库服务。您如果散布、传播反动、色情或其他违反国家法律、法规的信息，云数据库中存储的代码可能作为您违法的证据；因您实施上述行为，导致任何第三方提出索赔要求或衍生的任何损害或损失，由您自行承担全部责任。给百度智能云造成损失的，百度智能云保留向您追偿的权利。

2.4 若百度智能云经由通知、举报等途径，发现您在使用云数据库服务时违反任何本服务条款的规定，百度智能云有权要求您改正或直接采取一切百度智能云认为必要的措施，以减轻您不当行为造成的影响。

### 3. 用户的权利与义务

3.1 您同意遵守本服务条款以及服务展示页面的相关管理规范及流程。

3.2 您保证不会利用技术或其他手段破坏或扰乱本网站及百度智能云上其他客户。

3.3 您应按照百度智能云页面提示及本服务条款的约定支付相应的服务费用。

3.4 您保证使用本服务时将遵从国家、地方法律法规、行业惯例和社会公德，不会利用本公司提供的服务进行存储、发布、传播如下信息和内容：

- (1) 违反国家法律法规政策的任何内容（信息）；
- (2) 违反国家规定的政治宣传和/或新闻信息；涉及国家秘密和/或安全的信息；
- (3) 封建迷信和/或淫秽、色情、下流的信息或教唆犯罪的信息；
- (4) 博彩有奖、赌博游戏；
- (5) 违反国家民族和宗教政策的信息；
- (6) 妨碍互联网运行安全的信息；
- (7) 侵害他人合法权益的信息和/或其他有损于社会秩序、社会治安、公德的信息或内容。

3.5 您承诺不得为他人发布上述不符合国家规定和/或服务条款约定的信息内容提供任何便利，包括但不限于设置URL、BANNER链接，发送邮件等。您承认本公司有权在您违反上述约定时有权终止向您提供服务并不予退还任何款项，因您上述行为给本公司造成损失的，您应予赔偿。

3.6 当您在违反本服务条款的情况下，本公司有权拒绝继续为您提供任何服务。

3.7 您应自行对因使用本服务而存储在百度智能云数据库的各类数据等信息，在本服务之外，采取合理、安全的技术措施，确保其安全性和完整性，并对自己的行为（包括但不限于欠费，自行安装软件、采取加密措施或进行其他安全措施等）所引起的结果承担全部责任。

3.8 您在使用本产品时须遵守本服务条款以及百度智能云的相关要求，不得利用本服务从事包括但不限于以下行为，也不得为以下行为提供便利（包括但不限于以下内容）：

- (1) 进行博彩、赌博游戏、“私服”、“外挂”等非法互联网活动；
- (2) 发布、传送、传播电子邮件广告、垃圾邮件（SPAM）；
- (3) 发布、传送、传播不受欢迎的或者未经请求的电子邮件、电子广告或包含反动、色情等有害信息的电子邮件；
- (4) 将本产品用作磁盘空间出租；
- (5) 进行任何破坏或试图破坏网络安全的行为，包括但不限于以病毒、木马、恶意代码、钓鱼等方式，采取对其他网站、服务器进行恶意扫描、非法侵入系统、非法获取数据等；
- (6) 进行任何改变或试图改变本服务提供的系统配置或破坏系统安全的行为；
- (7) 不可大量占用，且不得导致程序或进程大量占用百度智能云的云计算资源（如云服务器、带宽、存储空间）所组成的平台中服务器的内存、CPU或带宽资源；并给百度智能云其他用户的服务器、网络、产品/应用带来严重的、不合理的负荷，影响百度智能云与互联网、特定网络、内部正常的联系；或导致百度智能云的产品与服务或者其他用户的服务宕机、关机或者用户的产品/应用不可访问等；
- (8) 利用技术或其他手段破坏、扰乱本服务的运营或他人对本服务的使用；
- (9) 以任何方式干扰或企图干扰百度任何产品或任何部分、功能的正常运行，或者制作、发布、传播上述工具、方法等；
- (10) 进行任何可能违反诚实信用、公序良俗、公德的行为；
- (11) 若您因从事任何业务，导致您频繁遭受攻击（包括但不限于DDoS攻击）且未及时更正您的行为或未根据百度要求消除影响，从而对本服务平台或其他本服务使用者产生威胁的，一经发现，本产品将被强制停止，账号也会立即冻结。
- (12) 任何违反中华人民共和国现行有效法律/法规/规章等法律禁止性行为，包括但不限于：

反对宪法所确定的基本原则的。危害国家安全，泄露国家秘密，颠覆国家政权，破坏国家统一的。损害国家荣誉和利益的。煽动民族仇恨、民族歧视，破坏民族团结的。破坏国家宗教政策，宣扬邪教和封建迷信的。散布谣言，扰乱社会秩序，破坏社会稳定的。散布淫秽、色情、赌博、暴力、凶杀、恐怖或者教唆犯罪的。侮辱或者诽谤他人，侵害他人合法权益的。

## 4. 百度智能云的权利与义务

4.1 百度智能云保留在您违反国家、地方现行法律法规规定或违反本服务条款的情况下单方终止为您提供服务的权利，并有权配合国家上级主管部门，对您进行审查，并且在任何情况下，百度智能云对任何间接、偶然、特殊及继起的损害不承担赔偿责任。

4.2 在服务期限内，百度智能云将为您提供7x24小时售后故障服务。

4.3 若服务涉及云数据库 Redis，百度智能云仅负责数据库系统以下的底层部分及百度智能云提供的软件的运营维护，即 Redis 的相关技术架构及操作系统等。数据库之上的应用部分由您自行负责。

4.4 百度智能云保留在您资源欠费超过一定天数（具体规则参见产品文档）时将资源彻底销毁的权利，一旦彻底销毁资源将不可恢复。您的业务数据一经删除，即不可恢复；您应自行承担数据因此被删除所引发的后果和责任，您理解并同意，百度智能云没有继续保留、导出或者返还您业务数据的义务。

4.5 您须依照《互联网信息服务管理办法》、《互联网电子公告服务管理规定》的规定保留自己网站的访问日志记录，包括发布的信息内容及其发布时间、互联网地址（IP）、域名等，该记录在国家有关机关依法查询时必须提供。您自行承担由于未按规定保留相关记录而引起的全部责任。

4.6 本服务条款中云数据库所处的服务器的所有权属于百度智能云，百度智能云为您提供网站控制台并进行网站的日常维护。

4.7 百度智能云可以帮助您进行日常的数据备份，支持自动备份和手动备份。但百度智能云不保证可完全备份您数据库上的数据，您应自行备份其数据库上的相关数据并检查其完整性。您充分认识到对您数据的备份并非百度智能云的义务，百度智能云亦不对您网站的数据备份工作承担任何责任。

4.8 百度智能云将不定期举办培训和技术咨询，帮助您充分地了解云数据库服务。

4.9 百度智能云将消除您非人为操作所出现的故障，但因您原因和/或不可抗力以及非百度智能云控制范围之内的事项除外。

4.10 您理解并认可，百度智能云将在必要时对您云数据库产品进行免费升级，或将主机进行机房迁移，期间可能会发生短暂中断。百度智能云将在上述操作前至少24小时通知您，由于上述操作所带来的服务中断等影响，百度可按照赔偿标准对所影响的时长进行赔偿。如您不同意百度智能云对数据库进行升级和/或迁移机房的，可在收到百度智能云通知后主动申请解除本服务条款，届时服务期限和费用将按实际服务月份计算（不足一个月的按一个月计），百度智能云将按照百度智能云网站公示的退款流程向您退还对应产品已支付但未发生消费使用的服务费。

4.11 本服务中可能会使用第三方软件或技术，若有使用，百度智能云保证已经获得合法授权，同时，百度智能云将按照相关法规或约定，对相关的协议或其他文件进行展示。前述通过各种形式展现的相关协议或其他文件，均是本服务条款不可分割的组成部分，与本服务条款具有同等的法律效力，您应当遵守这些要求，否则应承担相应的责任。

4.12 您应保证提交百度智能云的素材、对百度智能云服务的使用及使用百度智能云服务所产生的成果未侵犯任何第三方的合法权益。如有第三方基于侵犯版权、侵犯第三人权益或违反中国法律法规或其他适用的法律等原因而向百度智能云提起索赔、诉讼或可能向其提起诉讼，则您应赔偿百度智能云因此承担的费用或损失，并使百度智能云完全免责。

4.13 如果第三方机构或个人对您使用百度智能云服务所涉及的相关素材的知识产权归属提出质疑或投诉，您有责任出具相关知识产权证明材料，并配合百度智能云相关投诉处理工作。

4.14 若您使用百度智能云产品或服务从事您的权利与义务中禁止的事项，或利用百度智能云从事违法或有损他人的事项，百度智能云有权对您进行审查，停用账户，并删除相应违规资源。

## 5. 服务费用

5.1 您理解并认可：您须按照百度智能云现时有效的云数据库（及其配套产品）价格体系支付其所购买产品有效期内的所有费用。其产品价格依据当前时刻线上产品报价或双方线下约定履行，并具备时效性，服务期内如发生因百度智能云促销、产品成本提高或降低或其他因素导致的产品提价或降价，均不对合同所签署金额造成影响，合同期内百度智能云和您均没有补偿或追索合同差价的权利。

5.2 服务期满双方愿意继续合作的，您应在服务期满前一个月内支付续费款项，以使服务得以继续进行。如续费时百度智能云对产品体系、名称或价格进行调整的，双方同意按照届时有效的新的产品体系、名称或价格履行。

5.3 百度智能云保留在您未按照约定支付全部费用之前不向您提供服务和/或技术支持，或者终止服务和/或技术支持的权利。

5.4 您完全理解百度智能云价格体系中所有的赠送服务项目均为百度智能云在正常服务价格之外的一次性特别优惠，优惠内容不包括赠送服务项目的修改、更新及维护费用，并且赠送服务项目不可折价冲抵服务价格。

## 6. 服务期限及服务终止条件

6.1 服务有效期：自百度智能云收到您款项、且百度智能云成功创建您所需配置的产品时刻起计算，服务期限以您所选期限及

其支付的服务款项数额为依据。鉴于您申请云数据库后，百度智能云将为您预留空间以备随时开通，您承认其办理网站经营许可证/网站备案的相关期间将包含在您购买的服务期之内。

6.2 如果您没有按时支付续约款项，则在本服务有效期结束后，服务即告终止。百度智能云届时将关闭您的云数据库服务。关闭服务之日起7日内，若您依然没有续费，百度智能云将有权删除云数据库内的所有文件，由此带来的后果由您自行负责。

6.3 如您发生下列情形，服务期限提前终止：（1）双方协商一致提前终止的。（2）您违反本服务条款，百度智能云有权提前终止服务，并不退还您已经支付的费用。（3）您可以单方提出提前终止服务的要求，但应提前30天书面通知百度智能云，此时您已交纳的费用不再返还。您擅自终止本服务条款给百度智能云造成损失的，还应予以承担。

6.4 您理解：鉴于计算机、互联网的特殊性，下述情况不属于百度智能云违约：（1）百度智能云在进行服务器配置、维护时，需要短时间中断服务。（2）由于电信运营商设备故障或Internet上的通路阻塞造成您网站访问速度下降或不能进行数据传输的。（3）因台风、地震、海啸、洪水、停电、战争、恐怖袭击等不可抗力之因素，造成本公司系统障碍不能执行业务的。（4）因黑客问题、电信部门技术调整和政府管制等引起的事件。

6.5 若因百度智能云原因造成云数据库不能正常使用的，百度智能云按照实际故障时长进行赔偿，详情请参考《云数据库Redis 服务等级协议SLA》中的赔偿方案相关内容。

6.6 若您在云数据库上的应用给服务器带来异常大的负荷，以致影响该云数据库所在服务器的正常运行或者影响其他用户的正常使用（包括但不限于大量占用百度智能云服务器内存、CPU资源的行为），百度智能云有权停止您的数据库服务，终止服务且余款不予退还。

6.7 百度智能云对因第三方的过错或者延误而给您造成的损失不承担责任，同时，百度智能云对通过您间接接受百度智能云服务的第三方的损失不承担责任。

## 7. 披露与公开

7.1 百度智能云不会在未经您许可情况下披露、公开您存储于云数据库的代码及其他资料，除非：

- （1）有关法律、法规规定或云数据库合法服务程序规定。
- （2）在紧急情况下，为维护您及其他公众的权益。
- （3）为维护百度的商标权、专利权及其他任何合法权益。
- （4）其他依法需披露、公开代码及其他信息的情况。

## 8. 保密与信息披露

8.1 保密信息，是指不为公众所知悉、能为权利人带来经济利益、具有实用性并经权利人采取保密措施的技术信息和经营信息。

8.2 未经对方书面许可，任何一方不得向第三方（有关法律、法规或政府部门要求除外）泄露本服务条款条款的任何内容、本服务条款签订履行情况以及通过签订和履行本服务条款而获知的对方的保密信息。

8.3 双方在协议履行中发生分歧或需要有相关的重大信息披露时，应首先积极争取双方沟通协商一致，相关重大信息披露、采用、表述应获得双方的许可，双方各个领域自觉维护对方的品牌价值和利益。

8.4 保密条款不因本服务条款无效、解除、提前终止或不具操作性而失效。本服务条款期满后，各方保密义务并不随之终止，保密义务直到另一方同意其解除此项义务，或事实上不会因违反保密义务而给另一方造成任何形式损害时为止。

## 9. 免责声明

9.1 本公司保留您违反国家、地方法律法规规定或违反本服务条款的情况下中止或终止为您提供服务的权利。

9.2 鉴于网络服务的特殊性，您同意百度智能云在有正当理由的情况下可以随时变更、中断或终止部分或全部的服务，也无需向您承担任何责任。但是，百度智能云会尽可能事前通知您，以便您做好相关数据的转移备份以及业务调整等，以保护您的合法权益。

9.3 为了向您提供更完善的服务，百度智能云有权定期或不定期地对提供本服务的平台或相关设备进行检修、维护、升级等，如因此类情况造成相关服务在合理时间内中断或暂停的，百度智能云无需为此向您承担任何责任。

9.4 您理解并同意：百度智能云的服务是按照现有技术和条件所能达到的现状提供的。百度智能云会尽最大努力向您提供服务，确保服务的连贯性和安全性；但百度智能云不能保证其所提供的服务毫无瑕疵，也无法随时预见和防范法律、技术以及其他风险，包括但不限于不可抗力、病毒、木马、黑客攻击、系统不稳定、第三方服务瑕疵、政府行为等原因可能导致的服务中断、数据丢失、数据泄露或被窃取以及其他的损失和风险。所以您同意：即使百度智能云提供的服务存在瑕疵，但上述瑕疵是当时行业技术水平所无法避免的，其将不被视为百度智能云违约，同时，由此给您造成的数据或信息丢失等损失百度智能云将予以免责。

9.5 您理解并同意：在使用本服务的过程中，可能会遇到不可抗力（包括但不限于地震、火灾、战乱、政治影响、网络攻击等）等风险因素，使本服务发生中断。出现上述情况时，百度智能云将努力在第一时间与相关单位配合，及时进行修复，但是由此给您造成的损失百度智能云将予以免责。

## 10. 其他

10.1 本服务条款的订立、执行和解释及争议的解决均应适用中国法律并受中国法院管辖。如双方就本服务条款内容或其执行发生任何争议，双方应尽量友好协商解决；协商不成时，任何一方均可向北京市海淀区人民法院提起诉讼。

10.2 本服务条款构成双方对本服务条款之约定事项及其他有关事宜的完整协议，除本服务条款规定的之外，未赋予本服务条款各方其他权利。

10.3 如本服务条款中的任何协议无论因何种原因完全或部分无效或不具有执行力，本服务条款的其余协议仍应有效并且有约束力。

# 百度智能云可信云

## 1. 服务范围

云数据库 Redis（Simple Cache Service for Redis）提供稳定、高效以及高可扩展性的分布式缓存服务。云数据库 Redis 兼容 Redis/Memcached 协议，基于 Redis 提供标准版和集群版的架构模式，并支持自定义副本数量，为您提供多样化的数据结构支持。云数据库双副本 Redis 实例默认支持主从热备，实现故障自动切换、故障迁移、数据备份、实例监控等运维管理功能。

## 2. 服务可用性

2.1 服务可用性计算公式 服务可用性以单个实例为维度，按照如下方式计算：

$$\text{服务可用性} = \left( \frac{\text{服务周期总分钟数} - \text{服务不可用分钟数}}{\text{服务周期总分钟数}} \right) \times 100\%$$

### 2.2 服务可用性承诺

Redis承诺服务可用性不低于 99.99%，即用户每月业务可用时间应为 $30 \times 24 \times 60 \times 99.99\% = 43195.68$ （分钟），即存在 $43200 - 43195.68 = 4.32$ （分钟）的不可用时间。如Redis未达到上述服务可用性承诺，用户可以根据本协议第三条约定获得赔偿。赔偿范围不包括以下原因所导致的服务不可用时间：

- (1) 百度智能云预先通知用户后进行系统维护所引起的，包括割接、维修、升级和模拟故障演练。
- (2) 任何百度智能云所属设备以外的网络、设备故障或配置调整引起的。
- (3) 用户的应用程序受到黑客攻击而引起的。
- (4) 用户维护不当或保密不当致使数据、口令、密码等丢失或泄漏所引起的。
- (5) 用户不开启监控告警致使系统故障无法及时暴露引起的。
- (6) 用户执行清空数据等操作造成数据丢失引起的。
- (7) 用户的疏忽或由用户授权的操作所引起的。
- (8) 用户未遵循百度智能云产品使用文档或使用建议引起的。
- (9) 不可抗力或者其他意外事件引起的。
- (10) 其他非百度智能云原因所造成的不可用。
- (11) 购买的实例为单副本形态。

## 3. 数据知情性

3.1. 用户对于数据、备份数据所在数据中心地理位置、数据备份数量具有知情权，其中：

3.1.1 目前百度智能云数据中心分别位于华北、华东和华南等地域，用户在开通云服务时必须根据地理位置选择相应的数据中心，用户数据将存储在其指定的数据中心；

3.1.2 百度智能云服务具备自动数据备份功能，备份数量详见相关技术文档，备份数据默认与源数据存储在同一个数据中心。用户无需指定数据自动备份数量及自动备份数据所存储的位置。

3.1.3 百度智能云的数据中心将遵守当地的相关法律法规，用户对此具有知情权，并可联系百度智能云的客户服务人员获得详

尽信息。

3.1.4 除应当地法律法规、或政府监管部门的监管、审计要求，用户的所有数据、应用及行为日志不会提供给第三方。除用于百度智能云的产品运行状态的统计分析，用户的行为日志不会对外呈现用户个人信息数据。

#### 4. 服务可审查性

依据现行法律法规或根据政府监管部门监管、安全合规、审计或取证调查等原因的需要，在符合流程和手续/法律文书完备的情况下，百度智能云将提供用户所使用服务的相关信息，包括关键组件的运行日志、运维人员的操作记录、用户操作记录等信息。

#### 5. 服务功能

云数据库Redis是一种专业、高性能、高可靠的云数据库服务。云数据库Redis实例可以通过web或API的方式开通，并提供在线扩容、监控告警等管理功能。云数据库Redis服务的所有具体功能请详见百度智能云在官网上提供的详细说明文档、技术文档及帮助文档。云数据库Redis所有可能影响用户的功能性变更都将向用户公告。

#### 6. 服务资源调配能力

6.1 云数据库Redis提供多种规格和配置，用户可根据业务需要选择适合的规格和配置，并且可在多种规格和配置之间进行业务退订或升级。

6.2 云数据库Redis具备弹性扩容能力，用户可根据需要自行在线扩展或缩减所使用的Redis资源。资源调配的最大最小值不设限制，按用户需求为准，系统自动调配，调整速度为重启后生效。

#### 7. 故障恢复能力

7.1 云数据库Redis为付费用户的云服务提供7×24小时的运行维护，并以在线工单和热线支持等方式提供技术支持。

7.2 百度智能云具备完善的故障监控、自动告警、快速定位、快速恢复等一系列故障应急响应机制。

#### 8. 数据访问性能指标

云数据库Redis仅限于百度智能云内网环境访问使用，内网带宽限制为500Mbps，具体性能指标详见百度智能云在官网上提供的详细说明文档、技术文档及帮助文档。云数据库Redis也支持公网方式访问。百度智能云承诺规格性能达标，详情可见链接：<https://cloud.baidu.com/doc/SCS/s/vjxvpqv8d>。

#### 9. 服务计量准确性

9.1 百度智能云Redis服务具备准确、透明的计量计费系统，百度智能云根据用户订购以及实际使用量据实结算，实时扣费。

9.2 百度智能云Redis的具体计费标准以百度智能云官网公布的有效计费模式与价格为准。

9.3 用户的原始计费日志默认最少保留3年备查。

#### 10. 服务可用性的赔偿条款

10.1 服务可用性的赔偿 具体详见《云数据库Redis服务等级协议SLA》。

#### 11. 服务变更、终止条款

11.1 服务有效期：自百度智能云收到您款项、且百度智能云成功创建您所需配置的产品时刻起计算，服务期限以您所选期限及其支付的服务款项数额为依据。鉴于您申请云数据库后，百度智能云将为您预留空间以备随时开通，您承认其办理网站经营许可证/网站备案的相关期间将包含在您购买的服务期之内。

11.2 如果您没有按时支付续约款项，则在本服务有效期结束后，服务即告终止。百度智能云届时将关闭您的云数据库服务。关闭服务之日起7日内，若您依然没有续费，百度智能云将有权删除云数据库内的所有文件，由此带来的后果由您自行负责。

11.3 如您发生下列情形，服务期限提前终止：（1）双方协商一致提前终止的。（2）您违反本服务条款，百度智能云有权提前终止服务，并不退还您已经支付的费用。（3）您可以单方提出提前终止服务的要求，但应提前30天书面通知百度智能云，此时您已交纳的费用不再返还。您擅自终止本服务条款给百度智能云造成损失的，还应予以承担。

11.4 您理解：鉴于计算机、互联网的特殊性，下述情况不属于百度智能云违约：（1）百度智能云在进行服务器配置、维护时，需要短时间中断服务。（2）由于电信运营商设备故障或Internet上的通路阻塞造成您网站访问速度下降或不能进行数据传输的。（3）因台风、地震、海啸、洪水、停电、战争、恐怖袭击等不可抗力之因素，造成本公司系统障碍不能执行业务的。

（4）因黑客问题、电信部门技术调整和政府管制等引起的事件。

11.5 若因百度智能云原因造成云数据库不能正常使用的，百度智能云按照实际故障时长进行赔偿，详情请参考《云数据库Redis服务等级协议SLA》中的赔偿方案相关内容。

11.6 若您在云数据库上的应用给服务器带来异常大的负荷，以致影响该云数据库所在服务器的正常运行或者影响其他用户的正



常使用（包括但不限于大量占用百度智能云服务器内存、CPU资源的行为），百度智能云有权停止您的数据库服务，终止服务且余款不予退还。

11.7 百度智能云对因第三方的过错或者延误而给您造成的损失不承担责任，同时，百度智能云对通过您间接接受百度智能云服务的第三方的损失不承担责任。

## 12. 用户约束条款

您同意遵守本服务条款以及服务展示页面的相关管理规范及流程。 12.1 您保证不会利用技术或其他手段破坏或扰乱本网站及百度智能云上其他客户。

12.2 您应按照百度智能云页面提示及本服务条款的约定支付相应的服务费用。

12.3 您保证使用本服务时将遵从国家、地方法律法规、行业惯例和社会公德，不会利用本公司提供的服务进行存储、发布、传播如下信息和内容：（1）违反国家法律法规政策的任何内容（信息）；（2）违反国家规定的政治宣传和/或新闻信息；涉及国家秘密和/或安全的信息；（3）封建迷信和/或淫秽、色情、下流的信息或教唆犯罪的信息；（4）博彩有奖、赌博游戏；（5）违反国家民族和宗教政策的信息；（6）妨碍互联网运行安全的信息；（7）侵害他人合法权益的信息和/或其他有损于社会秩序、社会治安、公德的信息或内容。

12.4 您承诺不得为他人发布上述不符合国家规定和/或本服务条款约定的信息内容提供任何便利，包括但不限于设置URL、BANNER链接，发送邮件等。您承认本公司有权在您违反上述约定时有权终止向您提供服务并不予退还任何款项，因您上述行为给本公司造成损失的，您应予赔偿。

12.5 当您在违反本服务条款的情况下，本公司有权拒绝继续为您提供任何服务。

12.6 您应自行对因使用本服务而存储在百度智能云数据库的各类数据等信息，在本服务之外，采取合理、安全的技术措施，确保其安全性和完整性，并对自己的行为（包括但不限于欠费，自行安装软件、采取加密措施或进行其他安全措施等）所引起的结果承担全部责任。

12.7 您在使用本产品时须遵守本服务条款以及百度智能云的相关要求，不得利用本服务从事包括但不限于以下行为，也不得为以下行为提供便利（包括但不限于以下内容）：

- （1）进行博彩、赌博游戏、“私服”、“外挂”等非法互联网活动；
- （2）发布、传送、传播电子邮件广告、垃圾邮件（SPAM）；
- （3）发布、传送、传播不受欢迎的或者未经请求的电子邮件、电子广告或包含反动、色情等有害信息的电子邮件；
- （4）将本产品用作磁盘空间出租；
- （5）进行任何破坏或试图破坏网络安全的行为，包括但不限于以病毒、木马、恶意代码、钓鱼等方式，采取对其他网站、服务器进行恶意扫描、非法侵入系统、非法获取数据等；
- （6）进行任何改变或试图改变本服务提供的系统配置或破坏系统安全的行为；
- （7）不可大量占用，且不得导致程序或进程大量占用百度智能云的云计算资源（如云服务器、带宽、存储空间）所组成的平台中服务器的内存、CPU或带宽资源；并给百度智能云其他用户的服务器、网络、产品/应用带来严重的、不合理的负荷，影响百度智能云与互联网、特定网络、内部正常的联系；或导致百度智能云的产品与服务或者其他用户的服务宕机、关机或者用户的产品/应用不可访问等；
- （8）利用技术或其他手段破坏、扰乱本服务的运营或他人对本服务的使用；
- （9）以任何方式干扰或企图干扰百度任何产品或任何部分、功能的正常运行，或者制作、发布、传播上述工具、方法等；
- （10）进行任何可能违反诚实信用、公序良俗、公众道德的行为；
- （11）若您因从事任何业务，导致您频繁遭受攻击（包括但不限于DDoS攻击）且未及时更正您的行为或未根据百度要求消除影响，从而对本服务平台或其他本服务使用者产生威胁的，一经发现，本产品将被强制停止，账号也会立即冻结。
- （12）任何违反中华人民共和国现行有效法律/法规/规章等法律禁止性行为，包括但不限于：反对宪法所确定的基本原则的。危害国家安全，泄露国家秘密，颠覆国家政权，破坏国家统一的。损害国家荣誉和利益的。煽动民族仇恨、民族歧视，破坏民族团结的。破坏国家宗教政策，宣扬邪教和封建迷信的。散布谣言，扰乱社会秩序，破坏社会稳定的。散布淫秽、色情、赌博、暴力、凶杀、恐怖或者教唆犯罪的。侮辱或者诽谤他人，侵害他人合法权益的。

## 13. 服务商免责条款

13.1 本公司保留您违反国家、地方法律法规规定或违反本服务条款的情况下中止或终止为您提供服务的权利。

13.2 鉴于网络服务的特殊性，您同意百度智能云在有正当理由的情况下可以随时变更、中断或终止部分或全部的服务，也无需

向您承担任何责任。但是，百度智能云会尽可能事前通知您，以便您做好相关数据的转移备份以及业务调整等，以保护您的合法权益。

13.3 为了向您提供更完善的服务，百度智能云有权定期或不定期地对提供本服务的平台或相关设备进行检修、维护、升级等，如因此类情况造成相关服务在合理时间内中断或暂停的，百度智能云无需为此向您承担任何责任。

13.4 您理解并同意：百度智能云的服务是按照现有技术和条件所能达到的现状提供的。百度智能云会尽最大努力向您提供服务，确保服务的连贯性和安全性；但百度智能云不能保证其所提供的服务毫无瑕疵，也无法随时预见和防范法律、技术以及其他风险，包括但不限于不可抗力、病毒、木马、黑客攻击、系统不稳定、第三方服务瑕疵、政府行为等原因可能导致的服务中断、数据丢失、数据泄露或被窃取以及其他的损失和风险。所以您同意：即使百度智能云提供的服务存在瑕疵，但上述瑕疵是当时行业技术水平所无法避免的，其将不被视为百度智能云违约，同时，由此给您造成的数据或信息丢失等损失百度智能云将予以免责。

13.5 您理解并同意：在使用本服务的过程中，可能会遇到不可抗力（包括但不限于地震、火灾、战乱、政治影响、网络攻击等）等风险因素，使本服务发生中断。出现上述情况时，百度智能云将努力在第一时间与相关单位配合，及时进行修复，但是由此给您造成的损失百度智能云将予以免责。

#### 14.其他

您理解并确认，可信云以年度为单位进行认证，因此百度智能云将按年度根据认证机关的要求和百度智能云产品的变化，进行此可信云文档的更新和调整。

## 典型实践

### 客户端配置注意事项

云数据库 Redis 集群版和单机版兼容常用语言（Java/Golang/Python/Javascript 等）的主流客户端，且支持单机、redis cluster、sentinel 等连接模式。各个 SDK 均提供了丰富的配置项，本文将对部分比较重要的配置项做一些描述，指出部分注意事项。

#### 客户端超时时间

客户端配置的超时时间不宜过短，否则后端 server 可能尚未处理完成就发生了超时。另外也不建议不配置超时，或者超时时间太长，比如请求长时间未完成，或连接长时间未建立，此时客户端有理由允许请求快速失败。某些命令执行成本比其他命令要高，比如，一次性读写多个 key 的 MGET、MSET 命令，Lua 脚本，事务、响应比较大的命令，对于这类命令，建议使用更大的客户端超时，以避免命令尚未处理完成客户端就超时了，且重试后再次出现相同的情况。这类命令包含：

- 包含较多 key 的 MSET、MGET、DEL 等：这类命令会在 Proxy 层做拆分路由到多个分片，最终响应时间由最后一个分段收到响应的时间来确定，因此这类命令往往具有相对较高的延迟。
- 包含较多命令的 MULTI/EXEC 事务或 Lua 脚本：这类命令需要 server 处理较长的时间，一条 eval 命令脚本耗时可能是 GET/SET 的数千倍。但通常只要控制好复杂度和并发，则影响比较可控。
- 读取较大的值：如 hgetall、keys 这类命令，返回的内容较多，这需要更长的网络传输时间，通常具有较高的延迟。
- 执行诸如 BLPOP 之类的阻塞操作：这类阻塞命令，在无数据就绪时，本就需要等待较长时间。

目前 Proxy 和后端 server 间，针对非阻塞的请求超时默认为 2 秒，因此客户端配置的超时时间应该小于 2 秒。使用内存型 Redis 通常响应能在 10 毫秒内完成，如果遇到流量激增负载较重时，可能达到几十毫秒。因此，建议根据业务中对延迟的容忍程度，将超时时间设置为 50ms~2s 之间的数字。

#### 连接池

一些客户端使用独占连接来处理请求，即在一个请求未执行完成之前，不能处理其他请求，当应用程序需要同时发送多个 Redis 命令时，就需要使用连接池。使用连接池可以避免频繁地创建和释放连接，可有效提升性能。在配置使用连接池时，需要考虑连接池的容量，链接探活策略，连接获取顺序等。

#### 容量：

连接池容量不宜设置过大，否则会导致 server 端存在大量连接，同时也不可设置过小，否则会导致 Proxy 的流量不均衡，建议和发送命令的线程数保持在一个数量级。

#### 连接探活策略：



Proxy 会主动关闭空闲时间超过 3600s 的连接，因此建议开启连接探活，探活周期可以适当调大，小于 3600s 即可。

#### 🔗 连接获取顺序：

连接获取策略建议使用 FIFO (First In First Out)，使用 LIFO (Last In First Out) 可能导致只有少量连接被使用到，引起Proxy流量不均。客户端 SDK 一般默认 LIFO (Last In First Out) 的策略获取连接池中的连接，即每次使用最近放入连接池的连接。考虑一个极端的例子，有 50 个客户端，每个客户端中只有一个连接活跃，那么连接到云数据库 Redis 集群的活跃连接数则为 50。如果集群有 20 个 Proxy，50 个连接到 20 个 Proxy，则极有可能出现 Proxy 上负载不均衡的问题。这可能导致部分 Proxy CPU 被打满，而其他的则很空闲。因此，我们建议使用 FIFO (First In First Out) 的策略，即轮流地使用所有的连接。

#### 🔗 Jedis 连接池配置

下面给出使用 Jedis 时，配置使用连接池的例子：

```
String host = "redis.*****.baidubce.com";
int port = 6379;
String password = "*****";
JedisPoolConfig config = new JedisPoolConfig();
config.setMaxTotal(30);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
config.setLifo(false);
try (JedisPool pool = new JedisPool(config, host, port, 3000, password)) {
 Jedis jedis = pool.getResource();
 jedis.set("client", "jedis");
 assert jedis.get("client").equals("jedis");
}
catch (Exception e) {
 e.printStackTrace();
}
```Go
```

你可以在 `JedisPoolConfig` 对象上配置连接池的细节：

```
config.setMaxTotal(10); // 设置最大连接数
config.setTestOnBorrow(false); // 从连接池中拿连接是测试其可用性
config.setTestOnReturn(false); // 放回连接池中是测试其可用性
config.setTestWhileIdle(false); // 连接空闲时测试其可用性
config.setLifo(false); // 设置连接池是否为后进先出
```

参数	说明	默认值	建议
maxTotal	连接池中最大连接数	8	建议和同时发送命令的线程数保持在一个数量级。
maxIdle	连接池允许的最大空闲连接数	8	这表示连接池中最大允许的空闲连接数，当此值设置的比较大时，突发的大量请求会导致创建较多的连接，而后这些连接一直处于空闲状态。当如果设置的过小，连接可能会频繁地被创建和释放。建议根据流量波动来设置，如果波动很小，可以设置为一个较小值，比如保持默认值。如果波动比较剧烈，为了避免频繁地创建和释放连接，可以和 maxTotal 保持一致。
minIdle	连接池中最少空闲连接数	0	如果设置 minIdle 为非零的值，连接池中始终会有一些空闲连接。
blockWhenExhausted	当连接池中连接用尽后，获取连接时是否要等待。当值为 true 时，下面的 maxWait 才会生效。	true	建议使用默认值。
maxWait	从连接池获取连接时的最大等待时间。	-1（一直等待）	此值不宜设置得过大。在大多数使用 Redis 的场景下，网络延迟都在百毫秒以下，将此值设置为 2-5 秒比较合适。如果对延迟有更高的要求，可以设置得更小一些。
testOnBorrow	从连接池取连接时是否做连接有效性检测（发送 PING 命令）。检测到的无效连接将会被移除。	false	用于连接保活，连接池可以快速发现并剔除错误连接，如果高频地从连接池获取连接，开启后集群会增加不少探测流量。通常连接不会无故断开，另外用户在执行命令时也会做异常的检测，所以这里建议设置为 false。
testOnReturn	向资源池归还连接时是否做连接有效性检测（ping）。检测到无效连接将会被移除。	false	同 testOnBorrow，建议设置为 false。
lifo	从连接池中取连接的策略，是否采用 Last in first out 的策略	true	用于配置连接池获取连接的策略，和前文描述的 golang 客户端中 PoolFIFO含义类似，这里建议设置为 false，以保证 Proxy 上流量更均衡。

🔗 失败重试

用户使用云数据库 Redis 集群时，网络抖动、机器偶发故障等因素均可能导致请求失败，百度智能云数据库 Redis 集群有自动故障恢复的策略，设计合理的客户端自动重试机制能够显著提升应用程序的健壮性和用户体验。本节内容将描述配置客户端重试时需要注意的事项。

🔗 Jedis 客户端重试示例

Jedis 4.0.0 及以上版本支持重试机制。以下示例展示了如何使用 Jedis 5.0.0 进行重试操作。

```
PooledConnectionProvider provider = new PooledConnectionProvider(HostAndPort.from("127.0.0.1:6379"));
int maxAttempts = 5; // 最大重试次数
Duration maxTotalRetriesDuration = Duration.ofSeconds(10); // 最大的重试时间
UnifiedJedis jedis = new UnifiedJedis(provider, maxAttempts, maxTotalRetriesDuration);
try {
    jedis.set("key", "value");
} catch (Exception e) {
    // 多次尝试仍没成功
    e.printStackTrace();
}
```

🔗 GoLang 客户端重试示例

GoLang 中配置重试可关注以下三个配置项：

配置项	默认值	说明
MaxRetries	3	默认失败后重试三次
MinRetryBackoff	8	如果多次失败，重试间隔时间会逐渐变长，此为最短间隔
MaxRetryBackoff	512	此为最长间隔

下面是在 Golang 中配置重试的示例：

```
client := redis.NewClient(&redis.Options{
    Addr:      "127.0.0.1:6379",
    Password:  "password",
    MaxRetries: 3,
    MinRetryBackoff: 8 * time.Millisecond,
    MaxRetryBackoff: 512 * time.Millisecond,
})
```

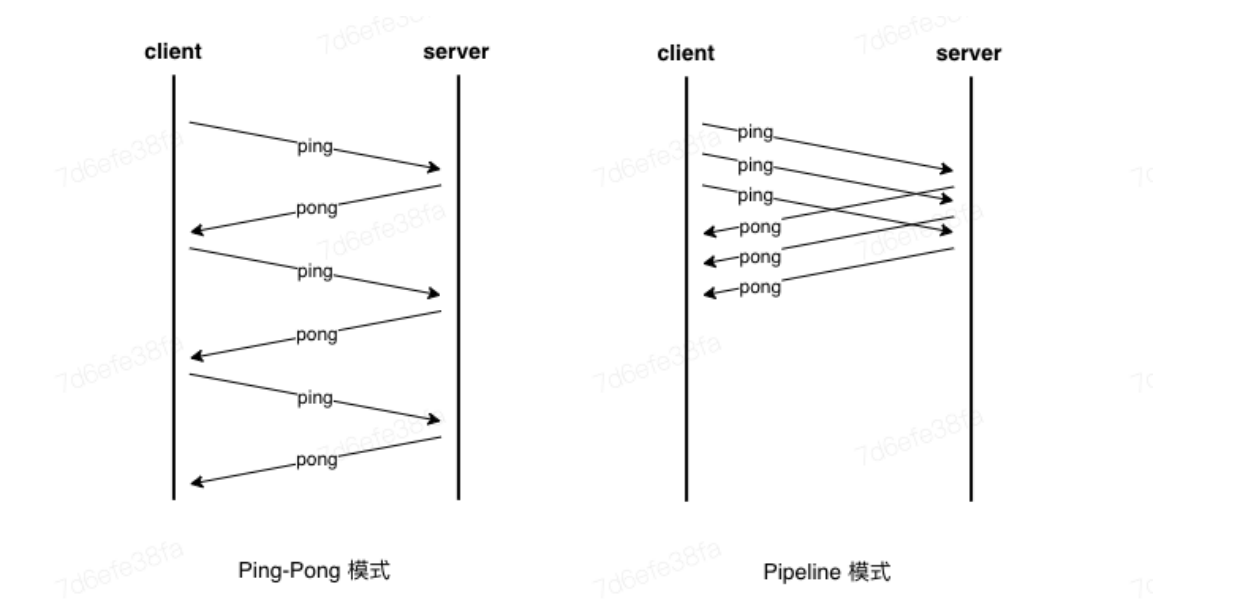
Redis Pipeline 机制介绍

概述

Redis Pipeline 是一种将多个命令打包并一次性发送到 Redis 服务器的机制，它可以有效提高执行命令的效率，可大幅提升吞吐量，从而增强性能。

工作原理

在传统的客户端与 Redis 通信模式（Ping-pong 模式）中，客户端发送一个命令后需等待其执行结果，收到结果后才发送下一个命令。这种模式适合交互式操作，但在处理大量命令时，网络往返时延（Round-trip time, RTT）可能成为性能瓶颈。相比之下，Pipeline 模式允许客户端将多个命令一次性发送给服务器，无需等待每个命令的响应。客户端在发送完所有命令后，再开始接收响应。下图为Ping-Pong模式与Pipeline模式的示意图：



客户端无需等待前面命令执行完成就发送后续命令，命令可以源源不断地到达 server 被处理，这就消除了网络延迟的影响。在需要批量写入和批量查询时，使用 pipeline 可以有效提升单位时间内处理的命令数量。另外，一次性发送多条命令，可大幅减少执行 write 的次数，同时 server 一个 read 操作就可以从 socket 中读取出多条命令，而执行命令的代价常常低于从 socket 中读取命令，因此合理的使用 pipeline 可有效降低处理同等数据量的命令的耗时CPU资源消耗。

使用 pipeline 的注意事项

pipeline 不保证原子性

一个 Pipeline 中的命令不保证被原子性地执行，因为 pipeline 模式仅仅是一次性发送多条命令，在 server 端，这些命令不能保证被一次性从 socket 中读取出来，也不会保证被一次性处理。一个 pipeline 中的命令在处理时，可能会穿插执行其他客户端的命令。

🔗 pipeline 不支持回滚

如果在 Pipeline 执行过程中发生错误，不支持回滚。如果命令之间存在依赖关系，应避免使用 Pipeline。

🔗 避免使用过长的 pipeline

当需要大批量地写入数据时，有的用户常常采用很长的 pipeline 来写入数据，而这很可能不够高效。云数据库 Redis 集群中包含多个 Proxy 节点，客户端发送 pipeline 时会选择一条连接来发送。如果 pipeline 很长，则这些命令均会由某一个 proxy 来处理，而其他的 proxy 可能会处于空闲状态。在需要大批量地写入数据时，推荐使用较短的 pipeline 并配合连接池来使用，这样可以将长度适中的 pipeline 发送给不同的 Proxy 来处理，这样不至于让某个 Proxy 成为性能瓶颈。

🔗 注意 Python 客户端的实现差异

Python 客户端支持使用事务命令 MULTI、EXEC 来模拟 Pipeline，这种方式下，一批命令会被放在在 multi 和 exec 中，这种方式可以保证 pipeline 中命令被原子性地执行。但如果仅仅是为了提升吞吐量，不推荐使用这种方式。

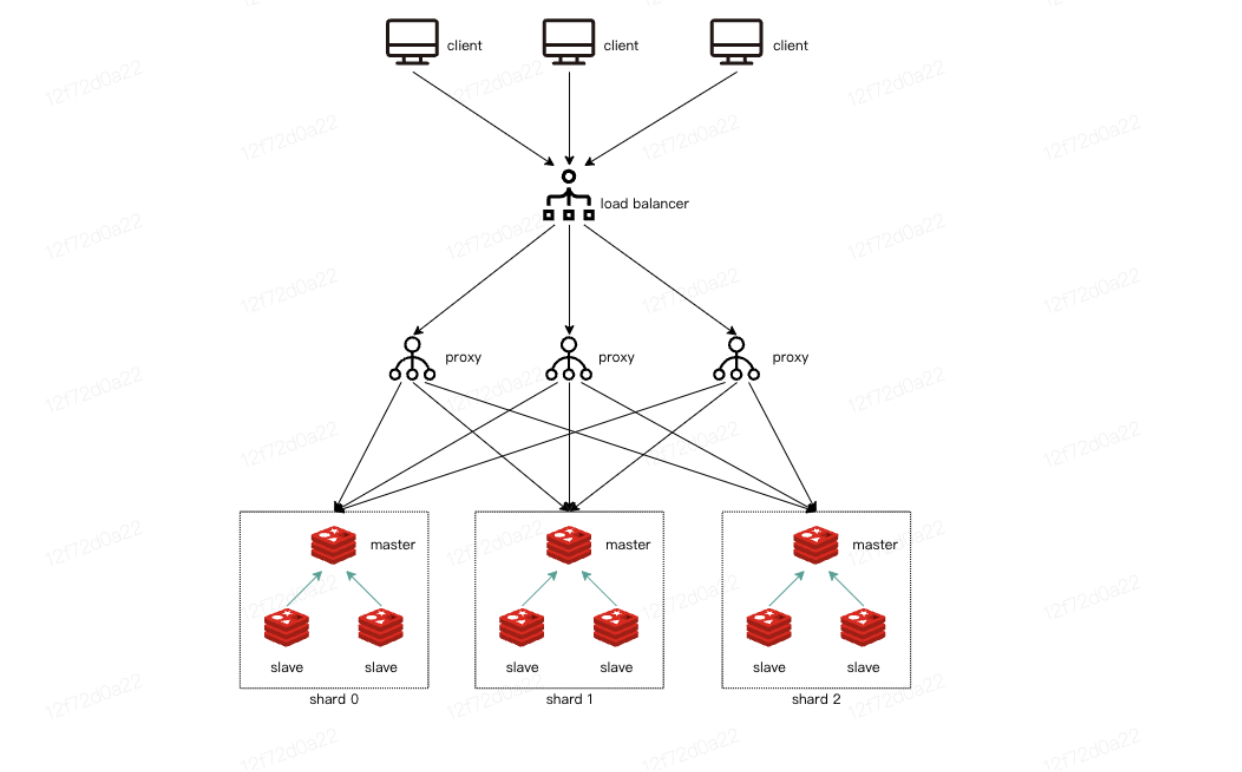
```
conn = redis.Redis("127.0.0.1", 6379)
pipe = conn.pipeline(transaction=True)
for i in range(1024):
    pipe.set(f"key_{i}", "value_{i}")
pipe.execute()
```

消息发布与订阅

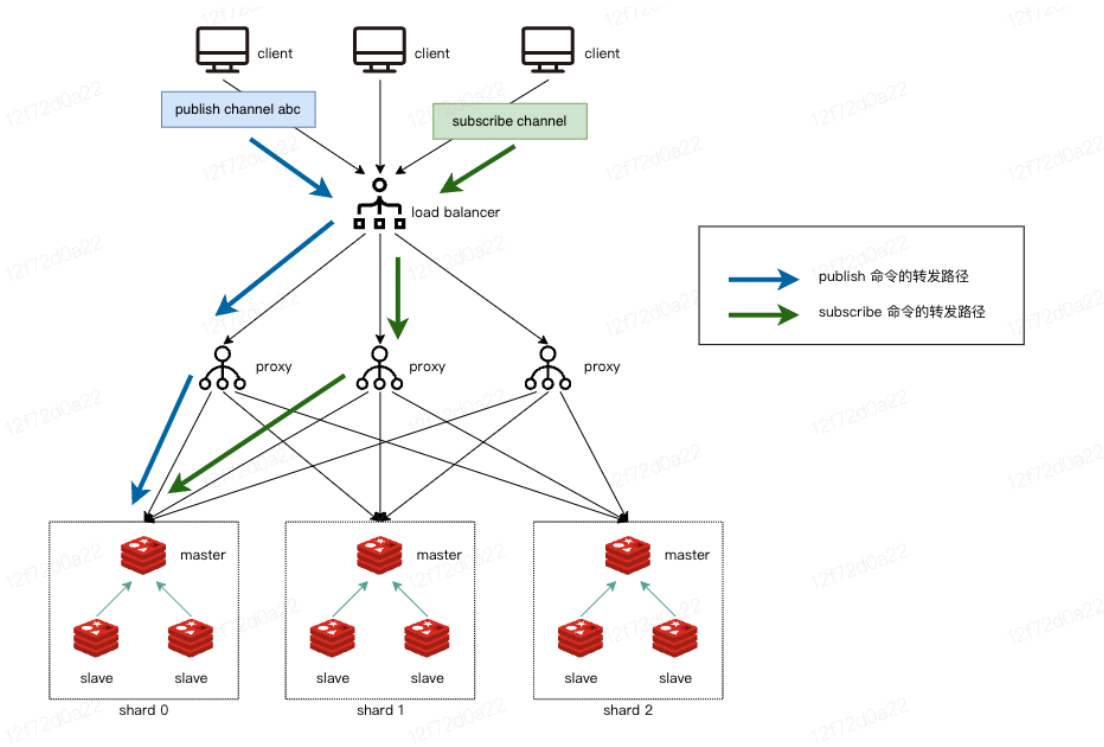
Redis 的发布订阅（Pub/Sub）是一种消息通信模式，其中消息发布者（publisher）发送消息到指定的频道（channel），而消息订阅者（subscriber）订阅这些频道以接收消息。百度智能云Redis 集群版和标准版均支持发布订阅，本文将描述使用发布订阅时的一些注意事项。

🔗 背景

云数据库Redis 使用基于代理的集群方案，其结构如下：



在集群版中，当用户使用发布与订阅时，客户端请求会经由 Proxy 转发给集群的第一个分片的主节点，因此所有发布与订阅都会请求都会由第一个分片主节点负责处理。



因此，当发布订阅的消息非常多时，使用集群版会给第一个分片的主节点造成很大的压力。

🔗 使用发布与订阅时的注意事项

🔗 避免建立大量大订阅连接

如前文背景信息中所描述，当客户端发送 subscribe 类命令时，Proxy 会与第一个分片的主节点建立连接，并转发此 subscribe 请求，这一条连接会被 subscribe 所独占。假如有大量的客户端均发送 subscribe 命令，那么不同的 Proxy 与第一个分片的主节点之间将建立大量的连接，这很可能导致该节点的连接数被打满。因此，在使用时，需要控制执行 subscribe 的客户端的数量，避免出现存储节点连接数打满的问题。

🔗 需使用独占的连接执行 subscribe 类命令

消息订阅者在执行 subscribe 命令时将独占一条到 Proxy 的服务器连接，并等待订阅的消息到来，这条连接上不能同时执行其他 Redis 命令。

🔗 无需单独创建客户端连接来执行 publish 命令

publish 命令发送至 Proxy 后，Proxy 会使用共享的连接转发至第一个分片的主节点，publish 命令和其他 GET/SET 这样的命令在使用上并无差别。因此无需为执行 publish 命令而单独创建一条到 Proxy 的连接。

Redis 事务使用注意事项

云数据库Redis 集群版和标准版均支持 Redis 事务（transaction）机制。您可以使用 **MULTI**，**EXEC**，**DISCARD**，**WATCH**，**UNWATCH** 命令来执行原子性的事务操作。

🔗 集群版事务相关命令限制

🔗 事务中涉及的 key 必须在同一个 slot 上

事务命令最终需要在 Redis 节点上被执行，为了保证原子性，集群版中 Proxy 会为执行事务的客户端关联一条到 server 的独占连接，这要求用户在事务中可以使用的命令对应的 key 需要落在一个 server 中，准确来说是落在一个分片中，但考虑到集群后续可能做扩缩，而扩缩是以 slot 为粒度进行的，因此当前落在一个分片中的 key，在将来不一定会处于同一个分片，因此要求事务中使用的 key 需要保证落在同一个 slot 上。

🔗 如何保证多个 key 落在同一个分片？

请参考[数据分片规则](#)

部分命令不支持在事务中使用 以下几类命令不支持在事务中使用：

- 不包含 key 的命令，如 info、dbsize、flushall、发布订阅类命令等
- 不支持使用阻塞类命令，比如 blpop 等

版本说明